РОЛИК Александр Иванович

Заведующий кафедрой автоматики и управления в технических системах факультета информатики и вычислительной техники Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского», директор Учебно-научного центра информационно-коммуникационных технологий НЕТКРЕКЕР - НТУУ «КПИ», доктор технических наук, профессор, лауреат Государственной премии Украины в области науки и техники. Область научных интересов: управление ИТ-инфраструктурой; информационные системы и технологии; искусственный интеллект.

ТЕЛЕНИК Сергей Федорович

Декан факультета информатики и вычислительной техники Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского», доктор технических наук, профессор, лауреат Государственной премии Украины в области науки и техники. Область научных интересов: разработка информационных систем и технологий; управление ИТ-инфраструктурой; системы искусственного интеллекта; компьютерная лингвистика.

ЯСОЧКА Максим Владимирович

Технический директор ООО «НЕТКРЕКЕР», доцент кафедры автоматики и управления в технических системах факультета информатики и вычислительной техники Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского», кандидат технических наук. Область научных интересов: управление ИТ-инфраструктурой; проектирование систем управления деятельностью провайдеров информационно-коммуникационных услуг; искусственный интеллект; бизнес-аналитика.

А.И.Ролик, С.Ф.Теленик, М.В.Ясочка

А.И.Ролик
С.Ф.Теленик
М.В.Ясочка

УПРАВЛЕНИЕ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРОЙ

УПРАВЛЕНИЕ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРОЙ

КИЕВ 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ
«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ ИМЕНИ ИГОРЯ СИКОРСКОГО»

А.И. РОЛИК
С.Ф. ТЕЛЕНИК
М.В. ЯСОЧКА

УПРАВЛЕНИЕ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРОЙ

АУТСКИ

КИЕВ НАУКОВА ДУМКА 2018

УДК 004

Монография посвящена проблемам управления новым базовым элементом архитектуры информационных систем ИТ-инфраструктурой. Предложен декомпозиционно-компенсационный подход как концептуальная основа функционирования информационной системы управления ИТ-инфраструктурой. Разработаны двухуровневая модель управления с координатором как базовый элемент управления в многоуровневой системе, а также комплекс моделей, методов и алгоритмов решения отдельных задач управления ИТ-инфраструктурой, прежде всего управления ресурсами и нагрузкой. Разработаны архитектура, структура, принципы функционирования информационной технологии управления ИТ-инфраструктурой. Приведены основные решения по разработке, внедрению и эксплуатации системы управления ИТ-инфраструктурой на основе созданной информационной технологии.

Для исследователей, специалистов-практиков, аспирантов, магистров и студентов разных специальностей, интересующихся проблемами управления ИТ-инфраструктурой.

Монографія присвячена проблемам управління новим базовим елементом архітектури інформаційних систем ІТ-інфраструктурою. Запропоновано декомпозиційно-компенсаційний підхід як концептуальна основа функціонування інформаційної системи управління ІТ-інфраструктурою. Розроблено дворівневу модель управління з координатором як базовий елемент управління в багаторівневій системі, а також комплекс моделей, методів і алгоритмів розв'язання окремих задач управління ІТ-інфраструктурою, насамперед управління ресурсами і навантаженням. Розроблено архітектуру, структуру, принципи функціонування інформаційної технології управління ІТ-інфраструктурою. Наведено основні рішення щодо розроблення, впровадження і експлуатації системи управління ІТ-інфраструктурою на базі розробленої інформаційної технології.

Для дослідників, фахівців-практиків, аспірантів, магістрів та студентів різного фаху, які цікавляться проблемами управління ІТ-інфраструктурою.

Рецензенты:

доктор физ.-мат. наук, профессор, декан факультета информатики, заведующий кафедрой информатики Национального университета «Киево-Могилянская академия» Н.Н. Глыбовец,

доктор техн. наук, профессор, заведующий кафедрой компьютерной инженерии Киевского национального университета имени Тараса Шевченко С.Д. Погорелый

Рекомендовано к печати ученым советом
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»
(протокол № 10 от 12 ноября 2018 г.)

Научно-издательский отдел физико-математической и технической литературы
Редактор А.Я. Бельдий

ISBN 098-966-00-1665-1 © А.И. Ролик, С.Ф. Теленик, М.В. Ясочка, 2018

ВВЕДЕНИЕ

Конец двадцатого и начало двадцать первого столетий характеризуется бурным развитием информационных технологий (ИТ). Одновременно изменения в мировой экономике постепенно выкристаллизовывали новое отношение бизнеса к ИТ. Теперь бизнес рассматривает ИТ как источник развития, выделяя прежде всего, их неограниченные возможности в повышении производительности и улучшении конкурентоспособности.

Постепенный переход от функционального подхода к управлению предприятиями к процессно-функциональному дополнился сервисным подходом. Воспользовавшись развитыми методами реинжиниринга, менеджеры и ИТ-специалисты поставили ИТ во главу угла в новой бизнес-модели, нацеленной на снижение издержек бизнеса и ускорения бизнес-инноваций [238]. Сегодня эффективная поддержка бизнес-процессов, процессов производственных и технологических осуществляется за счет автоматизации, обеспечиваемой ИТ-услугами, как на уровне между предприятиями, так и внутри предприятия (между бизнес-подразделениями и ИТ-подразделениями).

Необратимости становления новых отношений ИТ и бизнеса способствовал экономический кризис, последствия которого экономика и общество еще ощущают. Ведущие специалисты ИТ-отрасли и раньше предупреждали, что вложения в области автоматизации бизнеса не дают ожидаемой отдачи, хотя тогда ИТ были единственной отраслью, для которой не существовало понятия экономической эффективности [67]. И только экономический кризис заставил предприятия сокращать ИТ-бюджеты и остро ставить вопрос об экономичности ИТ.

Оказалось, что в условиях кризиса бизнес не может обеспечивать растущие требования ИТ-подразделений, связанные с приобретением новых технологий, программно-аппаратных средств и увеличением численности персонала. Действительно, автоматизация полнофункционального комплекса бизнес-процессов с учетом их изменений, усложнение распределенных приложений, возрастающее количество необходимых вычислительных ресурсов, ресурсов хранения и передачи данных, появление новых важных для бизнеса информационно-коммуникационных технологий (ИКТ) потребовали расширения ИТ-бюджетов предприятий, а в условиях кризиса это оказалось неосуществимым. Более того, ситуация требовала сокращения капитальных и текущих затрат на ИТ при одновременном расширении их положительного влияния на развитие бизнеса.

Выходом из этой сложной ситуации была консолидация вычислительных, телекоммуникационных и иных ресурсов поддержки бизнес-процессов в центрах обработки данных и представление их предприятиям в качестве услуги. Появилось и очень быстро получило признание понятие ИТ-инфраструктуры, под которым в наиболее общем виде понимают совокупность взаимосвязанных ИТ, сетей, ресурсов и оборудования конечных пользователей. Одной из важнейших

тенденций развития ИТ на переломе тысячелетий явилось создание глобальной, национальных и корпоративных ИТ-структур, обеспечивающих пользователям сбор, передачу, обработку и хранение информации. Сервисный подход был воплощен в ряде концепций, охватывая предоставление в качестве услуги ИТ-инфраструктуры, программного обеспечения, платформ разработки. Это позволило существенно уменьшить как капитальные, так и текущие затраты предприятий на создание и поддержание ИТ-инфраструктуры, оставляя возможность получения прибыли провайдерам указанных услуг.

Исходя из характера и особенностей предприятия, его возможностей и состояния, целей и интересов, можно говорить о создании собственной ИТ-инфраструктуры, ее построению на основе инфраструктурных сервисов либо о гибридном подходе. В любом случае отмеченные выше преимущества построения ИТ-инфраструктуры можно реализовать только при приверженности научному подходу к проектированию и управлению ИТ-инфраструктурой.

Таким образом, появился новый объект проектирования и управления, который требует исследования, постановки и решения соответствующих комплексов проблем. Что касается управления ИТ-инфраструктурой, то достаточно динамично развивается новый класс систем управления, а именно систем управления ИТ-инфраструктурой (СУИ) [106, 120, 154].

При построении СУИ используется достаточно распространенный класс ИТ управления ИТ-инфраструктурой (ИТ УИТИ), создаваемых для повышения эффективности ИТ-инфраструктуры и автоматизации процессов ее обслуживания.

Существующие СУИ и ИТ УИТИ выполнили свою стабилизирующую функцию. Данные аналитических агентств свидетельствуют о том, что инфраструктурными сервисами сегодня в той или иной степени пользуются большинство предприятий, а деятельность провайдеров этих сервисов невозможно представить без СУИ [237, 238, 240, 241, 242, 243].

Однако ситуация постоянно изменяется. Увеличение бизнес-спроса на ИТ-услуги, многообразие ИТ, а также постоянное совершенствование бизнес-процессов и вызванная этим необходимость разработки и внедрения новых ИТ приводят к чрезмерному усложнению СУИ, являющейся продуктом системной интеграции многообразных подходов и порой несовместимых решений от различных производителей. Постоянно возрастающая сложность управления ИТ-инфраструктурой и увеличение затрат на операционную деятельность, вызывает необходимость поиска новых подходов к управлению ИТ-инфраструктурой.

Управление ИТ-инфраструктурой является сравнительно новым разделом управления, но оно опирается на результаты ряда направлений научных исследований, имеющих статус классических. Современным взглядам на построение информационных систем присуща традиционная иерархическая концепция их построения с выделением организационного, производственного и технологического уровней. Соответственно и собственно ИТ-инфраструктура как объект управления не избежала такого деления. Естественным выглядит широкое применение для управления ИТ-инфраструктурой моделей и механизмов, применяемых при решении широкого круга задач управления в технических и организационных системах – от управления технологическими процессами до принятия решений при управлении предприятиями.

Системы управления ИТ-инфраструктурой являются новыми для теории управления техническими системами и характеризуются такими свойствами, как многообъектность, рассредоточенность объектов и распределенность

средств управления, большое количество параметров и переменных, наличие факторов неопределенности и риска. Высокая стоимость ИТ-инфраструктуры и ее важность для бизнеса, с одной стороны, низкая эффективность ИТ и медленный возврат вложений в ИТ-инфраструктуру, с другой, определяют необходимость разработки теоретических и методических основ автоматизированного управления ИТ-инфраструктурой. Характер задач, решаемых СУИ, существенно отличается от задач, решаемых традиционными системами автоматизированного и автоматического управления и регулирования. Необходимость согласования действий между подсистемами СУИ, цели функционирования которых могут существенно различаться, большое количество рисков и неопределенностей, возникающих при выборе управляющего воздействия, большой диапазон и динамика изменения внешних воздействий, множество режимов функционирования СУИ обусловили появление новых функциональных задач управления: рационального распределения и оперативного перераспределения ресурсов информационно-коммуникационной системы, координации действий подсистем, оперативного управления уровнем услуг и функционирования ИТ-инфраструктуры. Несмотря на практические успехи в области автоматизированного управления многообъектными распределенными системами управления, к которым относится СУИ, работы по созданию теоретических основ управления такими системами далеки от завершения [67].

Наиболее масштабные исследования в области управления ИТ-инфраструктурой проводятся международными организациями (TM Forum, Office of Government Commerce, IEEE, ISO и др.), аналитическими агентствами (Forrester Research, Gartner и др.), корпорациями (IBM, Cisco, Microsoft, Hewlett-Packard и др.). Процессному управлению ИТ-инфраструктурой посвящено множество работ [191, 204, 208, 256, 264], в то время как результаты исследований оперативного управления ИТ-инфраструктурой нашли отражение лишь в немногочисленных публикациях [105, 106, 108], количество которых постоянно увеличивается. Управление ИТ-инфраструктурой опирается на такие направления научных исследований как системный анализ [42], математические методы решения оптимизационных задач [41], информационные технологии [81, 137], теория информационных систем [32], современная теория управления [72, 73], теория иерархических систем [69], искусственный интеллект [168].

Однако эти исследования не решают всех проблем, связанных с разработкой ИТ УИТИ. Недостаточно проработаны такие аспекты управления ИТ-инфраструктурой, как оперативное управление качеством ИТ-услуг, рациональное использование ресурсов ИТ-инфраструктуры, анализ функционирования элементов ИТ-инфраструктуры, эффективное распределение ресурсов и нагрузки ИТ-инфраструктуры с учетом прогнозирования и др.

В настоящее время развитие ИТ находится под сильным влиянием таких факторов как консолидация и виртуализация ресурсов, широкое внедрение облачных вычислений, конвергенция телекоммуникационных технологий и услуг. В информационной индустрии под воздействием этих факторов осуществляется активный процесс глобализации информационных и телекоммуникационных технологий, формируется новая ИТ-среда, предоставляющая перспективные средства ведения бизнеса. Революционные преобразования в области ИТ способствуют прогрессу бизнес-технологий, однако эффективность применения ИТ сдерживает отставание в области технологий управления ИТ-инфраструктурой.

Высокая стоимость владения полнофункциональной ИТ-инфраструктурой и существенная зависимость успешности бизнеса от качества ИТ-услуг делают актуальной научно-прикладную проблему создания проблемно-ориентированной информационной технологии управления корпоративной ИТ-инфраструктурой, направленной на поддержание согласованного уровня ИТ-услуг при рациональном использовании ресурсов ИТ-инфраструктуры в условиях их виртуализации, кластеризации и консолидации, а также при значительной динамике запросов пользователей. Решению этой задачи и посвящена данная монография.

Авторы сделали попытку систематизировать задачи управления ИТ-инфраструктурой, разработать концептуальные основы управления ею, предложить набор моделей и методов решения отдельных задач и определить решения по созданию ИТ и СУ ИТ-инфраструктурой.

Поэтому в работе переплетаются технические, технологические, математические аспекты, во взаимосплетении которых читатель найдет интегральную составляющую, следование которой, по мнению авторов, принесет пользу тем из читателей, которые прочтут весь предложенный материал.

В течение длительного периода времени в «КПИ имени Игоря Сикорского» были разработаны, исследованы и внедрены многие методы и модели эффективного управления доступом к ограниченным информационно-вычислительным ресурсам ИТ-инфраструктуры [114, 148, 156, 157], методы распределения виртуальных машин в центрах обработки данных [147, 158, 159, 160], ряд систем управления функционированием ИУС и корпоративными ИТ-инфраструктурами [106, 120, 154, 155]. Поэтому авторы надеются, что монография является первой из серии монографий, посвященных как управлению ИТ-инфраструктурой, так и ее проектированию и эффективному использованию.

В Украине комплексом проблем проектирования, управления и исследования ИТ-инфраструктуры занимается ряд коллективов ведущих университетов и научно-исследовательских институтов.

В Киевском национальном университете имени Тараса Шевченко разработан теоретический аппарат проектирования, анализа свойств математических моделей ИТ-инфраструктур и инфраструктурных сервисов на основе транзиционных систем, сетей Петри, алгебраических систем и многоосновных алгебр, неклассических и нечетких логик [59, 60, 61, 311]. На основе разработанных моделей создано прикладную схематологию алгоритмического обеспечения для распределенных архитектур [71, 93, 278]. В университете накоплен опыт использования ИТ-инфраструктур для решения важных классов задач анализа, планирования и управления в разных сферах деятельности человека [92, 94, 140].

В Институте кибернетики имени В.М. Глушкова разработаны методология и ряд технологий решений сложных научных задач на суперкомпьютерах на основе концепции параллелизма вычислений на нескольких уровнях абстракции [83, 84]. Для отдельных уровней абстракции предложены технологии создания компонентов ИТ-инфраструктуры, в частности FPGA [311]. В Институте ведутся исследования по разработке мультиагентных и реконфигурируемых систем [35, 36, 82, 309, 310].

Эффективное использование ИТ-инфраструктуры для решения многих важных для экономики задач и формирования ИТ-среды составило предмет исследований в Национальном университете «Киево-Могилянская академия», Институте кибернетики имени В.М. Глушкова, Киевском национальном университете имени Тараса Шевченко. Это позволило разработать теоретические

основы предоставления программного обеспечения как сервиса и исходя из этого предложить платформы для создания прикладных программных систем [24, 25, 26].

Разработанные платформенные решения нашли широкое применение. Созданные на их основе программные сервисы используются в органах государственной власти для анализа и принятия решений в научно-исследовательских учреждениях для решения сложных научных задач, на предприятиях и в учебных заведениях [9, 19, 27,311].

Объединение усилий разных коллективов может дать интересные результаты, что позволит развить комплексный подход к решению проблем проектирования, управления и использования ИТ-инфраструктуры. Монография [71] – только один из примеров такого сотрудничества. В ней определены масштабы проблемы, очерчены направления исследований, предложены модели и методы для рада задач. В определенном смысле ее можно рассматривать как программу действий. Несомненно, следует ожидать не только новых коллективных монографий, но и внедрения предлагаемых решений в интенсивно создаваемые в Украине ЦОД.

АУТСКИЙ

ГЛАВА 1

АНАЛИЗ ПРОБЛЕМ, ТЕНДЕНЦИЙ И ПЕРСПЕКТИВ РАЗВИТИЯ УПРАВЛЕНИЯ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРОЙ

В главе проведен анализ проблемы в области управления информационными технологиями и ИТ-инфраструктурами. Определены основные факторы, под воздействием которых будет развиваться управление ИТ, и тенденции его развития. Осуществлена декомпозиция ИТ-инфраструктуры выделением в ней четырех иерархических уровней: сервисов, приложений, вычислительных ресурсов и сетевого взаимодействия. Предложена обобщенная схема ИТ-инфраструктуры корпоративного уровня, определен состав уровней и показано место системы управления ИТ-инфраструктурой.

1.1. КОРПОРАТИВНАЯ ИТ-ИНФРАСТРУКТУРА КАК ОБЪЕКТ УПРАВЛЕНИЯ

1.1.1. Анализ тенденций развития систем обработки информации

Современный этап развития систем обработки информации завершает этап эволюции ИТ-индустрии, возвращаясь к сосредоточенным центрам обработки информации, которые в 60-х–80-х годах прошлого века существовали в виде вычислительных центров коллективного пользования (ВЦКП), а в наше время – в форме центров обработки данных (ЦОД), представляющих собой совокупность информационной, телекоммуникационной и инженерных инфраструктур. При этом мы стали свидетелями очередного витка эволюции ИТ, признаком которого является повсеместное использование облачных вычислений, реализуемых на основе ЦОД. Принятие парадигмы облачных вычислений предопределяет формирование новой ИТ-среды, которая составит основу формирующегося ИТ-общества.

В современных больших корпоративных ИТ-инфраструктурах на качественно новом уровне реализуются идеи, связанные с усовершенствованием информационно-вычислительных процессов. Так, в 70-х–80-х годах прошлого столетия перспективной была идея концентрации ИТ-ресурсов в ВЦКП – специализированное подразделение, которое имело комплекс помещений с вычислительной техникой и обслуживающий персонал, и было предназначено для предоставления вычислительных услуг [131]. Ввод заданий в ЭВМ и вывод результатов осуществлялись через терминалы, расположенные на территории ВЦКП. Задачи пользователей выполнялись в режиме разделения времени, а в случае применения мультипроцессорных систем использовалась параллельная обработка информации.

Министерства и ведомства с их распределенной системой подчиненных предприятий, организаций, учреждений и отдельных подразделений видели в ВЦКП многообещающую форму обеспечения подчиненных органов управления

вычислительной техникой и увеличения эффективности ее использования. Воплощению этой идеи помешал ряд причин, прежде всего появление и повсеместное использование ПК, уступающих по мощности ЭВМ в ВЦКП, но имеющих гораздо меньшую стоимость. Это привело к повсеместному переходу от централизованной к децентрализованной обработке информации, а развитие сетевых технологий, появление интернета и увеличение скорости доступа к этой глобальной сети позволило объединять в единое целое совокупность разнородных вычислительных систем и терминалов и коллективно использовать вычислительные и коммуникационные ресурсы.

В настоящее время в средних и крупных предприятиях и организациях превалирует тенденция к консолидации вычислительных ресурсов в выделенных и специальным образом оборудованных серверных помещениях, что позволяет существенно сократить затраты на обслуживание парка вычислительной техники, осуществляющей обработку информации и обеспечивающей базу поддержки бизнес-приложений. Серверные помещения и ВЦКП стали прообразом специализированных зданий с мощной инженерной инфраструктурой, огромным количеством серверов, системами хранения данных и пр., называемых ЦОД. За рубежом бум создания ЦОД пришелся на период 1995–2000 гг. В настоящее время строительство ЦОД продолжается и их количество постоянно увеличивается.

Интеграция информационно-вычислительных ресурсов в ЦОД стала повсеместной и воплощена многими крупными корпорациями в собственных ИТ-инфраструктурах преимущественно для обеспечения доступа большого количества пользователей к коллективно используемым ресурсам в виде сервисов, приложений, вычислительных мощностей, данных и т. п.

Можно полагать, что ВЦКП вернулись в виде ЦОД – комплексных организационно-технических решений, предназначенных для создания высокопроизводительных, отказоустойчивых ИТ-инфраструктур. Ориентированные на решение проблем бизнеса путем предоставления услуг в виде ИТ-сервисов ЦОД обеспечивают эффективное консолидированное хранение и обработку данных пользователей, предоставление им прикладных сервисов и ИТ-услуг, а также поддержку функционирования корпоративных приложений. В ЦОД повсеместно используется технология виртуализации, позволяющая не только значительно повысить эффективность использования существующего парка серверов и сетевых ресурсов, но и исключить зависимость бизнес-процессов от производительности конкретных физических ресурсов.

Но оказалось, что эффективная организация ЦОД требует решения ряда проблем, прежде всего создания условий для функционирования информационно-вычислительных мощностей ЦОД, управления ресурсами, обеспечения надежности, безопасности и пр. Такие проблемы встают перед любой компанией, которая, вкладывая средства в создание ЦОД для обслуживания пользователей других компаний, приобретает статус хостинговой компании.

Таким образом, в области организации высокопродуктивных многопользовательских систем телеобработки информации четко просматриваются витки эволюции от централизованных систем на основе мэйнфреймов к децентрализованным на основе ПК и далее к централизованным на основе ЦОД с использованием клиент-серверных технологий. В перспективе просматривается переход к децентрализованным системам на основе Grid или подобных технологий, когда для предоставления услуг пользователям задействуются ресурсы нескольких ЦОД. Это подтверждается тем, что сегодня ИТ индустрия находится на пороге

возможных революционных преобразований, основой которых является парадигма облачных вычислений. Ведь при предоставлении всего спектра многочисленных ИТ-услуг, необходимых для успешного ведения современного бизнеса, потребуются ресурсы не одного, а нескольких ЦОД, каждый из которых будет предоставлять только услуги на основе облачных вычислений.

1.1.2. Основные современные концепции предоставления ИТ-услуг

В настоящее время развитие вычислительных сред и информационных систем разных классов ориентируется на технологии предоставления ИТ-услуг, а в организации доступа к ресурсам превалируют сервис-ориентированные модели.

Появление большого количества ЦОД позволило быстро воплотить в жизнь концепцию облачных вычислений (cloud computing). Метафора «облако», используемая в данном случае, подразумевает то, что все технические детали организации услуг, предоставляемых через интернет, скрыты от пользователей. Основополагающая терминология и обязательные характеристики облачных вычислений зафиксированы в стандарте [371].

Сегодня облачные вычисления стали общепризнанной парадигмой предоставления ИТ-сервисов. Отказ компании-потребителя от собственной ИТ-инфраструктуры и переход на арендованную в «облаке» позволяет ей не только минимизировать капитальные и эксплуатационные затраты на собственную ИТ-инфраструктуру, но и ускоряют внедрение услуг, при этом существенно возрастает их гибкость и эффективность при значительном снижении стоимости услуг. Сокращение затрат на собственную ИТ-инфраструктуру при использовании облачных ресурсов достигается отказом от собственных хранилищ данных, высокопроизводительных вычислительных комплексов и других дорогих специализированных систем.

Перспективность и популярность облачных вычислений объясняются используемой моделью обслуживания, обозначаемой акронимом XaaS, который является собирательным для терминов «X as a Service», «Anything as a Service» или «Everything as a Service»; их можно перевести как «Все что угодно как услуга».

Наиболее распространенными реализациями XaaS являются SaaS (программное обеспечение как услуга), IaaS (инфраструктура как услуга) и PaaS (платформа как услуга). Комбинацию этих трех сервисов называют моделью SPI (**S**aaS, **P**aaS, **I**aaS). Кроме этих сервисов популярность приобретают высокопроизводительные вычисления как услуга (High-Performance Computing as a Service, HPCaaS), хранение данных как услуга (STaaS), коммуникации как услуга (CaaS), сети как услуга (NaaS), мониторинг как услуга (MaaS) и др.

Акроним XaaS ассоциируется с ИТ-сервисами, предоставляемыми по принципу «on-demand computing» – вычисления по требованию. При этом пользователю может быть предоставлен практически неограниченный объем ресурсов без предварительных капиталовложений и последующих эксплуатационных расходов на администрирование ресурсов и систем. Ключевыми характеристиками облачных вычислений являются виртуализация и динамическая масштабируемость.

Концепция облачных вычислений позволяет в полной мере воплотить в жизнь идею коммунальных вычислений (utility computing), возникшую еще в 1960-е годы. В современном понимании коммунальные вычисления позволят

пользователям при минимальных затратах на ИТ-инфраструктуру, которая в идеале может быть сведена только к пользовательским компьютерам, подключенным к сети интернет, получать услуги, оплачивая только объем потребленных ресурсов. В этом случае пользователи не закупают лицензии на ПО и не несут затраты на его инсталляцию и обслуживание.

В тоже время для облачных вычислений не утрачивают актуальности вопросы безопасности. Пользователей продолжают беспокоить сохранность данных, уязвимость к атакам, управление учетной информацией пользователей, контроль доступа к ресурсам и данным, а также другие вопросы сохранности корпоративной информации, вынесенной за границы собственных сетевых экранов.

Получить в распоряжение большую вычислительную мощность и существенные объемы хранимых данных пользователю позволяет Grid-технология [232]. Это достигается путем объединения с помощью телекоммуникационной сети и специализированного ПО множества разнотипных вычислительных ресурсов и хранилищ или слабосвязанных, гетерогенных, разрозненных компьютеров в виртуальную систему с огромными вычислительными возможностями и/или большими объемами хранилищ данных. Кроме разрозненных компьютеров в Grid-систему могут быть объединены выделенные ресурсы нескольких ЦОД, с обеспечением доступа к ресурсам пользователей, которыми в настоящее время в основном являются сотрудники правительственных, научных и образовательных организаций.

Первоначально перед Grid-технологией ставилась цель объединения добровольно предоставляемых неиспользуемых в какое-то время ресурсов слабосвязанных персональных компьютеров в распределенную инфраструктуру для достижения большой вычислительной мощности. Полученный таким образом огромный, но по отдельности ненадежный ресурс применяли для решения исследовательских инициативных задач. Позже Grid-технология была взята на вооружение научными и образовательными организациями для создания Grid-систем, совместно использующих ресурсы инфраструктур этих организаций. Grid-системы в основном предназначаются для решения задач или реализации проектов, требующих больших вычислительных мощностей и/или больших суммарных объемов хранилищ данных.

Компоненты распределенных вычислений, называемых в Grid-технологии кластерами, объединяются для совместного решения задач с помощью специального ПО. ПО Grid-систем называется промежуточным ПО. Промежуточное ПО, взаимодействуя с системным и сетевым ПО, обеспечивает работу прикладных приложений пользователей, обеспечивающих решение конкретных прикладных задач. Наиболее популярным промежуточным ПО являются ARC, gLite, UNICORE и SGE. Это ПО управляет распределенным выполнением задач пользователей и обеспечивает выделение необходимых объемов распределенных вычислительных ресурсов, таких как процессоры, память, дисковое пространство.

Кроме промежуточного ПО для реализации Grid-системы требуется создание виртуальной организации, за которой стоят учреждения или отдельные физические лица, объединенные строгими правилами коллективного доступа к распределенным вычислительным ресурсам.

Обсуждая технологии хранения данных, нельзя забывать об одноранговых сетях, называемых peer-to-peer или P2P. Персональные компьютеры большого количества пользователей и соответствующее промежуточное ПО позволяют самим пользователям предоставлять контент другим пользователям. При этом в

P2P хранятся большие объемы данных, но о надежности хранения и доступе к этим данным можно судить, только оперируя статистическими показателями.

Необходимо отметить, что облачные и Grid-технологии не исчерпывают все современные идеологии и подходы в области распределенных вычислений. Появление новых или модификация известных парадигм распределенных вычислительных систем приводит постоянному перемешиванию разных концепций и подходов при организации и предоставлении ИТ-сервисов. Веб-ориентированные и клиент-серверные архитектуры построения программных систем, сервис-ориентированный и компонентно-базированный подходы к организации функционирования, проектирования и реализации информационных систем и другие концепции, обладая неоспоримыми преимуществами, не уступают модным подходам, не становятся полностью забытыми, а интегрируются с ними, создавая все возрастающее пространство решений. Это конечно, создает трудности выбора разработчиками систем технологий, сервисов, компонентов. Но одновременно формируется основа для превращения проектирования в симбиоз науки и искусства, появления комбинаторики построения нового решения из множества существующих.

Если сравнить наиболее популярные концепции предоставления ИТ-услуг, не забывая о централизованном предоставлении ресурсов, опираясь на практику их применения, многочисленные обзоры аналитических агентств и личный опыт, то можно отметить следующее.

Централизованная обработка данных в ЦОД или серверных решает множество задач построения ИТ-систем масштаба корпорации надежнее и безопаснее, чем распределенная на разрозненных серверах или Grid-системах.

Grid-системы ориентируются на решение определенного класса задач и оказываются очень эффективными при решении задач, возникающих в промышленности, науке и технике и требующих большого объема вычислений. Ориентация виртуальных организаций на совместное решение определенных задач сильно затрудняет использование Grid-технологии для массового предоставления ИТ-услуг разнообразным коммерческим структурам. Этому способствует и тот факт, что виртуальная организация не несет финансовой ответственности за качество предоставляемых услуг и не может гарантировать выполнения SLA. Это объясняется тем, что предоставление услуг, например XaaS, коммерческий ЦОД производит на платной основе, в то время как организации или физические лица предоставляют свои ресурсы в Grid-системы бесплатно, часто руководствуясь соображениями престижа или из волонтерских побуждений. Кроме того, снижению привлекательности со стороны малого бизнеса к использованию Grid-систем способствуют сложности организационного характера при получении большого количества разнообразных сервисов.

Арендованная ИТ-инфраструктура требует меньше затрат (как капитальных, так и эксплуатационных) и ускоряет разработку и внедрение ИТ-систем.

Количество облачных ресурсов постоянно растет, качество предоставляемых сервисов улучшается, а их выбор расширяется [328].

Таким образом, в настоящее время для корпораций и крупных компаний перспективным подходом к построению ИТ-систем является подход, предполагающий комплексное предоставление услуг, когда часть услуг предоставляется собственными централизованными ресурсами, а остальные услуги – сторонними провайдерами на основе облачных и других перспективных технологий. Это предполагает усиление роли рынка ИТ-услуг. Компании могут предоставлять

ИТ-ресурсы как услуги (выступая в роли провайдеров ИТ-услуг) и потреблять ИТ-ресурсы как услуги (выступая в роли потребителей). Существенно возрастает роль автоматизации процессов управления уровнем услуг и ИТ-инфраструктурой на стороне потребителя услуг и аппаратно-программными ресурсами ЦОД на стороне оператора облачных вычислений.

1.1.3. Обобщенная схема корпоративной ИТ-инфраструктуры

Терминология в области предоставления ИТ-услуг только складывается. Это касается и собственно термина «ИТ-инфраструктура» [121, 191, 265]. Все определения ИТ-инфраструктуры, которые можно найти в литературе и сети интернет, согласны в том, что ИТ-инфраструктура включает в себя аппаратно-программные элементы и все компоненты информационных технологий, обеспечивающие основу информационной системы. К ИТ-инфраструктуре относят все, что играет хоть какую-нибудь роль в предоставлении ИТ-услуг бизнес-пользователям. Разные определения термина «ИТ-инфраструктура» различаются тем, что персонал, отвечающий за непрерывное предоставление ИТ-услуг и обеспечивающий надлежащее функционирование аппаратно-программных компонентов, включается или выносится за рамки ИТ-инфраструктуры. Так, во многих работах, в частности в [121], отмечается, что администраторы, разработчики и другой персонал, имеющий доступ к аппаратно-программным компонентам и в той или иной степени отвечающий за качество ИТ-услуг, также являются частью ИТ-инфраструктуры. В тоже время терминология ITIL [191, 265] в явном виде выносит ИТ-персонал за рамки ИТ-инфраструктуры. Это объясняется тем, что согласно идеологии ITIL ИТ-персонал, процессы управления и документация – это предмет особого рассмотрения, которому в документации по ITIL уделено основное внимание.

В данной работе ИТ-персонал, занимающийся управлением ИТ-инфраструктурой и ее обслуживанием, в явном виде в состав ИТ-инфраструктуры не включен, но анализ и решение проблем управления ИТ-инфраструктурой осуществляются без отрыва от важнейшего организационно-структурного компонента управления ИТ-инфраструктурой, каковым является ИТ-подразделение. ИТ-подразделение, политики и процедуры управления, а также правила эксплуатации можно рассматривать как организационную структуру управления ИТ-инфраструктурой.

Во избежание возможных недоразумений сформируем авторское видение ИТ-инфраструктуры путем актуализации существенных для изложения ее особенностей из обширного набора приписываемых ей потенциальных черт.

ИТ-инфраструктура (инфраструктура информационных технологий) – организационно-техническая совокупность программных, вычислительных и телекоммуникационных средств, предоставляющая информационные, вычислительные и телекоммуникационные ресурсы и ИТ-услуги бизнес-пользователям, необходимые для автоматизации выполнения сотрудниками или партнерами процессов деятельности и решения бизнес-задач. ИТ-инфраструктура является важнейшей частью автоматизированной или информационно-управляющей системы (ИУС) предприятия, поскольку она включает в себя средства, необходимые для создания, функционирования и развития ИТ-среды предприятия. От эффективности функционирования ИТ-инфраструктуры непосредственно зависит качество ИТ-сервисов и, как следствие, эффективность ведения бизнеса.

В настоящее время ввиду эффективности централизованной обработки информации и низких экономических затрат на обслуживание корпоративные ИТ-инфраструктуры целесообразно строить по схеме, приведенной на рис. 1.1. В этом случае услуги пользователям представляются в виде комбинации услуг, предоставляемых собственной инфраструктурой, и ИТ-услуг, предоставляемых сторонними провайдерами. Причем тот факт, что часть, а то и все услуги предоставляются посредством облачных технологий, может быть неизвестен и не ощущаться пользователями, поскольку за качество абсолютно всех услуг отвечает ИТ-подразделение компании.

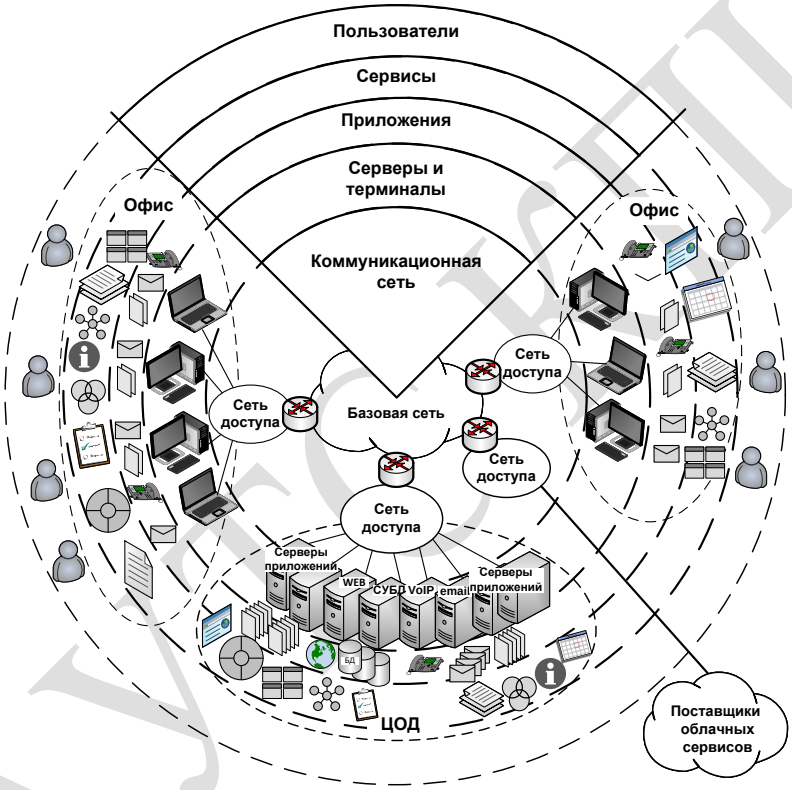


Рис. 1.1. Обобщенная схема корпоративной ИТ-инфраструктуры с централизованной обработкой данных и привлечением облачных услуг

Необходимо отметить, что ИТ-инфраструктуру следует проектировать и внедрять только с максимальной функциональностью. Так, предприятия, ограничивающиеся использованием в локальной сети лишь некоторых бизнес-приложений установленных на сервер и позволяющих несколько повысить производительность сотрудников за счет автоматизации ряда бизнес-процессов, могут нести существенные убытки из-за, например, потери данных вследствие отсутствия системы резервного копирования. ИТ-инфраструктура, в которой отсутствует только часть важных составляющих, будет более эффективной, но величина этого эффекта может быть незначительной. Максимальный эффект от функционирования ИТ-инфраструктуры может быть достигнут только при наличии

комплексной полноценной ИТ-инфраструктуры, позволяющей рационально выполнять процессы деятельности и эффективно решать бизнес-задачи. В тоже время организация такой инфраструктуры – сложный и длительный процесс, требующий от предприятия серьезных капиталовложений на ее создание и существенных текущих затрат на ее поддержание.

Под управлением ИТ-инфраструктурой подразумевается весь спектр задач и проблем управления: от управления элементами до управления уровнем предоставляемых ИТ-услуг. Управление ИТ-инфраструктурой охватывает и задачи управления ИТ-персоналом.

Для повышения эффективности функционирования ИТ-инфраструктуры и ее управления, а также автоматизации процессов обслуживания используются СУИ, которые создаются на основе ИТ УИТИ.

При рассмотрении вопросов, связанных с управлением ИТ-инфраструктурой, в зависимости от специфики бизнеса и характера решаемых в СУИ задач, выделяется разное количество иерархических уровней. Так, в [191] в инфраструктуре ОТС выделяются пять иерархических уровней: телекоммуникационной технологии; сети; сервисов; абонентский; телекоммуникационного бизнеса. Следуя модели OSI [261], концепция системы управления сетями TMN рассматривает сетевую логическую архитектуру, в которой выделяется пять уровней управления [320]: уровень сетевых элементов; уровень управления элементами; уровень управления сетью; уровень управления услугами; уровень бизнес-управления ОТС. В [45] концепция управления ИТ-инфраструктурой рассматривается применительно к иерархической структуре, состоящей из трех иерархических слоев: сетей; систем; ИТ-сервисов.

В настоящее время отсутствуют методы и принципы идеальной декомпозиции любой сложной системы. Декомпозиция проводится исследователем, который руководствуется только характером решаемых им задач [42].

Исходя из решаемых в работе задач управления, целесообразно представлять изображенную на рис. 1.1 ИТ-инфраструктуру предприятия в виде обобщенной иерархической схемы (рис. 1.2). Такое представление позволяет уточнить место ИТ-инфраструктуры в обобщенной схеме предоставления ИТ-услуг и взаимодействие основных составляющих ИУС – корпоративной ИТ-инфраструктуры, ИТ-подразделения и СУИ.

В ИТ-инфраструктуре (совокупности серверов, систем хранения данных, программных приложений, ПК и телекоммуникационной сети [120]), предлагается выделить четыре иерархических уровня: сервисов; приложений; вычислительных ресурсов; сетевого взаимодействия. При необходимости каждый из четырех уровней может быть разделен на подуровни [121].

Сервисы четвертого уровня делятся на бизнес-сервисы и технологические сервисы. Бизнес-сервисы используются непосредственно для автоматизации выполнения бизнес-процессов. Технологические сервисы не зависят от специфики бизнеса предприятия. К ним относятся сервисы информационной безопасности, электронная почта, видеоконференцсвязь, веб-сервисы, сервис передачи файлов, IP-телефония, средства обеспечения и контроля доступа в интернет, служба мгновенного обмена сообщениями и пр. Сервисы такого типа используются в технологических (обеспечивающих) подсистемах ИУС. К этому же уровню относятся коммуникационные технологии и средства коллективной работы, необходимые для эффективной работы компании.

Когда для предоставления услуг используются облачные технологии по моделям SaaS, PaaS, IaaS и другим, ИТ-подразделение решает организационные вопросы и осуществляет контроль качества услуг.

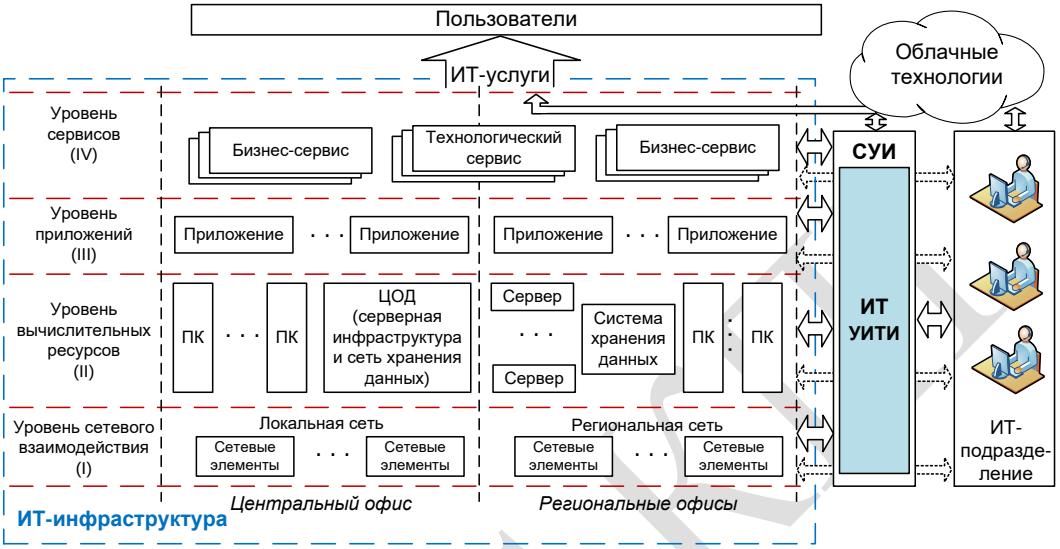


Рис. 1.2. Обобщенная иерархическая схема корпоративной ИТ-инфраструктуры

К третьему уровню относятся распределенные приложения, имеющие непосредственное отношение к автоматизации выполнения бизнес-процессов или процессов деятельности и предоставления ИТ-услуг пользователям. На этом уровне работают приложения, обеспечивающие базовую функциональность ИУС, такие как планирование ресурсов предприятия (ERP), управление персоналом (HRM), управление взаимоотношениями с клиентами (CRM), управление информацией об изделиях (PLM), управление рабочими процессами (Work Flow), документооборотом (СЕДО), управление бизнес-услугами (BSM) и другие, необходимые для успешного ведения бизнеса. С этим уровнем связаны программные продукты, определяющие значимость бизнес-процессов для предприятия на некотором интервале времени, и компоненты СУИ, управляющие распределением вычислительных и коммуникационных ресурсов. Так, например, BSM позволяет настраивать инфраструктурные ресурсы на приоритеты бизнеса. На этом уровне сервис предоставляют функциональные подсистемы ИУС.

Ко второму уровню отнесено серверное оборудование, ПК пользователей, системы хранения и управления данными, базы данных (БД) с соответствующими системами управления БД (СУБД), кластерные решения, а также все вычислительные ресурсы ЦОД. Небольшие предприятия или филиалы организаций практически всегда вместо ЦОД используют серверные или кластерные решения, на которых размещают серверные части ПО уровней бизнес-приложений и универсальных сервисов. Крупные предприятия используют ЦОД, в которых осуществляются консолидированное хранение и обработка данных пользователей, предоставление им ИТ-услуг, централизованная поддержка функционирования приложений. Для повышения гибкости и быстрого перераспределения мощностей пулов серверов под нужды приложений, а также с целью снижения стоимости владения ЦОД повсеместно применяются виртуализация и кластеризация

ресурсов, унифицированные коммуникации и иные базовые технологии. Опыт эксплуатации ЦОД показал, что эффективное использование их ресурсов возможно только при оснащении ИТ-персонала инструментарием управления ИТ-инфраструктурой. Создание инструментария требует разработки моделей и методов решения задач управления с последующей их реализацией в ИТ УИТИ.

Элементы первого уровня – уровня сетевого взаимодействия – обеспечивают доступ пользователей ко всем услугам и ресурсам ИТ-инфраструктуры. При этом доступ сотрудников центрального офиса осуществляется по локальной сети, а сотрудников региональных офисов или сотрудников, находящихся вне компании, по региональной сети, которой практически всегда является интернет, с использованием средств удаленного доступа. Телекоммуникационная сеть, будучи конвергентной, используется для транспорта данных ИТ-сервисов, для передачи сигналов управления ИУС и сетью, а также операторами телекоммуникационных сервисов (ОТС) для предоставления телекоммуникационных услуг.

Для эффективного функционирования ИУС необходимо согласованное взаимодействие трех базовых компонентов – ИТ-инфраструктуры, ИТ-подразделения и СУИ. Соответственно, можно выделить три инфраструктурных объекта управления ИУС – ИТ-инфраструктура, ИТ-подразделение и СУИ.

Обязательной составляющей ИУС предприятий среднего и крупного бизнеса является ИТ-подразделение, его главная задача – обеспечение непрерывности предоставления пользователям ИТ-услуг, внесенных в каталог. Подробности предоставления ИТ-услуг регламентируются пакетом соглашений об уровне сервиса (SLA), заключенных между бизнес-подразделениями и ИТ-подразделением. В SLA определяются значения ключевых показателей эффективности (KPI) и качества (KQI) – ограниченный набор объективно измеримых параметров, позволяющий оценить качество ИТ-услуг [10]. Одним из критериев эффективности функционирования ИТ-инфраструктуры является соответствие фактических значений показателей качества значениям показателей качества ИТ-услуг, заданными в SLA [10]. Для поддержания значений KPI и KQI на уровне, зафиксированном в SLA, администраторы обеспечивают непрерывное функционирование ИТ-инфраструктуры, осуществляют модернизацию, обслуживание и ремонт.

При этом администраторы могут управлять элементами ИТ-инфраструктуры или с помощью СУИ, или непосредственно (см. пунктирные двунаправленные стрелки на рис. 1.2). В последнем случае используются специализированные или универсальные средства управления отдельными элементами ИТ-инфраструктуры.

В соответствии с ITIL в управлении ИТ-услугами принимают участие минимум два отделения ИТ-подразделения – поддержки сервисов (Service Desk), обеспечивающее поддержку пользователей, и управление ИТ-инфраструктурой (ICT Infrastructure Management Team), отвечающее за функционирование ИУС.

Управления ИТ-подразделением подразумевает прежде всего организационные формы управления, которые хорошо проработаны в материалах ITIL [230, 380], а СУИ предоставляет средства автоматизации управления ИТ-подразделением.

Внедрение СУИ позволяет существенно уменьшить затраты на содержание ИТ-подразделения предприятия не только за счет автоматизации функций управления ИТ-инфраструктурой, но и за счет решения задач автоматизации управления ИТ-подразделением.

В данной работе исследуются только методы и средства управления ИТ-инфраструктурой и не рассматривается система эксплуатации, являющаяся социально-технической средой, которая охватывает и ИТ-ресурсы, и ИТ-специалистов, взаимодействует с бизнес-процессами, во многом определяет технологическую культуру компании и уровень компетенции ее сотрудников. Вопросы управления ИТ-подразделением можно рассматривать отдельно, что не повлияет на концепцию построения и функционирования СУИ.

1.1.4. Предназначение системы управления ИТ-инфраструктурой

СУИ предназначена для организационного и программно-технического управления ИТ-инфраструктурой, функциональными и вспомогательными подсистемами ИУС, всеобщей информатизации и автоматизации процессов управления ИТ-инфраструктурой, а также автоматизации деятельности администраторов и руководителей ИТ-подразделения [71, 120].

СУИ в целом должна обеспечивать наилучшие значения параметров функционирования функциональных и технологических подсистем ИУС, телекоммуникационной сети и остальных элементов ИТ-инфраструктуры. Для этого в ней должны быть реализованы методы и алгоритмы достижения оптимальных значений основных критериев оценки качества работы ИТ-сервисов, поддерживающих выполнение бизнес-процессов или процессов деятельности. Кроме того, СУИ должна вести инвентаризационные базы данных об информационных, вычислительных и телекоммуникационных активах предприятия, осуществлять мониторинг состояния и управление поддержанием состояния этих активов на заданном уровне, управлять доступом и распределением вычислительных и телекоммуникационных ресурсов, качеством ИТ-услуг, контролировать и анализировать поведение и поддержку пользователей, автоматизировать планирование и управление закупкой ИТ-активов, а также ведение отчетности ИТ-подразделения. СУИ должна автоматизировать решение множества других задач, перечень которых определяется размерами и особенностями ИТ-инфраструктуры [71].

В настоящее время создание и развертывание СУИ в основном осуществляется с целью обеспечения надлежащей работы функциональных, вспомогательных подсистем ИУС и телекоммуникационной сети, а также их эффективной совместной работы в соответствии с определенным регламентом.

Согласно современным взглядам на проектирование систем управления СУИ представляет собой многомерную многообъектную систему управления (СУ), имеющую иерархическую структуру с частичной децентрализацией функций управления [22, 32, 42, 68, 72, 73, 137, 138]. Для таких объектов, как ИТ-инфраструктура, оптимальное управление совокупностью взаимосвязанных объектов невозможно или связано с существенным снижением эффективности управления, а централизованное управление трудно реализуемо по причине большой размерности и трудоемкости задач управления [73]. Поэтому управление такими объектами, как ИТ-инфраструктура, осуществляется либо рациональным распределением функций управления между подсистемами СУИ с иерархической структурой (при этом допускается некоторая потеря эффективности и качества управления), либо создается централизованная многообъектная СУ, характеризующаяся большой размерностью. В последнем случае формализации легко поддаются задачи управления на верхних уровнях иерархии, в основном сводимые к

генерации макрокоманд, заданию условий, выбору моделей и методов для нижних уровней иерархии СУ, где определение управляющих воздействий сталкивается с существенной сложностью, обусловленной большой размерностью. Децентрализация на нижних уровнях управления может привести к потере управляемости и недостижению целей управления [73]. Появление высокопроизводительных компьютеров и средств эффективного сетевого взаимодействия стало побудительным мотивом для исследования и проектирования СУ элементами ИТ-инфраструктуры, распределенными как в топологическом, так и функциональном аспектах [70, 165].

СУИ позволяет увеличить эффективность управления ИТ-инфраструктурой и уменьшить затраты на содержание ИТ-подразделения путем всеобщей информатизации и автоматизации процессов управления и работы администраторов. Стандарт ISO 20000 [262, 263] подчеркивает важнейшую роль СУИ при управлении ИТ-услугами для контроля метрик, характеризующих поведение ИТ-процессов.

1.1.5. Анализ ИТ-инфраструктуры и ее элементов как объектов управления

Анализ ИТ-инфраструктуры и ее элементов позволил выявить ряд их особенностей, определяющих выбор концепции управления и основных решений, касающихся ее воплощения.

Управление ИТ-инфраструктурой в целом. Для эффективной работы компании необходимо обеспечить согласованную работу большого количества разнородных составляющих ИТ-инфраструктуры, являющейся информационно-инфраструктурной основой бизнеса. Для автоматизации управления ИТ-инфраструктурой требуется решение проблемы перехода от управления сетями, компьютерными системами, приложениями и ИКТ к комплексному, процессно-ориентированному и оперативному управлению ИТ-инфраструктурой.

Создание системы интегрированного управления, охватывающей все четыре уровня ИТ-инфраструктуры, требует решения множества сложных задач.

При управлении ИТ-инфраструктурой в целом необходимо решить задачи, которые не ограничиваются следующим перечнем:

- декомпозиция бизнес-процессов, определение необходимых для их поддержки сервисов, типов и объемов ресурсов;
- сведение метрик для оперативной оценки качества функционирования бизнес-процессов;
- обеспечение надежного и эффективного функционирования ИТ-инфраструктуры;
- интегрированное управление всеми приложениями и сервисами и функционированием ИТ-инфраструктуры, адаптируемое к реальной ситуации в ИТ-инфраструктуре и учитывающее значимости приложений, доступность и объем общих ресурсов в текущий период времени, сложившуюся ситуацию в телекоммуникационной сети (ТКС) и другие факторы;
- обеспечение взаимодействия идеологически и технологически разрозненных СУ отдельными приложениями, а также ТКС;
- централизация управления с обеспечением взаимодействия между отдельными СУ;
- поддержание и пополнение базы знаний СУИ, необходимых для выполнения функций управления;

• сбор и хранение описаний элементов ИТ-инфраструктуры;

• определение приоритетов приложений и процессов;

• анализ влияния сбоев в ИТ-инфраструктуре на качество ИТ-услуг и эффективность выполнения бизнес-процессов;

• управление качеством услуг посредством организации сквозного процесса управления ИТ-инфраструктурой от сбора информации о функционировании объектов управления первого и второго уровней, до перераспределения ресурсов для обеспечения требуемого качества предоставляемых услуг;

• установление корреляции событий, происходящих в ИТ-инфраструктуре;

• анализ влияния сбоев на первых трех уровнях на качество ИТ-услуг четвертого уровня для бизнес-процессов;

• прогноз параметров функционирования ИТ-инфраструктурой и качества предоставляемых ИТ-услуг;

• управление устранением неисправностей;

• моделирование и анализ работы ИТ-инфраструктуры;

• автоматизация управления ИТ-инфраструктурой в целом;

• автоматизация процессов внесения изменений с сохранением правил доступа к ресурсам, которые должны быть сделаны для достижения заданного качества предоставляемых ИТ-услуг;

• генерация отчетов о функционировании ИТ-инфраструктуры;

• управление закупками, ремонтом, обновлением ПО и аппаратных средств;

• управление взаимоотношениями с поставщиками;

• управление модернизацией ИТ-инфраструктуры с решением задач автоматизации принятия решений по модернизации технического и программного обеспечения ИТ-инфраструктуры с учетом информации о прогрессивных ИТ, данных о производителях и поставщиках, о сравнительных технических и экономических характеристиках решений;

• автоматизация процессов контроля, оптимизации и управления поставкой и монтажом нового технического и программного обеспечения.

При управлении ИТ-инфраструктурой в целом используются такие виды и методы управления:

• автоматизированное и ручное управление;

• генерирование политик управления;

• рациональное управление;

• управление изменениями и конфигурациями.

Например, путем генерирования политик здесь можно осуществлять управление функциональными подсистемами, а посредством изменения приоритетов можно осуществлять управление качеством выполнения бизнес-процессов, наделяя некоторые из них привилегированными правами при доступе к информационно-вычислительным ресурсам.

Ведущие производители ПО, такие как Microsoft, Hewlett-Packard, IBM, предлагают проприетарные СУ отдельными уровнями ИТ-инфраструктуры. Практически не проработаны вопросы интегрированного управления сетями, серверами, приложениями и сервисами с учетом значимости бизнес-процессов. В теории и на практике хорошо проработаны вопросы управления ИТ-подразделением как бизнес-подразделением с предложением большого количества метрик для оценки эффективности работы ИТ-подразделения [10].

Отсутствуют работы, посвященные вопросам управления всеми уровнями ИТ-инфраструктуры в комплексе. Не существует универсального решения или ПО управления всем комплексом задач управления ИТ-инфраструктуры, также нет инструментария, осуществляющего интеграцию механизмов и средств управления отдельными уровнями.

Объекты управления четвертого уровня ИТ-инфраструктуры. Объектами управления IV-го уровня являются сервисы, предоставляемые ИТ-инфраструктурой пользователям.

Выше отмечено, что эти сервисы разделены на бизнес-сервисы и технологические сервисы. Бизнес-сервисы обеспечиваются распределенными приложениями и зависят от специфики бизнеса предприятия. К технологическим (универсальным) сервисам отнесены интернет-сервисы, а также телекоммуникационные сервисы, используемые внутри любого предприятия, независимо от характера бизнеса.

Из-за существенных различий функций, выполняемых сервисами этого уровня, невозможно создание единого метода мониторинга их работоспособности и управления качеством их предоставления. Особенности каждого из сервисов и уникальность набора свойств и параметров качества затрудняют их классификацию для последующего выбора методов управления.

При управлении на уровне сервисов решаются такие задачи:

- декомпозиции бизнес-процесса на операции, определения необходимых услуг;
- согласования уровня услуг и определения объемов ресурсов, необходимых для обеспечения согласованного качества услуг;
- ведения каталога сервисов с описанием услуг и порядка их использования;
- обработки запросов на ИТ-услуги;
- управления введением новых услуг;
- управления расчетами;
- автоматизации поддержки пользователей.

Управление бизнес-сервисами подразумевает:

- поддержание качества услуг на заданном уровне;
- моделирование структуры предоставления сервиса;
- мониторинг качества предоставления сервисов;
- управление вычислительными процессами отдельной задачи и получение информации о ее статусе.

При управлении бизнес-сервисами могут быть использованы следующие виды и методы управления:

- компенсационное управление;
- выделение дополнительных ресурсов приложениям, предоставляющим услуги;
- увеличение или ограничение полосы пропускания приложениям или сервисам;
- управление системными функциями;
- активация дополнительных программных агентов.

В тоже время при управлении технологическими сервисами используются:

- механизмы административного управления:
- установка контролирующего ПО;
- анализ сетевого трафика;

• приоритезация трафика;
• увеличение полосы пропускания в сегменте сети;
• управление правами доступа пользователей.

При управлении технологическими сервисами решаются такие задачи:
• мониторинг трафика технологических сервисов;
• обеспечение высококачественной передачи голосовых сообщений;
• контроль перегрузок в сети;
• контроль параметров передачи данных (производительности, доступности, скорости передачи, времени отклика, загруженности и др.);
• добавление новых пользователей;
• управление доступом к ресурсам и сервисам;
• управление учетом, включая [49, 285] информирование пользователей о начисленных стоимостях или использованных ресурсах; установка предельных учетных значений и распределение тарифов за использование ресурсов; комбинирование стоимостей при привлечении многих ресурсов.

Обычно технологические сервисы разделяют на интерактивные, оперативного и отложенного прочтения.

Наиболее распространенными и универсальными являются сервисы, которые можно отнести к группе сервисов отложенного прочтения, отличающиеся достаточно высокой требовательностью к ресурсам серверов и ТКС. Характерной особенностью этих сервисов является то, что запрос и получение информации существенно разнесены во времени. Примером такого типа сервисов является электронная почта – самый распространенный интернет-сервис. Для электронной почты используются протоколы SMTP, POP3, IMAP или их улучшенные аналоги SMTPS, POP3S, IMAPS с криптографическими протоколами для защиты трафика от нежелательного прослушивания, шифрования линии связи между пользователем и почтовым сервером.

Задачи, которые решаются при управлении электронной почтой: создание учетной записи; фильтрация электронной почты для удаления нежелательной корреспонденции и вирусов; управление полномочиями; распределение ресурсов электронной почты по нескольким физическим серверам; обеспечение надежности хранения данных путем синхронизации; предотвращение перегрузки серверов электронной почты.

Основные задачи мониторинга электронной почты: обнаружение и предотвращение несанкционированного доступа; выявление массовых рассылок рекламного и вирусного характера; мониторинг нагрузки серверов; мониторинг трафика электронной почты путем передачи поступающей почты контролирующей программе, прослушивания и перехвата сетевого трафика.

Другим распространенным технологическим сервисом является VoIP – технология передачи голосовых сигналов в IP-сетях, позволяющая создавать системы корпоративной или офисной IP-телефонии без значительных финансовых затрат с предоставлением большого количества сервисных функций.

При управлении VoIP осуществляется: настройка и управление ресурсами сети для работы с голосовыми вызовами; контроль прохождения голосовых сообщений по разным маршрутам; управление трафиком; определение нагрузок в сети и решение проблем, например, увеличением полосы пропускания в конкретном сегменте сети; обработка прерванных вызовов; контроль параметров, влия-

ющих на качество голосового обмена; оценка качества передачи данных; составление отчетов с данными анализа схем вызовов, обзора трафика и пр. Задачи мониторинга VoIP: получение подробной информации о каждом вызове; контроль загруженности сети для контроля качества передачи голоса; накопление данных истории производительности для анализа причин ухудшения качества сервиса; отслеживание изменений параметров, влияющих на качество передачи данных; фиксация качества текущих разговоров.

Сервис FTP обеспечивает обмен файлами между компьютерами, работающими в сетях TCP/IP. Управление сервисом FTP подразумевает распределение прав доступа к файлам; учет и контроль трафика по пользователям, группам, интерфейсам, блокам адресов, времени и др. Задачи мониторинга FTP: получение информации об отдельном соединении; составление отчетов о производительности; проверка готовности портов.

Объекты управления третьего уровня ИТ-инфраструктуры. Объектами управления третьего уровня являются распределенные приложения, обеспечивающие поддержку бизнес-процессов и предоставление ИТ-услуг пользователям.

Под управлением распределенными приложениями, как правило, понимают мониторинг использования приложениями сетевых и вычислительных ресурсов и изменение параметров работы этих приложений для достижения наибольшей эффективности использования ресурсов или эффективности функционирования приложений.

При управлении распределенными приложениями решаются такие задачи:

- поддержание работоспособности приложений и сервисов в соответствии с заданным регламентом;
- выделение, учет, мониторинг и перераспределение ресурсов, необходимых для эффективного функционирования приложений;
- распределение задач по доступным вычислительным ресурсам;
- достижение наибольшей эффективности использования ресурсов или эффективности функционирования приложений;
- поддержание требуемого качества обслуживания;
- управление распределенными приложениями за счет изменения конфигурации, запуска, перезапуска или останова приложений и пр.;
- управление доступом пользователей к приложениям, правами пользователей, идентификацией и аутентификацией;
- обеспечение взаимодействия идеологически и технологически разрозненных систем и средств управления отдельными приложениями;
- мониторинг использования приложениями сетевых ресурсов;
- управление безопасностью.

При управлении приложениями используются такие виды и методы управления:

- подписка и публикация;
- создание управляемых событиями присоединенных процедур для выполнения предопределенных действий;
- синхронный удаленный вызов процедур;
- асинхронный запрос/ответ;
- переадресация запросов;
- обмен сообщениями;
- организация очередей;

- диалоги;
- отслеживание транзакций;
- изменение параметров работы;
- изменение конфигурации;
- увеличение или ограничение полосы пропускания;
- выделение дополнительных ресурсов;
- запуск, перезапуск или останов приложений.

Эффективность $E_i,\ i=\overline{1,N}$ функционирования i-го распределенного приложения A_i, где N – количество приложений, можно оценить по среднему времени отклика Δt_i серверной части приложения A_i на запрос пользователя, измеренного на стороне пользователя, или по максимальному количеству запросов $n_{\max i}$, обслуживаемых серверной частью приложения A_i в единицу времени.

Можно выделить два способа повышения эффективности $E_i,\ i=\overline{1,N},$ функционирования распределенного приложения $A_i,\ i=\overline{1,N}$:

- выделение приложению $A_i,\ i=\overline{1,N}$ дополнительных вычислительных ресурсов;
- обеспечение приоритетной передачи данных, относящихся к приложению $A_i,\ i=\overline{1,N},$ по ТКС.

Первый способ использует методы управления распределением и перераспределением ресурсов между приложениями и пользователями. Реализация этих методов осуществляется на втором уровне (решению задач выделения приложению $A_i,\ i=\overline{1,N},$ дополнительных ресурсов посвящена глава 7). Второй способ повышения эффективности работы приложений заключается в обеспечении приоритетного прохождения запросов и ответов приложения $A_i,\ i=\overline{1,N},$ по ТКС. Методы управления трафиком реализуются на первом уровне (см. главу 9).

Целесообразно управление приложениями осуществлять через визуальный интерфейс СУИ на базе API, позволяющий администраторам с учетом данных базы управляющей информации, содержащей данные о клиентах, серверах и приложениях, конфигурировать, отслеживать и управлять распределенными приложениями с единой консоли управления. Это позволит уменьшить стоимость эксплуатации ИТ-инфраструктуры, повысить доступность ИТ-инфраструктуры для конечных пользователей и предоставит возможность отслеживать функционирование приложений, а также изменять их конфигурацию.

Управление на третьем и четвертом уровнях часто строится по принципу «подписка и публикация», что позволяет СУИ оперативно реагировать на бизнес-события, осуществляя изменения в бизнес-операциях. При этом могут создаваться управляемые событиями присоединенные процедуры, используемые приложениями для совершения предопределенных действий.

Используются также традиционные методы, такие как синхронный удаленный вызов процедур, асинхронный запрос/ответ, переадресация запросов, обмен сообщениями; организация очередей, диалоги и другие.

Как правило, процесс управления начинается с получения сообщений о неполадках, после чего определяется место неисправности. Традиционные решения СУИ собирают много сведений о событиях в ИТ-инфраструктуре, но при этом

сложно определить приложения или транзакции, пострадавшие в результате неисправности. В масштабных ИТ-инфраструктурах, содержащих тысячи серверов, большое количество коммутаторов и БД, очень сложно определить причины невыполнения отдельной транзакции. Непрохождение транзакции легко обнаруживается на стороне пользователя, а администратору необходимо связать события в ИТ-инфраструктуре с проблемами конкретной транзакции. Пользователь должен связаться с администратором и сообщить ему о проблемах с определенной транзакцией. Решение такой задачи традиционными методами представляет достаточно трудоемкую задачу. В таких случаях весьма перспективным является применение ПО управления производительностью приложений (ApM) [232, 240], призванное обнаруживать и устранять проблемы до того, как они проявятся у конечного пользователя.

Система ApM учитывает все аспекты контроля работоспособности и эффективности предложений, в том числе следит за восприятием работы приложения конечным пользователем – QoE. В последнем случае отслеживаются транзакции и контролируется время отклика серверов приложений. Если оно абнормально, то предпринимаются действия по восстановлению эффективности работы приложения. Отслеживание отдельных транзакций позволяет уменьшить количество отслеживаемых элементов ИУС, ограничиваясь только компонентами, задействованными при выполнении транзакции. После определения задействованных элементов проводится поиск причины неполадки.

Отслеживание отдельных транзакций в ApM является большим прогрессом в управлении приложениями [232]. Такой подход позволяет существенно сократить количество серверов и сетевого оборудования, работоспособность которого анализируется при выявлении причины снижения эффективности работы бизнес-приложения. ПО ApM, собирая данные из разных источников, может установить корреляцию между компонентами ИУС и бизнес-сервисами. Высоким уровнем аналитики обладают решения от компаний HP, Netuitive, IBM, BMC и Quest Software [240].

Вопросы управления на третьем уровне вследствие новизны решаемых задач проработаны недостаточно глубоко. Так, перспективное управление производительностью приложений (системы ApM) находится в зачаточном состоянии и на рынке представлено лишь незначительное количество ПО с ограниченным функционалом.

В настоящее время не существует готового универсального решения, покрывающего весь спектр задач управления распределенными приложениями в ИТ-инфраструктуре. Это объясняется, прежде всего, отсутствием принятого ИТ-сообществом стандарта управления приложениями и стандарта на протокол информационного обмена с распределенными приложениями, подобного SNMP или RMON для технических средств. Быстрый выход из этого положения, например, компания Microsoft видит в создании MIB для приложений, что позволяет программным агентам работать с приложениями, как с физическими устройствами. В этом случае, например, DHCP- и WINS-сервера, работающие под ОС Windows NT, становятся SNMP-управляемыми, однако управление возможно только серверной составляющей распределенного приложения.

Объекты управления второго уровня ИТ-инфраструктуры. Повысить эффективность выполнения бизнес-процессов можно выделением дополнительных вычислительных ресурсов критичным приложениям. Реализация управления, связанная с перераспределением ресурсов между приложениями, осуществляется на

уровне вычислительных ресурсов. Модели и методы распределения ресурсов при дефиците и излишке ресурсов рассмотрены в работах [114, 147, 156, 160].

При управлении вычислительными ресурсами решаются такие задачи:

- учет всех ресурсов (программно-технического обеспечения серверов, включая информацию о технических характеристиках и учетных параметрах), а также их администраторов и пользователей (как людей, так и приложений);
- управление производительностью и доступностью ресурсов;
- мониторинг состояния распределенных ресурсов, сбор статистики;
- мониторинг производительности серверов, систем хранения и сбор статистики, генерация отчетов о функционировании ресурсов;
- анализ работы ресурсов;
- администрирование и управление отдельными ресурсами, их комбинацией, а также рассредоточенными ресурсами;
- контроль доступа;
- распределение и перераспределение ресурсов без перезагрузки системы;
- управление виртуализацией;
- учет использования ресурсов и расчет финансовых характеристик, например, вычисление стоимости хранения данных;
- генерация отчетов о функционировании ресурсов;
- управление распределением ресурсов;
- управление инициализацией и поддержанием параметров ресурсов, топологией, трафиком, маршрутизацией, переключением на резерв;
- управление устранением неисправностей;
- проведение технического обслуживания.

На этом уровне могут быть использованы такие виды и методы управления:

- автоматическое, автоматизированное или ручное управление;
- управление конфигурациями;
- механизмы административного управления;
- управление по результатам проверки превышения пороговых значений контрольных показателей, например, коэффициента загруженности сервера или времени отклика на запросы пользователя;
- настройка системных служб.

При ручном управлении администраторы обычно управляют вычислительными ресурсами, контролируя превышение пороговых значений таких показателей, как коэффициент загруженности сервера или время отклика на запросы конечного пользователя. Выделение дополнительных ресурсов распределенным приложениям может проводиться по графику планируемых мероприятий, например, обработка бухгалтерией платежных ведомостей в конце месяца. В результате такого ручного управления предприятие может сократить затраты на управление ИТ-инфраструктурой, при этом пользователи выигрывают от временно повысившейся производительности ИУС. Естественно, что автоматизация этого процесса позволит существенно повысить эффективность функционирования бизнес-приложений при одновременном повышении эффективности использования вычислительных ресурсов.

Вопросы управления элементами ИТ-инфраструктуры на втором уровне проработаны достаточно глубоко. Для управления объектами второго уровня существует большое количество как специализированных, так и универсальных решений управления.

АУС КП

Для мониторинга и управления серверами производители оборудования выпускают проприетарное ПО. Лидируют дорогостоящие и ресурсоемкие решения Microsoft, Hewlett-Packard, IBM. Универсальные и СУ, использующие проприетарное ПО (например, решения HP Operations Manager, IBM Tivoli, Microsoft SCOM), позволяют накапливать данные о значениях большого количества разнообразных параметров, относящихся к работе серверов, однако функции анализа данных сильно ограничены и практически отсутствуют средства анализа функционирования совокупности серверов. Недостаточно детально проработан вопрос управления ограниченными ресурсами ИТ-инфраструктуры с учетом важности бизнес-процессов, а также эффективного управления ресурсами ЦОД.

Объекты управления первого иерархического уровня ИТ-инфраструктуры. Объектами управления этого уровня являются сетевые элементы и коммуникационные технологии.

Один из основных комплексов задач, решаемых ИТ УИТИ, связан с управлением сетевыми ресурсами, поскольку от работоспособности сети существенно зависит эффективность функционирования всей ИУС. Вопросы сетевого управления давно известны и хорошо проработаны вплоть до создания систем автоматического управления некоторыми сетями на канальном и физическом уровнях модели OSI. Имеется множество универсальных и специализированных решений управления сетями, кроме того, существуют встроенные в отдельные технологии системы автоматического управления.

Существует три семейства руководящих материалов по вопросам управления в области телекоммуникаций: стандарты международных организаций ISO и ITU-T; рекомендации Internet; документы промышленных, профессиональных или административных организаций и консорциумов, участвующих в разработке стандартов в области сетевого управления, таких как IEEE, TM Forum, OSF и других.

Стандарты ISO и их аналоги в ITU-T являются развитием принципов, заложенных в основу OSI ISO. Стандарты этого семейства охватывают обширную сферу вопросов, связанных с управлением в области телекоммуникаций – от описания информационных моделей объектов сетевого управления и управляющих протоколов до принципов построения сетей управления телекоммуникациями. Так, в документах серии M ITU-T подробно разработаны положения модели TMN. Протоколы управления этого семейства, например CMIP, характеризуются большой сложностью, высокой стоимостью внедрения, но благодаря своей функциональности обеспечивают качественное управление. Протокол ICMP [187, 198, 320] предназначен для обнаружения ошибок в сетях и передачи информации о таких ошибках, в то время как применение запросов ICMP для определения доступности устройств, подключенных к сети, ограничивается нормативными документами Internet [199, 200].

Управление в Internet базируется на использовании протокола SNMP [210, 290], который разрабатывался для применения в сетях TCP/IP, но из-за простоты и широкого внедрения в оборудование различных производителей стал международным стандартом де-факто. Данные, представленные в соответствии с языком ASN.1 [45, 46, 47, 48] абстрактного описания управляемых объектов и блоков данных протокола, используются для управления сетевыми объектами. Протоколы управления этого семейства могут быть просто и быстро использованы, но не позволяют достичь высокого качества управления.

При управлении на уровне сетевого взаимодействия решаются следующие задачи:

- управление производительностью (рабочими характеристиками), включая:
  - – сбор и анализ статистических данных о функционировании сети;
  - – ведение и анализ журналов регистрации ошибок;
  - – оценка эффективности использования сетевых ресурсов;
  - – выявление «узких мест» сети;
  - – анализ сетевых протоколов;
  - – планирование развития сети;
  - – определение эксплуатационных характеристик системы в обычных и искусственных условиях;
  - – изменение режимов работы системы с целью выполнения мероприятий по управлению эксплуатационными характеристиками;
- управление устранением неисправностей [49, 285]:
  - – ведение и анализ журналов регистрации ошибок;
  - – прием уведомлений об обнаружении ошибок и выполнение соответствующих действий;
  - – отслеживание и идентификация неисправностей;
  - – выполнение последовательностей диагностических тестов;
  - – устранение неисправностей;
- управление конфигурацией [49, 285]:
  - – установка параметров, управляющих стандартными операциями;
  - – логическая увязка имени с администрируемыми объектами и наборами администрируемых объектов;
  - – активизация и деактивация администрируемых объектов;
  - – сбор информации по требованию о текущем состоянии элементов;
  - – изменение конфигурации;
- управление безопасностью [49, 285]:
  - – создание и удаление услуг и механизмов защиты, контроль за их действиями;
  - – распределение относящейся к защите информации;
  - – отчетность о событиях, относящихся к защите информации;
  - – управление доступом и полномочиями пользователей;
  - – контроль и управление межсетевыми взаимодействиями;
  - – защита от несанкционированного доступа извне;
  - – обнаружение и устранение вирусов;
- автоматическое построение топологии сети;
- мониторинг событий и определение работоспособности сетевых устройств;
- управление сетевыми ресурсами для повышения надежности работы, уменьшения времени простоя, улучшения производительности корпоративной сети;
- анализ трафика;
- учет загруженности каналов связи с фиксацией перегрузок сети;
- сбор и анализ статистики использования ресурсов ТКС;
- генерация отчетов;
- управление пользовательским трафиком.

На этом уровне часто используются следующие виды и методы управления:

- изменение конфигураций;
- управление маршрутизацией;

АУЦ КП

- управление трафиком;
- механизмы управления потоками: регулирование возможности установления и количества логических соединений; управление использованием полосы пропускания; управление скоростью поступления в сеть данных от пользователей и приложений; блокирование работы пользователей, приложений и пр.;
- управление приоритетами пользователей, приложений и сервисов;
- управление по результатам моделирования;
- механизмы административного управления.

Для решения задач сетевого управления существует большое количество решений как с открытым кодом, так и проприетарных, например, Nagios, Spectrum от CA, Operations Manager от Hewlett-Packard, Tivoli от IBM и др.

Концепция TMN выделяет пять задач сетевого управления [320], которые вписываются в функции группы управления предоставлением и производительностью ИТ-сервисов. Этими задачами являются управления: производительностью; устранением неисправностей; конфигурацией, расчетами и безопасностью. К продуктам, в той или иной степени реализующим все пять задач управления на первом уровне ИТ-инфраструктуры, относятся Spectrum от CA, Operations Manager от Hewlett-Packard, Tivoli от IBM и др. Существующие зарубежные системы сетевого управления, а также системы, которые производители позиционируют как системы интегрированного управления сетями, ИТ-услугами и бизнес-процессами, характеризуются высокой стоимостью, которая очень часто не компенсируется эффектом от внедрения, закрытостью и отсутствием методик или рекомендаций для определения параметров настройки. Поэтому продолжается активная разработка моделей, методов и ПО управления первом уровнем.

Большинство СУ сетями ориентированы на управление только оборудованием с целью поддержания эксплуатационных характеристик и показателей работы сети на заданном уровне. При этом управление сетями осуществляется без учета значимости передаваемых данных для поддержания бизнес-процессов. Это приводит к тому, что при перегрузке сети, в которой не задействованы механизмы QoS, ухудшаются параметры передачи всех потоков данных, независимо от их важности.

Управление сетевым трафиком – одна из важных задач сетевого управления. В [247] рассмотрена возможность снижения скорости передачи данных с помощью передачи специально подготовленного потока ICMP. Существует большое количество ПО, осуществляющего мониторинг и управление локальными и глобальными сетями, в том числе управление трафиком. Однако существующие продукты не решают задачу управления потоками данных с учетом значимости бизнес-процессов.

Управление ИТ-подразделением. Эффективное функционирование ИТ-инфраструктурой не может быть осуществлено без эффективного управления ИТ-подразделением. При управлении ИТ-подразделением решаются следующие задачи:

- автоматизированное управление ИТ-инфраструктурой;
- обработка запросов на ИТ-услуги;
- управление введением новых услуг;
- управление расчетами;
- поддержка пользователей и пр.

Объектом управления в данном случае является само ИТ-подразделение и его руководство.

1.1.6. Контуры управления информационно-управляющей системой

С точки зрения управления ИТ-инфраструктурой можно выделить три контура управления – внешний, внутренний и оперативный (обозначены овалами на рис. 1.3). Внешний контур, относящийся к управлению информационно-управляющей системой, осуществляет проектирование и внедрение бизнес-процессов с последующим контролем эффективности их выполнения. Внутренний контур обеспечивает отображение бизнес-процессов на ИТ-услуги с определением целевых значений, фиксируемых в соглашении об уровне услуг (SLA), и контролем качества ИТ-услуг. Декомпозиция ИТ-сервисов определяет задачи оперативного контура, которые сводятся к непрерывному поддержанию уровня услуг на зафиксированном в SLA уровне с минимальными затратами.

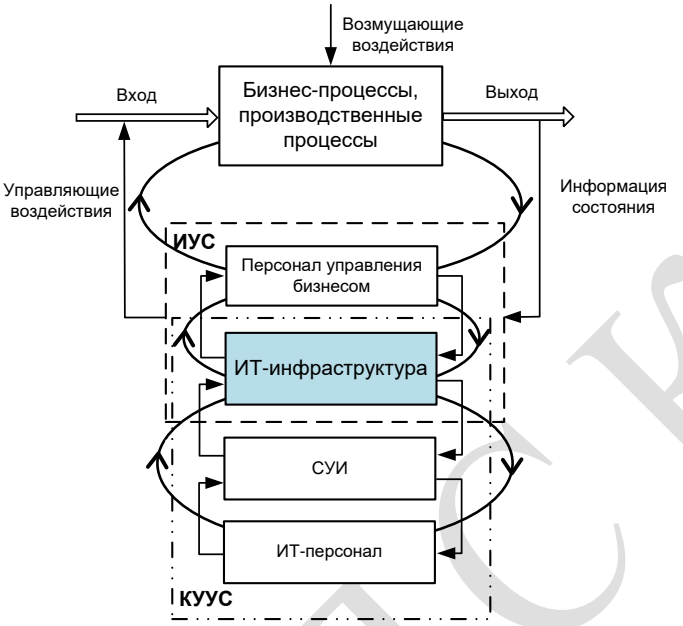


Рис. 1.3. Контуры управления в ИУС

При принятии решений по управлению бизнес-процессами (управляющими, операционными и поддерживающими) или производственными процессами используют критерии эффективности бизнеса $\mathcal{Y}$.

Информационно-управляющая система (ИУС) – это система, в которой объединяются персонал управления бизнесом (основной и вспомогательной деятельностью) и ИТ-инфраструктура. ИУС обеспечивает поддержку принятия решений, обеспечивающих выполнение критерия $\mathcal{Y}$.

В тоже время контур управления уровнем услуг (КУУС), охватывающий ИТ-инфраструктуру, персонал поддержки ИТ-инфраструктуры (ИТ-персонал), а также технические, программные и информационные средства такой поддержки в виде СУИ, обеспечивает предоставление бизнес-подразделениям ИТ-услуг с согласованным уровнем качества при рациональном использовании ресурсов ИТ-инфраструктуры. Поэтому управление ИТ-инфраструктурой нельзя рассматривать в отрыве от управления бизнесом, а при повышении эффективности функционирования ИУС с ориентацией на поддержание согласованного уровня ИТ-услуг главная роль отводится ИТ УИТИ.

Для эффективного ведения бизнеса должна быть сформирована и скоординирована целостная система процессов всех трех контуров управления с выделением в контурах следующих процессов:

1) во внешнем: прогнозирование → планирование развития → учет → оценивание по критерию доходность/качество → контроль → корректирование планов развития;

2) во внутреннем: планирование → выполнение → проверка → действия по выполнению плана (цикл Деминга);

3) в оперативном: мониторинг → анализ состояния → управление → системная оптимизация → планирование развития [107].

Своевременность и точность выполнения вертикально и горизонтально интегрированного комплекса процессов трех контуров обеспечивает достижение целей бизнеса.

Для поддержки замкнутых контуров управления в ИТ УИТИ операционное управление ИТ-инфраструктурой интегрируется с координированным взаимодействием компонентов ИУС [71]. Координация содействует достижению цели функционирования ИТ УИТИ как составляющей ИУС. При этом должны обеспечиваться координированное взаимодействие и обмен информацией между контурами управления с соответствующим обобщением информации в каждом из них для принятия решений. Такое переплетение операционного и координационного аспектов управления существенно усиливает роль ИТ УИТИ, в которой произошло смещение акцентов управления ИТ-инфраструктурой от поддержки ИКТ и оборудования в сторону поддержки уровня ИТ-услуг.

В оперативном контуре появляются задачи измерения, анализа, оценивания, учета, контроля, прогнозирования, планирования и управления, что порождает необходимость разработки соответствующих моделей и методов решения этих задач. Прежде, чем разрабатывать эти модели и методы, следует определиться с общим подходом к решению задач оперативного контура с учетом его комплексного характера и необходимости взаимодействия и координации с внешним и внутренним контурами управления. Ниже такой общий подход изложен в виде концепции оперативного управления ИТ-инфраструктурой.

1.1.7. Базовая модель управления ИТ-инфраструктурой и ИУС

Для управления ИТ-инфраструктурой представляются перспективными модели теории управления, исследования операций, математического программирования, которые должны задать взаимосвязи следующих параметров:

1) состав ИТ-инфраструктуры – совокупность объектов, являющихся элементами ИТ-инфраструктуры, распределенных по четырем иерархическим уровням;

2) структура ИТ-инфраструктуры – совокупность информационных, управляющих и других связей между элементами ИТ-инфраструктуры и СУИ, включая отношения подчиненности и разделение прав принятия решений в иерархической СУИ;

3) число итераций управления – однократность или многократность управляющих воздействий в течение определенного периода для достижения максимального значения эффективности управления;

4) целевые функции, функции полезности, выигрыша или предпочтения бизнеса, отображенные на ИТ-процессы, на основании которых определяются целевые функции СУИ. Управление должно осуществляться так, чтобы максимизировать значения этих целевых функций;

5) допустимые множества $\mathcal{G}$ возможных состояний элементов ИТ-инфраструктуры, их ограничения и влияние на состояния зависимых элементов, методы получения и оценки состояний элементов;

6) порядок функционирования СУИ – последовательность получения данных мониторинга, анализ состояния и выбора управления.

Спецификация параметров модели влияет на механизмы управления ИТ-инфраструктурой.

В данной работе выбран индуктивный подход к исследованию, основанный на последовательном переходе от простейших моделей к сложным. Поэтому сначала рассмотрим базовую модель управления ИТ-инфраструктурой в виде, представленном на рис. 1.4. Эта модель называется базовой, потому что она простейшая с точки зрения структуры, описания и исследования. В рамках этой модели не учитываются взаимозависимость элементов, иерархическое деление ИТ-инфраструктуры, неопределенность состояний и многие другие факторы, которые будут учтены в расширениях базовой модели. Такая модель может быть использована и при рассмотрении аспектов управления ИУС со стороны СУИ и ИТ-подразделения.

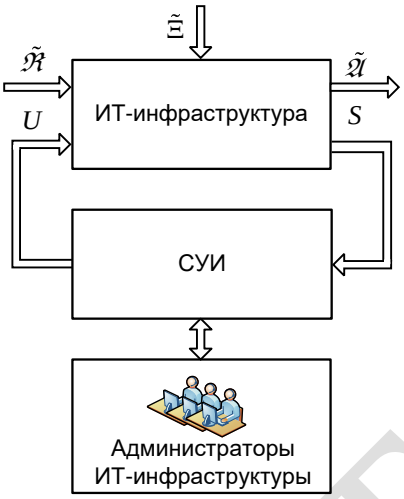


Рис. 1.4. Базовая модель управления ИТ-инфраструктурой

На примере модели, приведенной на рис. 1.4, можно выявить многие проблемы управления ИТ-инфраструктурой и отследить закономерности управления ИУС при дальнем исследовании более сложных моделей.

На вход ИТ-инфраструктуры поступает поток запросов $\tilde{\mathcal{R}}$, на который ИТ-инфраструктура реагирует потоком результатов $\tilde{\mathcal{Q}}$. На ИТ-инфраструктуру воздействуют возмущающие воздействия $\tilde{\Xi}$. В СУИ анализируется текущее состояние $S \in \mathcal{G}$ ИТ-инфраструктуры и выбирается управление $U \in \mathcal{U}$, при котором достигается максимальная эффективность управления.

Согласно с рис. 1.3 ИТ-инфраструктура является общей составляющей взаимодействующих компонентов ИУС и КУУС. Поэтому можно выделить два масштабных объекта управления – бизнес-процессы (производственные процессы) и ИТ-инфраструктура и две системы управления – ИУС и СУИ, в контурах управления которых занят персонал управления бизнесом и ИТ-инфраструктурой соответственно.

Управление ИТ-подразделением, в котором сосредоточен ИТ-персонал, подразумевает прежде всего организационные формы управления, которые хорошо проработаны в ITSM [264] по ISO/IEC 20000, при этом ИТ УИТИ предоставляет средства для автоматизации управления ИТ-подразделением.

Механизмы управления СУИ закладываются на этапе проектирования и в основном включают изменение конфигурации и самодиагностику.

Управление ИТ-инфраструктурой осуществляет ИТ-подразделение с использованием методов автоматического, автоматизированного и ручного управления.

1.1.8. Модель многоконтурной системы управления

В настоящее время автоматизация функций СУИ в основном ограничивается мониторингом и анализом функционирования ИТ-инфраструктуры, а при решении задач управления ИТ-инфраструктурой зачастую полагаются на опыт и квалификацию администраторов, деятельность которых регламентируется инструкциями и другими руководящими документами. При этом управление сводится к настройкам операционных систем и перезагрузке серверов и рабочих станций, изменению конфигурации активного сетевого оборудования и пр.

Эффективность управления можно существенно повысить, внедряя в СУИ модели и методы теории автоматического управления (ТАУ) [7]. Это даст возможность строить замкнутые системы управления ИТ-инфраструктурой с обратной связью, что не только повысит оперативность и качество управления, но и позволит минимизировать влияние человеческого фактора [104].

Для поддержания значений параметров функционирования ИТ-инфраструктуры в заданных пределах и решения множества других задач целесообразно представлять СУИ в виде многоконтурной замкнутой системы, каждый контур которой управляет отдельными аспектами функционирования ИТ-инфраструктуры. Замкнутые СУИ с обратной связью в соответствии с ТАУ [7] обладают высоким быстродействием и качеством управления.

В СУИ может быть организовано множество контуров управления, каждый из которых решает отдельный спектр задач управления. Это могут быть задачи управления, ориентированные на сохранение работоспособности ИТ-инфраструктуры поддерживающие качества функционирования элементов и подсистем ИТ-инфраструктуры на заданном уровне, а также управление надежностью [125], устранением неисправностей и пр., оборудованием, доступом к ограниченным ресурсам [114, 149, 157] и множество других задач.

Для создания замкнутых контуров управления необходимо получить формальные модели управления, проверить их адекватность и провести исследование, разработать алгоритмы и проверить их работоспособность, найти эффективные методы определения числовых показателей для оценки качества функционирования элементов ИТ-инфраструктуры [104].

Замкнутая многоконтурная (многомерная) СУИ, в соответствии с ТАУ [7], может иметь вид, как на рис. 1.5. На вход системы воздействует пространство входных величин (задающих воздействий) $\mathcal{G} = \{G_N\}$, $n = \overline{1, N}$, где N – общее количество контуров управления, каждый элемент G_n, $n = \overline{1, N}$, которого представляет собой вектор $G_n = \{g_{n,1}, g_{n,2}, ..., g_{n,Kn}\}$ задающих воздействий, элементы $g_{n,k}$, $k = \overline{1, K_n}$, которого относятся к отдельному n-му контуру управления и определяют требования к параметрам функционирования ИУС в определенном аспекте.

Пространство выходных (управляемых) величин $\mathcal{Y} = \{Y_n\}$ состоит из элементов Y_n, $n = \overline{1, N}$, каждый из которых представляет собой вектор $Y_n = \{y_{n,1}, y_{n,2}, ..., y_{n,Kn}\}$ выходных величин отдельного n-го контура управления. Например, задающие воздействия n-го контура управления могут соответствовать множеству значений величин, характеризующих качество предоставляемого сервиса ИУС и оговоренных в SLA, а на выходе присутствуют реальные значения параметров, обеспечиваемые ИУС.

Пространство рассогласования $\mathcal{X} = \{X_n\}$, полученное в результате вычитания $\mathcal{X} = \mathcal{G} - \mathcal{Y}$, воздействует на вход СУИ. Каждый элемент X_n, $n = \overline{1,N}$, пространства рассогласования $\mathcal{X}$ представляет собой вектор $X_n = \{x_{n,1}, x_{n,2}, ..., x_{n,Kn}\}$, $n = \overline{1,N}$, причем $x_{n,k} = g_{n,r} - y_{n,k}$ для всех $n = \overline{1,N}$, $k = \overline{1,K_n}$. Для решения задач автоматического регулирования СУИ должна работать таким образом, чтобы сводить к нулю рассогласование $x_{n,k}$, $n = \overline{1,N}$, $k = \overline{1,K_n}$.

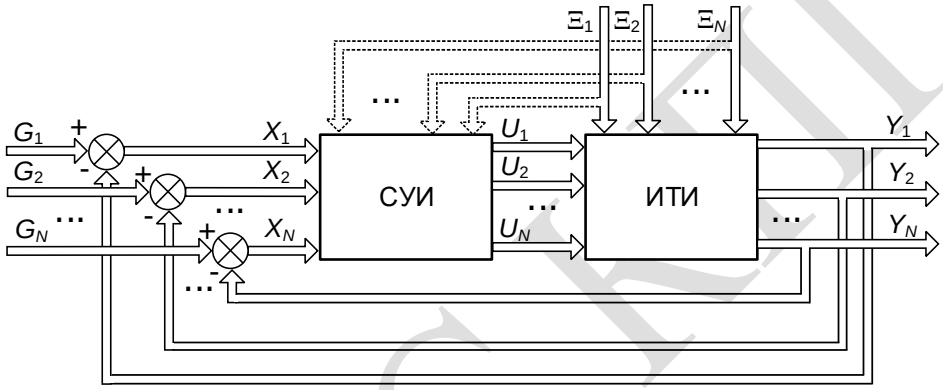


Рис. 1.5. Модель многоконтурной СУИ

СУИ вырабатывает управляющие воздействия из пространства $\mathcal{U} = \{U_n\}$, $n = \overline{1,N}$, состоящего из элементов U_n, $n = \overline{1,N}$, каждый из которых представляет собой вектор управляющих воздействий $U_n = \{u_{n,1}, u_{n,2}, ..., u_{n,Kn}\}$, $n = \overline{1,N}$, n-го контура управления. Управляющие воздействия $\mathcal{U}$ являются подмножеством $\hat{\mathcal{U}}$ команд управления (2.8).

Пространство $\Xi^0 = \{\Xi_n\}$ – пространство возмущающих воздействий на ИУС, элемент Ξ_n, $n = \overline{1,N}$, которого представляет собой вектор $\Xi_n = \{\xi_{n,1}, \xi_{n,2}, ..., \xi_{n,Kn}\}$, $n = \overline{1,N}$, возмущающих воздействий в n-м контуре управления. На рис. 1.5 пунктирными линиями обозначены возмущающие воздействия, которые могут быть оценены или измерены СУИ, следовательно, учтены при определении компенсирующего управляющего воздействия, вырабатываемого СУИ.

При практической реализации все функции, связанные с получением и измерением выходных величин и возмущающих воздействий, а также вычисление разницы, необходимой для определения рассогласования, проводятся средствами СУИ.

Каждый n-й, $n = \overline{1,N}$, контур управления предназначен для поддержания отдельных групп характеристик и качественных показателей работы ИУС на заданном уровне и характеризуется векторами задающих и выходных величин, а также специфическими возмущающими и управляющими воздействиями.

На рис. 1.6 приведена модель *n*-го контура управления, а СУИ рассматривается как *K*-мерная. Если рассматривается только *n*-й контур, то без потери общности можно считать, что $K_n = K$.

Такая модель подходит для описания большого количества задач, решаемых СУИ, например, управления доступом к ограниченным ресурсам ИТ-инфраструктуры, управления качеством функционирования подсистем ИТ-инфраструктуры, управления устранением неисправностей и др.

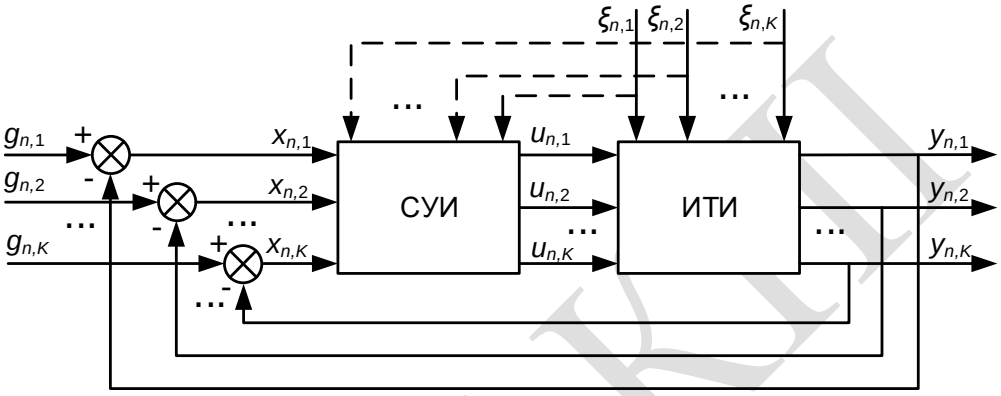


Рис. 1.6. Модель *n*-го контура управления

Если *n*-й контур (рис. 1.6) модели СУИ на рис. 1.5 используется для решения задач управления доступом к ограниченным ресурсам [106], то элементами упорядоченного множества задающих воздействий $G_n = \{g_{n,k}\}$, $k = \overline{1, K_n}$, могут быть параметры, характеризующие время обслуживания заявок приложений с максимальным приоритетом, количество заявок в очереди и пр. Возмущающими воздействиями $\Xi_n = \{\xi_{n,k}\}$, $k = \overline{1, K_n}$, является множество запросов пользователей, процессов или операций на доступ к ИТ-ресурсам. Пунктирной линией обозначены параметры возмущающих воздействий, которые могут быть измерены или оценены СУИ. Выходными величинами являются элементы упорядоченного множества $Y_n = \{y_{n,k}\}$, $k = \overline{1, K_n}$, определяющие параметры, которые соответствуют среднему времени обслуживания запросов, среднему количеству заявок, стоящих в очереди, и пр. При этом значения всех элементов упорядоченного множества Y_n могут быть измерены или вычислены. Управляющими воздействиями $U_n = \{u_{n,k}\}$, $k = \overline{1, K_n}$, являются сигналы блокирования или ограничения потока низкоприоритетных или малозначимых запросов бизнес-процессами.

Если *n*-й контур (рис. 1.6) модели СУИ на рис. 1.5 используется для управления устранением неисправностями [123], то элементами упорядоченного множества задающих воздействий $G_n = \{g_{n,k}\}$, $k = \overline{1, K_n}$, могут быть требования к работе функциональных и технологических подсистем, определенные регламентами. Возмущающими воздействиями $\xi_{n,k}$, $k = \overline{1, K_n}$, являются элементы упорядоченного множества $\Xi_n = \{\xi_{n,k}\}$, $k = \overline{1, K_n}$, соответствующие неисправ-

ностям, возникающим в компонентах ИУС. Пунктирной линией обозначены параметры неисправностей, которые могут быть зарегистрированы в СУИ. Выходными величинами являются элементы упорядоченного множества $Y_n = \{y_{n,k}\}$, $k = \overline{1, K_n}$, соответствующие текущему состоянию ФПС и ТПС. Управляющими воздействиями из упорядоченного множества $U_n = \{u_{n,k}\}$, $k = \overline{1, K_n}$, являются команды и сигналы восстановления работоспособности ИУС, поступающие к администраторам или на элементы ИТ-инфраструктуры.

При решении задачи управления функционированием ИТ-инфраструктуры необходимо обеспечить поддержание параметров функционирования подсистем, сервисов и служб в заданном диапазоне значений параметров качества функционирования.

Практическая реализация предложенных выше моделей сопряжена с большими трудностями. Поэтому в чистом виде методы ТАУ при управлении ИТ-инфраструктурой практически не применялись, поскольку в основном осуществляется ручное управление с автоматизацией некоторых процессов. Несмотря на наличие развитой теории автоматического управления автоматическая обратная связь при управлении ИТ-инфраструктурой и ее элементами не применяется, а для большинства специалистов в области ИТ-управления выглядит революционно. Перспективность использования классических моделей и методов ТАУ при управлении ИТ-инфраструктурой, делает актуальной задачу применения ТАУ для решения ряда задач управления ИТ-инфраструктурой.

1.1.9. Общая постановка задачи управления ИТ-инфраструктурой

Задачу управления ИТ-инфраструктурой в общем виде можно определить следующим образом.

Пусть состояние ИТ-инфраструктуры описывается переменной $S \in \mathcal{G}$, принадлежащей множеству $\mathcal{G}$ возможных состояний. Состояние ИТ-инфраструктуры в некоторый момент времени зависит от управляющих воздействий:

$$U \in \mathcal{U} : S = F(U). \tag{1.1}$$

Предположим, что на множестве $\mathcal{U} \times \mathcal{G}$ задан функционал $\Phi(U, S)$, определяющий эффективность функционирования ИТ-инфраструктуры. Величину

$$\mathrm{K}(U) = \Phi(U, F(U)) \tag{1.2}$$

назовем эффективностью управления $U \in \mathcal{U}$. Тогда задача СУИ заключается в выборе такого допустимого управления, которое максимизировало бы значение эффективности управления при условии, что известна реакция ИТ-инфраструктуры (1.1) на управляющие воздействия:

$$\mathrm{K}(U) \to \max_{U \in \mathcal{U}}. \tag{1.3}$$

Эффективность функционирования больших, многокомпонентных и сложных систем, какими являются ИТ-инфраструктуры, прежде всего, определяется текущим состоянием $S \in \mathcal{G}$. Поэтому в моделях управления такими системами управляющие воздействия в явном виде не входят в функционал эффективности.

АУТСОРС

В данном случае функционал эффективности $\Phi(U,S)$ рассматривается как отражающий влияние СУИ на эффективность функционирования ИТ-инфраструктуры.

Для технических систем, характеризующихся детерминизмом реакции на управляющие воздействия, зависимость $S = F(U)$ является моделью системы – управляемого объекта, отражающей законы и ограничения ее функционирования. В таком случае управляющие воздействия СУИ должны поддерживать штатный режим функционирования ИТ-инфраструктуры, когда значения параметров, определяющих состояние элементов и подсистем ИТ-инфраструктуры, соответствуют утвержденному регламенту. При этом относительно ряда услуг можно утверждать, что если ИТ-инфраструктура функционирует в штатном режиме, то эти услуги предоставляются и качество соответствует согласованному уровню.

1.1.10. Выделение функциональных областей управления корпоративной ИТ-инфраструктурой

ИТ УИТИ должна решать большое количество разнообразных задач и выполнять множество сильно различающихся функций. Все задачи и функции ИТ УИТИ можно отнести к одной из трех обобщенных функциональных областей управления ИТ-инфраструктурой, полностью покрывающих весь функционал управления (рис. 1.7):

1) поддержание качества ИТ-сервисов на согласованном уровне;

2) рациональное использование ресурсов ИТ-инфраструктуры;

3) автоматизация операционной деятельности ИТ-подразделения.

Рис. 1.7. Функциональные области управления ИТ УИТИ

При ***поддержании качества ИТ-сервисов на согласованном уровне*** для повышения удовлетворенности пользователей решаются следующие задачи:

- поддержание целевых значений параметров функционирования приложений;
- поддержание согласованного уровня услуг [331];
- повышение качества обслуживания и удовлетворенности пользователей.

Функции и задачи ***рационального использования ресурсов ИТ-инфраструктуры:***

- обеспечение максимально эффективного использования имеющихся ИТ-ресурсов при работоспособном состоянии ИУС и осуществление адаптации к возмущающим воздействиям, вызывающим перегрузки вследствие нестационарности процессов в ИУС или в результате проявления неисправностей;
- повышение надежности и доступности критически важных ИТ-ресурсов;
- распределение и перераспределение ресурсов ИТ-инфраструктуры.

При ***автоматизации операционной деятельности ИТ-подразделения*** решаются следующие задачи и выполняются такие функции:

АУТСКП

• повышение эффективности работы предприятия за счет нацеленности ИУС на потребности бизнеса и улучшения взаимодействия между бизнес-структурами и ИТ-подразделением;

• обеспечение функционирования ИТ-инфраструктуры в соответствии с заданным регламентом;

• повышение производительности и эффективности управления ИУС;

• улучшение функциональности и оперативности управления функциональными и технологическими подсистемами ИУС;

• повышение надежности функционирования ИТ-инфраструктуры, ее подсистем и сети, а также надежности и доступности критически важных ИТ-ресурсов;

• улучшение функциональности и оперативности управления ИТ-инфраструктурой;

• повышение производительности работы ИТ-систем и эффективности управления ими;

• повышение оперативности и качества управления распределенными подсистемами и сетями;

• повышение уровня обслуживания пользователей функциональных и технологических подсистем;

• повышение уровня обслуживания оборудования, ПО и оперативности выявления и устранения неисправностей;

• повышение производительности труда администраторов ИУС, а также администраторов компонентов различных уровней иерархии ИТ-инфраструктуры, подсистем и телекоммуникационных сетей благодаря высокому уровню автоматизации процессов управления, улучшению взаимодействия администраторов различных подсистем и сервисов, центрального и регионального офисов;

• улучшение взаимодействия между администраторами.

1.2. АНАЛИЗ ТЕНДЕНЦИЙ РАЗВИТИЯ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ТЕХНОЛОГИЯМИ И ИТ-ИНФРАСТРУКТУРОЙ

Управление ИТ-инфраструктурой нельзя рассматривать в отрыве от управления ИТ и управления ИУС. Возрастающая сложность управления ИТ и ИТ-инфраструктурой сопровождающаяся увеличением затрат на операционную деятельность ИТ-подразделения, делает актуальной задачу поиска новых подходов к управлению ИТ, ИТ-инфраструктурой и ИУС. Поэтому данный параграф посвящен анализу основных тенденций и перспектив развития управления ИТ, опосредованно, ИТ-инфраструктурой и ИУС.

1.2.1. Факторы, влияющие на развитие ИТ

По прогнозам аналитиков до 2020 г. должны кардинально измениться отношения между бизнесом, технологиями и ИТ-организациями. Эти изменения обусловлены воздействием на развитие ИТ следующих трех факторов [258]: 1) облачные вычисления, предоставляющие технологии по принципу «как сервис» («as-a-service»); 2) уполномоченные высококвалифицированные сотрудники; 3) радикально более сложные ИТ-среды ведения бизнеса. Кроме того, ИТ-организации будут обеспечивать повышение полезности и гибкости ИТ-услуг для увеличения доходов и удовлетворенности пользователей, при этом постоянно

снижая затраты на ИТ-услуги. Кроме того, изменится роль ИТ-организаций, которые из поставщика ИТ-услуг превратятся в посредника, предлагающего общие и частные услуги.

Рассмотрим более подробно влияние этих трех факторов.

1. **Облачные вычисления.** Станет нормой парадигма «как услуга», которая будет предоставлять ИТ-решения, готовые для использования бизнесом, кроме того, будут широко внедряться технологии самообслуживания клиентов. Прогнозируемые расходы предприятий на SPI возрастут примерно от $28 млрд в 2012 г. до $258 млрд в 2020 г. и составят примерно 45% от общих затрат на ИТ-услуги. Перейдя на активное использование облачных технологий, можно существенно сократить ИТ-бюджет, разгрузить ИТ-специалистов от выполнения рутинных работ, легально использовать большое количество продуктов без платы за лицензии, оплачивая только использованные услуги. Кроме того, существенное сокращение времени выхода на высокопроизводительный режим предоставления ИТ-услуг и быстрое последующее совершенствование ИТ-сервисов склоняют бизнес к широкому использованию в области ИТ облачных технологий [305, 362, 364].

В настоящее время ИТ-подразделения стремятся перестроить корпоративные ИТ-инфраструктуры и организовать свою деятельность так, чтобы функционировать по принципу внутреннего сервис-провайдера ИТ-услуг. Сочетание виртуализации, автоматизации и самонастраивающихся технологий рассматривается ИТ-подразделением в качестве основы предоставления IaaS для улучшения адаптируемости, гибкости и снижения затрат. Однако несмотря на доступность технологий и передового опыта Forrester считает, что только 5% ИТ-организаций имеют достаточно виртуализации, стандартизации, автоматизации и уровня самообслуживания пользователей, чтобы добиться успеха в частных облаках или IaaS [258].

2. Привлечение к управлению ИТ и ИТ-инфраструктурами уполномоченных высококвалифицированных специалистов. К работе с технологиями заказчика будут активно привлекаться независимые технически грамотные специалисты, самостоятельно принимающие решения. Это проще и дешевле, чем содержать штат собственных сотрудников, занимающихся разработкой и обслуживанием ИТ. К 2020 г. количество привлекаемых уполномоченных специалистов составит около 45% штата специалистов ИТ-отдела. Уже в настоящее время 34% таких специалистов считают, что у себя дома они имеют оборудование и технологии лучше, чем на работе, а 48% заявили, что приобретенные ими за собственные средства смартфоны они используют для рабочих целей [258].

3. Радикально более сложные ИТ-среды ведения бизнеса. Постоянно растущая квалификация бизнес-пользователей в сфере ИТ, жесткая конкуренция в бизнесе, повсеместное внедрение ИТ во все сферы человеческой деятельности, позволяет создать и использовать для каждого рода деятельности более сложную бизнес-среду. Расширение рынка ИТ-услуг ожидается в основном за счет появления новых клиентов, которые не смогут позволить себе приобретение продуктов, ориентированных на крупный и средний бизнес. Это потребует инноваций от ИТ-организаций для создания и предложения качественных ИТ-услуг по низким ценам [258] для удовлетворения потребностей малого бизнеса.

Кроме этих трех факторов, которые Forrester считает основными, на развитие ИТ безусловно окажут влияние множество других факторов и тенденций, к которым следует отнести: широкое использование концепции ITSM [264]: клиент-ориентированный подход; виртуализацию; стандартизацию; автоматизацию процессов управления ИТ; самообслуживание пользователей и др.

Ключевым при переходе от роли поставщика ИТ-услуг к роли посредника является клиент-ориентированный подход. В настоящее время около 70% ИТ-бюджета выделяется на поддержание текущей деятельности ИТ-подразделения, обслуживание систем и оборудования [258]. Даже при наличии средств большинство ИТ-организаций нацелены на обеспечение работоспособности ИТ-инфраструктуры, а не на поддержку клиентов и повышение их удовлетворенности. Подобно идеологии бизнеса, согласно который бизнес изначально должен быть полезным клиентам, ИТ-организации необходимо строить свою деятельность по принципу максимального удовлетворения интересов пользователей. Бизнес рассматривает и будет рассматривать ИТ как инструмент увеличения доходов и прироста клиентов, а ИТ-организации должны стремиться оправдать ожидания бизнеса.

Повсеместная виртуализация ресурсов ИТ-инфраструктуры требует совершенствования ИТ-управления виртуализацией и всей виртуальной средой. Методы и средства управления виртуализацией становятся важной частью ИТ УИТИ и интегрированных решений СУИ.

Ключевым фактором, оказывающим влияние на развитие ИТ, в ближайшие пять лет будет экономика [238]. Для сокращения расходов и повышения производительности ИТ-подразделений необходимо мыслить терминами ИТ-индустриализации, делая акцент на рационализацию ИТ-процессов и инструментария, которые должны привести к более гибкому, предсказуемому и надежному управлению ИТ-инфраструктурой с минимальной стоимостью. На сегодняшний день текущие затраты на управление ИТ-инфраструктурой связаны в основном с зарплатой ИТ-персонала.

Повысить эффективность работы ИТ-подразделения можно путем автоматизации процессов повседневной деятельности. Кроме того, автоматизация ИТ-процессов открывает новые возможности интеграции разрозненных инструментов ИТ-управления. Поэтому востребованность средств автоматизации продолжает возрастать. При этом ожидается переход от локальной автоматизации отдельных процессов к глобальной автоматизации, которая не только приведет к масштабной экономии, но и позволит перейти от использования единичных СУИ к СУИ массового производства.

В будущем повысить эффективность обслуживания пользователей и клиентов предполагается путем широкого использования систем самообслуживания (Self-Service), представляющих собой такие решения управления ИТ, которые позволяют клиенту найти ответы на вопросы, касающиеся ИТ-сервисов, или решить проблемы использования ИТ-услуг без обращения в службу поддержки. Как правило, система самообслуживания представляет собой упорядоченную по категориям базу знаний и автоматизированный поиск, доступ к которой клиенты осуществляют через веб-браузер. Обычно такие системы персонифицированы под каждого клиента. Использование систем самообслуживания экономит массу времени и средств за счет сокращения затрат на поддержку пользователей.

1.2.2. Эволюция выполнения ИТ-операций

Увеличение количества ЦОД и конкуренция между ИТ-организациями позволили предоставлять большее количество ИТ-услуг с лучшим качеством за меньшую стоимость [238]. Это в свою очередь увеличивает спрос на ИТ-услуги и количество новых бизнес-сервисов, а также повышает их сложность.

Для повышения конкурентоспособности ИТ-организации необходимо наряду с сохранением надлежащего качества обслуживания повышать производительность операционной деятельности. В то же время постоянное снижение стоимости электронных компонентов позволяет при сохранении затрат на закупку оборудования получить более производительные вычислительные и коммуникационные ресурсы, что позволит предоставлять большее количество ИТ-услуг с лучшим качеством за меньшую стоимость [238, 362]. Это также способствует активизации спроса на ИТ-услуги, которые могут быть предоставлены в сжатые сроки с низкой стоимостью, что позволит увеличить общее количество и сложность новых бизнес-сервисов, которые призваны обслуживать ИТ-сервисы. Для поддержания экстенсивного роста количества бизнес-сервисов и поддерживающих их ИТ-сервисов руководству ИТ-подразделения необходимо:

- обеспечивать повышение производительности труда при одновременном снижении удельных затрат. Причем увеличение должно измеряться не процентами, а порядками. Этого можно достичь только путем автоматизации развертывания и управления ИТ-услугами;
- сохранять и улучшать качество предоставления ИТ-услуг для обслуживания бизнес-процессов, поскольку качество бизнес-услуг оказывает непосредственное влияние на результативность и конкурентоспособность бизнеса. При этом очень важно соблюдать баланс между качеством сервисов и затратами на обслуживание [238];
- быстро и эффективно удовлетворять спрос на новые услуги. Технология борьбы за клиентов подразумевает постоянное приспособление бизнес-процессов и бизнес-услуг к спросу, быстрота реагирования на изменение спроса является ключевым конкурентным преимуществом [257]. Это требует от ИТ-подразделения способность быстро внедрять новые ИТ-услуги с сохранением целостности оптимизированной платформы предоставления бизнес-услуг, что может быть обеспечено за счет использования новых технологий, таких как виртуализация и облачные вычисления;
- обеспечивать постоянное развитие ИТ-организации. Отягощенные обслуживанием и поддержкой большого количества бизнес-услуг ИТ-организации часто не в состоянии качественно поддерживать новые услуги и удовлетворить спрос на гибкость и ожидаемое качество обслуживания. В 2009 г. 66% ИТ-директоров были обеспокоены неспособностью ИТ-подразделения адекватно поддерживать развитие бизнеса [238]. Для преодоления этой проблемы ИТ-специалисты должны уменьшить количество времени, которое тратится на ручную настройку и развертывание ИТ-услуг, автоматизируя сложные процессы, склонные к ошибкам и переделкам.

Традиционный подход в борьбе с проблемами в ИТ-инфраструктуре за счет привлечения большого количества ИТ-специалистов оказывается впредь не эффективным, так как увеличение персонала подчиняется закону убывающей отдачи. Поэтом большая численность персонала может оказаться непродуктивной и быстро натолкнуться на финансовые и организационные ограничения [238].

АУТСКИЙ

ИТ должны преодолеть все эти проблемы, повышая производительность на порядок в течение ближайших лет. Если использовать терминологию модели индустриализации, то существенное увеличение производительности выполнения ИТ-операций возможно только при переходе от единичного производства к серийному или массовому производству [238]. По словам Ч. Перроу, «производительность в большей степени зависит от масштаба технологических изменений и экономики, чем от человеческих усилий» [316]. Поэтому ключевым для повышения производительности в сфере ИТ является переход на модель массового производства путем оптимизации цепочки предоставления сервисов, включая выбор провайдеров, обеспечивающих лучшее соотношение цена/качество; стандартизации в сфере проектирования и управления ИТ-инфраструктурой, предоставляющей услуги; повсеместной автоматизации процессов с реализацией в соответствующем инструментарии для повышения скорости, надежности и предсказуемости предоставления услуг.

Применение стандартизации и масштабной автоматизации ИТ-операций позволяет одному администратору поставщика облачных услуг и крупных ОТС, такому как Amazon.com, eBay, Google, Yahoo, обслуживать 2000 серверов в то время, как нормой для обычных ИТ-организаций является 10–20 серверов на одного администратора [238]. Достижению высокого уровня производительности ИТ-организациям препятствует не только многообразие и объем предоставляемых ИТ-услуг, но и слабая автоматизация управления ИТ-инфраструктурой.

За многие годы ИТ-организации накопили большие номенклатуры сложных приложений и услуг, которые должны быть сохранены и поддерживаться. Кроме того, номенклатура приложений и услуг продолжает увеличиваться, и все это происходит на фоне необходимости уменьшения стоимости предоставления бизнес-услуг и повышения производительности ИТ-операций. Преодолеть это препятствие можно с помощью автоматизации и технологии облачных вычислений [238]:

- автоматизация может снизить остроту проблемы, вызванной большой номенклатурой ИКТ и услуг. Бизнес-услуги часто собираются из существующих компонентов с помощью ПО промежуточного слоя. В результате получаются сложные распределенные приложения, которыми трудно управлять. Но в то же время большая номенклатура позволяет выделить множество схожих рутинных операций и задач, которые могут быть автоматизированы с использованием соответствующего инструментария. Разнообразие инструментария порождает новую проблему – обучение персонала для работы с инструментарием и обеспечение совместной работы инструментов и персонала. Эта проблема также может быть решена автоматизацией, позволяющей сгладить различия и интегрировать инструментарий, упростив администрирование и снизив требования к численности и квалификации персонала;
- облачные вычисления предоставляют большой потенциал для экономии [328]. Опыт применения облачных технологий свидетельствует, что относительно дешевые, стандартные аппаратные средства в сочетании с виртуализацией ресурсов позволяют обеспечить большую гибкость в управлении доступом, в распределении вычислительных и телекоммуникационных ресурсов наряду с прогрессом в области СУ консолидированными ресурсами. При этом с точки зрения повышения производительности не имеет значения какие используются облака – внешние или внутренние.

1.2.3. Влияние технологии облачных вычислений на развитие ИТ

Постоянно возрастающая сложность ИТ влечет за собой увеличение сложности ИТ-управления, требует повышения квалификации персонала и, несмотря на использование виртуализации и автоматизации некоторых процессов ИТ-управления, приводит к увеличению численности персонала ИТ-подразделения. Выходом из такой ситуации является переход к облачным технологиям, которые не только обеспечивают сокращение капитальных затрат, но и требуют изменений в ИТ-технологиях и обусловливают необходимость изменений в организационных моделях. Облачные вычисления гарантируют экономию за счет масштаба и автоматизации, обеспечивают рост производительности и могут покрыть будущие потребности не только большого и среднего, но и массового малого бизнеса. В ближайшие пять лет управление ИТ на основе облачных технологий будет развиваться от стадии накопления опыта работы к использованию зрелых решений [238].

Переход на облачные технологии требует изменений в операционной деятельности ИТ-подразделения. Кроме того, необходимо учитывать множество технологических особенностей, которые возникают при использовании облачных вычислений. В случае внутреннего облака это будет отличающаяся от традиционной работа с приложениями, конфигурациями, мониторингом и управлением производительностью. Внешние облака требуют дополнительного внимания к вопросам стоимости, безопасности и учета сетевых задержек, которые могут быть существенными.

Для полной реализации преимуществ облачной модели управления ИТ Forrester рекомендует развивать дополнительные дисциплины [238]:

- *обнаружение и управление активами*. При принятии облачной модели необходимо знать, где и что находится. Поэтому обнаружение и отслеживание активов и приложений в режиме реального времени становится более важным, чем при традиционном подходе к организации ИТ. Конфигурации могут быть легко изменены и приложения могут легко перемещаться, сохранение контроля над ЦОД требует абсолютной четкости. ИТ-подразделения должны внедрять новейшие решения обнаружения зависимостей, имеющихся в различном инструментарии, таком как управление производительностью приложений или мониторинг бизнес-транзакций;
- *мобильность приложений.* Должна быть возможность легко перемещать приложения, а для обеспечения автоматической конфигурации сами приложения должны обладать способностью к легкой загрузке и настройке. Для этого необходим автоматизированный процесс, который соберет приложения, зависимости между ними и элементы конфигурации в мобильный блок. Это повлияет на жизненный цикл приложений и процесс обновления версий. ИТ-организации должны полностью контролировать развитие приложений и участвовать в автоматизации процесса управления релизами;
- *новую архитектуру приложений.* Традиционная архитектура приложений непригодна для применения в облаках. Современные приложения используют ПО промежуточного слоя для связи иногда очень разных, но функционально зависимых частей кода. Эта зависимость является препятствием к использованию гибкости, обеспечиваемой облачными вычислениями. Необходимы новые архитектуры приложений, ориентированные на применение во внутренних или внешних облаках [328, 355];

• *систему финансового управления*. Проведение тщательного стоимостного анализа должно помочь ИТ-руководителям выбрать, будет ли новая услуга разработана или приобретена, развернута во внутреннем или внешнем облаке;

• *инструментарий планирования мощности*. Облачные вычисления расширяют возможности, но ресурсы облаков не бесконечны и не бесплатны. Прогнозирование мощности внутренней или внешней по-прежнему требует понимания размеров капитальных и эксплуатационных затрат. Большой выбор и простота закупки, обеспечиваемые облачными технологиями, делают планирование мощности более сложным;

• *инструментарий управления рабочими характеристиками и мощностью, автоматизации мониторинга и распределения ресурсов*. Распределение ресурсов основано на использовании свойства гибкости, характерного для виртуализации. Решение об увеличении или уменьшении ресурсов, выделяемых сервису, является функцией службы управления рабочими характеристиками сервиса. Гармоничное распределение ресурсов происходит после процедуры инициализации. Сегодня это ручная задача, требующая от администратора анализа данных о производительности. В будущем управление распределением ресурсов будет проводиться на основе обработки сложных взаимосвязанных событий. Управление производительностью приложений – типичное решение, которое сочетает в себе мониторинг производительности с аналитикой, необходимой для обеспечения гармоничности распределения ресурсов [238];

• *решения для управления ИТ со свойствами абстрагирования от разнообразия и маскирования сложности везде, где это возможно*. Отличия ИТ должны быть нивелированы с точки зрения управления ИТ, а СУИ должны предоставлять специфический, простой и единый интерфейс взаимодействия со всеми процессами управления ИТ, независимо от конкретной ИТ-инфраструктуры. Это необходимо для маскирования разнообразия ИТ-инфраструктур. Независимо от того, развернут сервис на физических или виртуальных серверах, с использованием проприетарного или открытого ПО, либо предоставляется посредством SPI методами облачных вычислений, все детали должны быть известны не только для операционной деятельности, но и с точки зрения развития и управления ИТ. Поскольку многие решения ИТ-управления по-прежнему специфичны для каждой из ИКТ, ИТ-руководители должны убедиться, что элементы ИТ-управления, предоставляемые разными поставщиками, могут быть легко интегрированы и способны работать с общим интерфейсом, позволяющим абстрагироваться от каждой из ИКТ, используемых в корпоративной ИТ-инфраструктуре.

В настоящее время ИТ-инфраструктуры, как правило, не предназначены для использования облачных сервисов, потому что организованы из набора технологий: мэйнфреймов, серверов, систем хранения, сетей, ПК, телефонии и пр. (см. рис. 1.1 и 1.2). Для использования преимуществ облачных технологий Forrester дает такие рекомендации ИТ-директорам [238]:

• роли персонала должны стать более специализированными. В настоящее время сложность ИТ-инфраструктур вынуждает одного администратора поддерживать несколько систем и услуг. Так, администратор сети решает широкий спектр несвязанных задач от разработки и администрирования до устранения неисправностей. Поддержание работоспособности сложных ИТ-инфраструктур не оставляет времени для работы над развитием ИУС;

• перенимать лучшие практики ITIL. Возможно, эту рекомендацию Forrester можно будет принять после наработки большого объема опыта работы с облачными вычислениями;

• разработать новые метрики, способствующие постоянному повышению эффективности и производительности;

• следует добавлять новые должности и рабочие группы в ИТ-организации. Появление новых функций при работе с облачными вычислениями требует, например, введения должности архитектора решений, который определяет структуру предоставления сервисов приложениями, функционирующими во внутренних или внешних облаках, осуществляет рациональное размещение элементов ИТ-инфраструктуры, производит сборку сервисов путем интеграции внутренних, внешних и облачных услуг, предоставляемых несколькими провайдерами;

• изменить структуру всей ИТ-организации. При переходе к облачным вычислениям повысится специализация ИТ-отделов в области управления проектами, запросами, данными, безопасностью и услугами;

• перейти от операционной деятельности к контролю и директивным указаниям. Чем больше услуг будет предоставляться облачными технологиями, тем меньше традиционных функций выполняют сотрудники ИТ-подразделения. Так, потребность в разработке, развертывании и поддержке ИТ будет снижаться, а функции проектирования, компоновки, контроля и оценки качества сервисов, предоставляемых облаками, усиливаться вместе с возрастающей ответственностью за правильную интеграцию разрозненных частей, осуществленную за счет объединения всех внутренних и внешних услуг;

• акцент на консалтинг. ИТ-специалисты будут оказывать больше консультационных услуг пользователям по вопросам организации и работы бизнес-сервисов, согласования потребностей бизнеса с возможными ограничениями услуг, эффективного и безопасного пользования услугами, баланса затрат и качества услуг, предоставляемых с помощью облачных вычислений.

Одним из основных факторов, сдерживающих стремительное развитие облачных технологий, является недоверие к облакам с точки зрения стабильности предоставления услуг, а также опасения за безопасность корпоративных данных. Кроме того, многие компании четко не понимают, зачем им нужен переход к облачным технологиям.

1.3. АНАЛИЗ ОСОБЕННОСТЕЙ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ

1.3.1. Основные факторы современного этапа развития систем управления ИТ-инфраструктурой

Исследования, проведенные международными аналитическими агентствами в области управления ИТ в 2012 г. [258], показали, что несмотря на признание важности ИТ для бизнеса осознание ИТ-руководителями места и роли ИТ-подразделения в организационной структуре предприятия, понимание целей и задач ИТ, которые сводятся к обеспечению максимальной эффективности бизнес-технологий с наименьшими затратами на ИТ, повсеместно фиксируются следующие особенности [258]:

• несогласованность приоритетов бизнеса с приоритетами ИТ-подразделения. Очень часто бизнес-менеджеры и ИТ-руководство считают критичным совершенно разные вещи. В то время как бизнес интересует совокупность скоординированных сервисов, ИТ-подразделение предлагает множество разрозненных ресурсов, технологий и приложений, решающих отдельные задачи;

• бизнес хочет осознавать ценность инвестиций и отдачу от внедрения ИТ. Для удовлетворения запросов и потребностей бизнеса ИТ-организации должны использовать модели операционной деятельности, предполагающей представление услуг, а не просто поддержание функционирования приложений. Управление ИТ-инфраструктурой должно претерпеть изменения, делая акценты на службу каталогов, порталы самообслуживания клиентов, автоматизацию и координацию операционной деятельности. Эффективная организация и автоматизация рутинных процессов управления ИТ-инфраструктурой позволят высвободить персонал для решения стратегических и инновационных задач создания перспективной архитектуры ИТ;

• отсутствие интеграции процессов управления ИТ-инфраструктурой. Даже такие родственные процессы, как управление устранением неисправностей, проблемами, инцидентами, изменениями, все еще слабо связаны. Позитивным является переход от накопления технологий к накоплению процессов и сервисов. Следующим шагом должны стать координация и интегрирование процессов и услуг [258];

• особая актуальность процессов, улучшающих удовлетворенность заказчиков ИТ-услуг, что очень важно для облачных технологий, когда акценты работы ИТ-подразделения смещаются в сторону процессов, связанных с обеспечением качества обслуживания заказчиков, таких как, например, управление портфелем услуг и управление взаимоотношениями с заказчиками;

• заинтересованность бизнеса и ИТ-руководителей в повышении отдачи от ИТ. По опросу почти трех тысяч ИТ-руководителей крупнейших компаний основным приоритетом 2012 года является существенное повышение эффективности ИТ, которое наряду с оптимизацией затрат позволит перевести операционную деятельность ИТ-подразделения на модель сервис-провайдера услуг, так как это необходимо бизнесу [258].

Исследования в области управления ИТ-инфраструктурой сосредоточены главным образом на том, как ИТ-подразделение должно быть организовано и управляемо. Сюда включаются вопросы создания корпоративной ИТ-инфраструктуры, спецификация метрик для ИТ-функций, управление проектами, разработка и поддержание стратегического ИТ-плана, а также создание организационной структуры ИТ-подразделения.

Концептуальную роль в сфере управления ИТ безусловно играет библиотека ITIL, обобщающая передовой мировой опыт [208, 230, 339, 380]. В соответствии с положениями ITIL ИТ-подразделение превращается в сервисную организацию, являющуюся партнером бизнеса с понятными для бизнес-подразделений функциями, а СУИ должны поддерживать процессное управление ИТ, автоматизировать выполнение операций, поддерживать высокое качество ИТ-услуг и, самое главное, предоставлять зависимость бизнес-сервисов от ИТ-инфраструктуры. Отличительной чертой библиотеки ITIL является ориентация на ИТ-операции. При правильном внедрении ITIL позволяет улучшить качество предоставления ИТ-сервисов, снизить продолжительность простоев ИТ-ресурсов, ускорить разрешение проблем и обеспечить высокий уровень безопасности. В настоящее

время версия ITILv3 является фактическим стандартом ITSM (IT service management), принятым почти 60% организаций, которые предоставляют ИТ-услуги [258]. По опросу Forrester Research, проведенному в 2012 г., 85% ИТ-директоров отмечают, что применение ITIL позволяет значительно повысить продуктивность сервисов, улучшить качество ИТ-сервисов (83%), повысить престиж бизнеса (65%) и существенно сократить затраты на операционную деятельность (41%) [258]. По мнению ИТ-директоров, выгоды, получаемые от внедрения методологии ITIL, полностью окупают затраты времени и человеческих ресурсов на реализацию процессного подхода. Например, только за счет построения службы поддержки пользователей компании Finisar в соответствии с ITIL степень удовлетворенности клиентов возросла с 33% до 95%, а затраты на поддержание работоспособности ИТ-инфраструктуры сократились с 4% до 2,4% от доходов компании [380].

Поиск путей повышения производительности и предсказуемости работы ИТ привел к появлению двух центральных направлений очередной волны в сфере ПО управления ИТ: управления бизнес-сервисами (Business Service Management, BSM) и базы данных управления конфигурациями (Configuration Management Database, CMDB), позволяющей создавать упорядоченное описание объектов технологической инфраструктуры для бесшовной поддержки процессов ITSM. База CMDB необходима при осуществлении контроля за ресурсами, поскольку содержит историю всех выполненных операций управления ИТ-инфраструктурой. CMDB является основой ITIL и представляет собой отражение всех ИКТ, автоматизированных систем, маршрутизаторов, серверов, персональных компьютеров и т. д., содержит записи всех изменений, связанных с ресурсами ИТ-инфраструктуры, историю инцидентов, выявленных при работе этих ресурсов, с описанием места ресурсов в ИТ-инфраструктуре. При проведении модернизации ИТ-инфраструктуры или внесении существенных изменений в конфигурационные файлы, которые могут привести к временной неработоспособности бизнес-сервисов, CMDB может помочь понять, на какие бизнес-процессы это повлияет. Появляется возможность оптимизировать время проведения модернизации с учетом графика работ и загруженности бизнеса для сокращения возможных негативных последствий от приостановки предоставления сервиса. В версии ITILv3 роль BSM и CMDB существенно возросла. Однако разрекламированные широкие возможности CMDB-продуктов, которые должны были привести к появлению монолитных и всеобъемлющих хранилищ данных о конфигурациях элементов ИТ-инфраструктуры с последующей чуть ли не автоматической реализацией процессов управления, не оправдались [3].

Несмотря на популярность ITIL внедрение рекомендаций сопряжено с рядом трудностей. Так, эксперты отмечают существенное увеличение сложности ITILv3 по сравнению с ITILv2, обусловленной тем, что библиотека стремится быть максимально универсальной и дать ответы на все вопросы. Для внедрения методологии ITIL ИТ-консультанты рекомендуют обращаться к интеграторам, которые решат проблемы на основе конкретных продуктов, при этом выбор и настройка конкретных инструментов управления ИТ-услугами часто оказываются гораздо дороже лицензий на интегрируемые продукты. Это объясняется тем, что книги последней версии ITIL создавались консультантами или производителями, зарабатывающими на жизнь внедрением описанных методик [380].

Бизнес развивается очень динамично, а реализация ITIL сопряжена с длительным поэтапным внедрением всех процессов и требует много времени на достижение ими стадии постоянного совершенствования. На внедрение новых процессов у большинства ИТ-организаций уходят годы, и еще годы требуются для того, чтобы оценить достигнутые результаты [380]. При этом СУИ, построенная в соответствии с ITIL, не предполагает внесения изменений.

Несмотря на то, что ITIL является обобщением лучшего опыта управления ИТ, библиотека не содержит рекомендаций или советов по практическому применению обобщенного в ней опыта, а также методик оптимальной декомпозиции процессов [266]. Фактически ITIL не внедряется в организации, а используется как методология проведения преобразований в ИТ-подразделениях, на основе которой разрабатываются собственные процедуры, исходя из принципов ITIL [380].

Основной недостаток ITIL пользователи видят в том, что применение рекомендаций ITIL оправданно для крупных корпораций, финансовых или банковских структур, телекоммуникационных компаний с большими ИТ-подразделениями, но затруднено для среднего бизнеса и неприменимо для малого бизнеса.

Однако аналитики видят в ITIL перспективную основу для организации управления ИТ, а большинство производителей и разработчиков СУИ приняли процессную модель, основанную на ITIL.

Экономический кризис заставил бизнес ограничивать бюджет на ИТ и жестко контролировать ИТ-расходы, при этом пересмотреть свое отношение к ИТ и впредь рассматривать управление ИТ по принципу бизнес в бизнесе. Теперь ИТ-подразделение должно руководствоваться экономическими факторами, а не технической целесообразностью. Это принципиально изменяет общую картину рынка ПО управления ИТ-инфраструктурой.

1.3.2. Особенности управления ИТ-инфраструктурой

Первоначально ИТ УИТИ создавались для управления сетевым оборудованием. Большая часть фирменных СУ предназначается для работы с оборудованием собственного производства, а модернизация и применение этих систем для управления оборудованием других производителей сильно затруднены или невозможны. Такая же картина имеет место и в области управления ИТ-инфраструктурой, где используется большое количество ИКТ и оборудования разных производителей. Каждая из используемых в ИТ-инфраструктуре ИКТ может иметь собственную СУ, работающую независимо от других СУ. При этом разрозненные СУ решают только собственные частные задачи и, как правило, совершенно не учитывают проблемы, возникающие в других СУ. В этом случае СУ, непосредственно управляющие отдельными технологиями, оборудованием или средствами и опосредованно – отдельными бизнес-процессами, могут не учитывать значимость других бизнес-процессов и приоритеты приложений. Более того, разрозненные СУ не могут учитывать изменение приоритетов приложений при изменении условий функционирования ИТ-инфраструктуры.

Создание СУИ сопряжено с необходимостью решения множества разнообразных взаимосвязанных задач эффективного управления ИТ, ИТ-инфраструктурой и ИУС в условиях динамики доступности и производительности ресурсов, изменяющихся свойств телекоммуникационной сети и пр.

В ИТ-инфраструктурах крупных корпораций, ОТС, государственных и правительственных структур вместе с основной ИТ-инфраструктурой, предназначенной для поддержки бизнес-процессов корпораций, создаются и развиваются масштабные решения вспомогательных ИТ-систем, повышающих эффективность работы функциональных, обеспечивающих подсистем и ИУС, а также самих ИТ-подразделений и осуществляющих, наряду с множеством функций управления и такие функции, как надежное хранение данных, защиту информации и пр. При традиционной организации ИТ-подразделения создается несколько ИТ-служб, обеспечивающих информационную поддержку одного или нескольких бизнес-процессов, отвечающих за работоспособность отдельных распределенных приложений, ИТ-сервисов, сервисов и ресурсов ИУС и пр. В результате растут финансовые затраты, увеличиваются количество ИТ-персонала и вероятность сбоев по мере увеличения масштаба ИТ-инфраструктуры, что отрицательно влияет на эффективность финансово-хозяйственной деятельности предприятий. Кроме того, при таком подходе большое количество различных информационных технологий и ресурсов в ИТ-инфраструктуре, с точки зрения управления технологиями, службами и ресурсами, приводит к тому, что на предприятиях возникает следующая ситуация:

- при управлении ИТ-инфраструктурой отсутствует единое видение у менеджеров ИУС, администраторов ИТ-служб и руководителей предприятия по причине местнического подхода и различия в понимании целей функционирования отдельных ИКТ и ИТ-инфраструктуры в целом;
- отсутствует единая СУ функционированием ИТ-инфраструктуры и средств интеграции управления различными ИТ и ИКТ, что требует использования большого количества СУ, а также знания разнообразных ИТ, принципов управления разными ресурсами, различными телекоммуникационными системами и сетями и пр.; при этом существенно увеличивается количество администраторов и менеджеров;
- повсеместное отсутствие единой информационной базы данных, необходимой для эффективного управления, вынуждает администраторов и менеджеров пользоваться различными данными для принятия решений;
- отсутствует единый центр управления, что особенно характерно для больших корпораций; это приводит к несогласованным действиям администраторов и менеджеров, нарушающим функционирование ИТ-инфраструктуры, неэффективной работе бизнес-приложений и нерациональному использованию ресурсов и процессов.

Учитывая, что ИТ-инфраструктуры отличаются размерами, характером решаемых задач, определяемых бизнесом, создание единой ИТ УИТИ с обширным функционалом, полностью решающей проблемы управления ИТ-инфраструктурой, представляет чрезвычайно сложную, трудоемкую, дорогостоящую, но решаемую задачу.

1.3.3. Анализ стандартов, практик, протоколов и существующих систем управления ИТ-инфраструктурой и уровнем услуг

Аспекты построения СУ верхними уровнями предоставления услуг, а также управления ИТ-подразделением хорошо проработаны в международных стандартах. Решению вопросов корпоративного ИТ-управления, актуальных для

крупных международных компаний и важных для среднего и малого бизнеса, посвящен ряд международных стандартов. В документах ITILv3, ISO/IEC 20000 [262, 263], COBIT [215], Six Sigma [170], рекомендациях M.3050 ITU-T и TM Forum по eTOM и др. рассматриваются процессы управления предоставлением ИТ-услуг и вопросы менеджмента ИТ-подразделения (табл. 1.1).

Таблица 1.1. Основные стандарты, практики и протоколы, применяемые в управлении ИТ-сервисами и ИТ-инфраструктурой

Организация	Стандарты, практики, протоколы	Сфера действия
OGC	ITIL	Способы организации работы ИТ-подразделений или компаний, предоставляющих услуги в ИТ-области
ISACA	COBIT	Принципы управления и аудита ИТ
Motorola	Six Sigma	Методика настройки бизнес-процессов с целью минимизации вероятности возникновения дефектов в операционной деятельности
IBM	PRM-IT	Процессная модель управления ИТ
TM Forum	eTOM	Архитектура бизнес-процессов телекоммуникационных компаний и рекомендации по созданию СУ для ОТС
RosettaNet	PIPs	Индустриальные стандарты для B2B коммуникаций
ISO	ISO 20000	Система, процессы управления качеством ИТ-сервисов
	CMIP	Интеллектуальные агенты, многофункциональная сложная система управления, объектно-ориентированный подход для представления управляемых устройств
	SPICE	Подход к оценке зрелости процессов
IETF	SNMP	Агенты выполняют элементарные функции, управление простое, ориентировано на переменные
DMTF	WBEM	Веб-ориентированное управление сетями и системами
	CIM	Объектно-ориентированная схема объектов управления
ITU-T	TMN	Архитектура системы управления ТКС
SEI	CMMI	Набор рекомендаций в виде практик, реализация которых необходима для совершенствования процессов в организациях разных размеров и видов деятельности. Доработанный вариант модели СММ
OMG	CORBA	Поддержка разработки, развертывания и функционирования объектно-ориентированных распределенных программных систем

АУТСТИ

Эффективное руководство бизнесом требует улучшения корпоративного управления, которое в последние годы получает законодательное регулирование. Так, в США закон Сарбейнза-Оксли ужесточает режим контроля и регулирования финансовой деятельности и, в частности, размещения активов. Для управления жизненным циклом ИТ-активов могут использоваться процессы управления конфигурациями и изменениями, являющиеся в соответствии с ITIL основой управления ИТ-услугами.

Стандарт ISO 20000 делает конкретной центральную контролирующую функцию, а соответствие действий ИТ-подразделения положениям этого стандарта позволит оценить также и степень зрелости корпоративного управления. В то же время в ISO 20000 не рассматриваются аспекты управления ИТ-инфраструктурой, ИУС и распределенными приложениями.

Группа документов GB921 Международной ассоциации TM Forum определяет расширенную карту процессов телекоммуникационных компаний (eTOM), которая используется ОТС в качестве модели платформы предоставления услуг, основанной на концепции бизнес-процессов и стандартах управления сетями. eTOM, концентрируясь на бизнес-процессах ОТС, не уделяет должного внимания сфере управления коммуникационной инфраструктурой.

В разработанном TM Forum и itSMF (IT Service Management Forum – форум по ИТ сервис-менеджменту – некоммерческое партнерство, активно продвигающее концепцию ITIL) документе [204] показано, как eTOM и ITIL могут быть использованы совместно. Несмотря на стратегическую схожесть ITIL и eTOM, терминология, область применения и определения eTOM и ITIL существенно различаются. Прямого отображения ITIL на eTOM не существует, и ITIL подходит только для высокоуровневых процессов eTOM. В то же время eTOM, концентрируясь на бизнес-процессах ОТС, не уделяет должного внимания сфере управления коммуникационной инфраструктурой.

Обобщение мирового опыта, а также анализ международных стандартов и рекомендаций, имеющих отношение к ИТ-управлению, составляет пакет документов COBIT [208]. Главная задача COBIT – преодоление разрыва в видении целей бизнеса руководством компании и понимании целей бизнеса ИТ-подразделением, осуществляющим поддержку ИТ-услуг, которые способствуют достижению этих целей. Являясь средством аудита ИУС, описанием целей и принципов управления, объектов управления, управления ИТ-безопасностью, стандартизацией ИТ-процессов, протекающих в компании на верхних уровнях ИТ-бизнеса, COBIT не уделяет должного внимания вопросам управления ИТ-инфраструктурой.

В стандартах ITU-T на TMN [320] предложена концепция и описана архитектура СУ сетями разных уровней и масштабов, предоставляющих различные типы услуг. Концепция TMN призвана объединить функции существующих СУ с добавлением высокоуровневого сервиса для организации единой сетевой структуры, обеспечивающей взаимодействие различных типов управляющих средств и телекоммуникационного оборудования, использующих стандартные протоколы. Универсальность и гибкость, характерная для TMN, стала причиной сложности архитектуры и интерфейсов, и, как следствие, незначительной практической реализации принципов TMN. Концепция TMN, позволяющая объединять сети и сетевые технологии, не затрагивает вопросы управления ИТ-инфраструктурой.

Компания IBM предложила свою процессную модель IBM PRM-IT(IBM Process Reference Model for IT) [266], которая концептуально мало чем отличается от ITIL. Модель PRM-IT рассматривает ИТ как важный компонент бизнеса, управление которым аналогично управлению производственными ресурсами.

Основные принципы процессной модели IBM PRM-IT следующие:

- независимо от организации и технологий существует некоторый фундаментальный набор процессов, необходимых для управления любой ИТ-средой.
- эти процессы не существуют и не выполняются независимо от других процессов, фактически процессы взаимосвязаны и взаимодействуют друг с другом.
- не существует единственной, доказуемо правильной декомпозиции процесса, или каких-либо способов демонстрации того, что конкретная реализация ИТ-процесса лучше любой альтернативной реализации. Все зависит от конкретных условий внедрения процесса.
- устоявшееся определение термина «лучшая практика» из ITIL представляет собой стандарт де-факто для множества ИТ-процессов, известных как «Service Management».

Компания IBM уверяет, что следование этой модели позволит поэтапно проектировать и внедрять ИТ УИТИ с использованием инструментария различных вендоров, с непротиворечивым дополнением модели успешными практиками из CMMI, COBIT, ITIL, Lean, Six Sigma и т. д. Модель позволяет определять роли и распределять ответственность с гарантией их полноты и с непротиворечивостью для возможного отражения на любую структуру ИТ-организации.

Модель компании IBM по сути является попыткой объединения ITIL, COBIT и других практик.

С аналогичной инициативой совместного использования ITIL и COBIT выступила компания BMC [256], пытаясь на базе собственных продуктов реализовать идею слияния стандартов. В настоящее время стандарты развиваются самостоятельно и не противоречат друг другу.

По мнению аналитиков, ИТ-организации для достижения успеха должны совершенствоваться, двигаясь в сторону процессов управления ИТ-услугами – ITSM [258, 264], принимая лучшие практики, описанные в ITIL. Forrester определяет ITSM как «основанные на процессах практики, целью которых является стабильное предоставление ИТ-услуг с учетом потребностей предприятия с акцентом на преимущества для пользователей. ITSM требует смены парадигмы от предоставления и управления ИТ как стеком отдельных технологий на сосредоточении поставки и управления ИТ-услугами с использованием лучших моделей процессов практик, таких как ITIL, признанных во всем мире лучшей практикой ITSM» [258].

В ходе опроса [306] почти пятисот ведущих специалистов, участников американской секции форума IT Service Management Forum (itSMF), работающих в области ИТ-инфраструктур и операционной деятельности по вопросу перспектив развития ITSM после 2011 г., агентство Forrester обнаружило, что почти все крупные предприятия США имеют некоторую форму подхода к ITSM, основанную на ITIL. Причем основанные на ITIL программы и сертификаты направлены не только на повышение производительности и качества ИТ-услуг, но и на улучшение репутации и престижа бизнес-лидеров, контроля расходов на ИТ и пр.

Аналитики Forrester выбрали 18 наиболее важных перспективных процессов [239, 258, 280], представляющих собой прошлое, настоящее и будущее ITSM (см.

табл. 1.2). Эти процессы основаны на ITILv3 и способствуют переходу ИТ-организации к 2030 г. от поставщика ИТ-услуг к роли посредника. Часть этих процессов используется уже много лет, часть являются относительно новыми, а некоторые ранее использовались, но сейчас происходит их возрождение в кардинально новых условиях предоставления ИТ-услуг с повсеместным применением облачных технологий.

Рынок ПО управления ИТ и ИТ-инфраструктурами – один из наиболее динамичных и многогранных рынков ИТ-индустрии, значимость которого существенно возрастает с годами. Аналитики Forrester [279] считают, что под категорию систем управления ИТ подпадают все программные продукты, осуществляющие мониторинг, обнаружение и идентификацию любого аномального поведения ИТ-инфраструктуры. Сюда относят ПО, осуществляющее управление серверами, приложениями, сетями, событиями, производительностью БД, работой пользователей, инструментарием автоматизации ЦОД и другими компонентами. Под эту категорию подпадает и ПО, осуществляющее управление самой ИТ-инфраструктурой (управление ресурсами, изменениями и конфигурацией), поддержку ее функционирования и эксплуатации (планирование работ, управление потоками работ), управление коммуникациями как внутри ИТ-подразделения, так и с бизнес-подразделениями (служба поддержки, управление качеством обслуживания, управление бизнес-сервисами). Таким образом, управление ИТ-инфраструктурой рассматривается как составляющая управления ИТ в целом.

В табл. 1.2 описаны 18 наиболее важных процессов управления ИТ, которые, по мнению аналитиков Forrester, будут актуальными до 2020 г.

Таблица 1.2. Основные перспективные процессы управления ИТ

Процесс	Описание процесса
Управление доступностью и производительностью	Отвечает за измерение и улучшение доступности и производительности ИТ-услуг и связанных с ними элементов конфигурации или ИТ-активов (серверы, сети, БД, приложения). Используется для поиска причин проблем с производительностью приложений или бизнес-услуг. Понимание зависимости работы пользователя от конкретных приложений, БД, ПО промежуточного слоя и элементов ИТ-инфраструктуры в комплексе позволяет ИТ-подразделению управлять и автоматизировать доступность и производительность, как это необходимо бизнесу
Управление мощностями	Оптимизация мощности ИТ-услуг, бизнес-услуг и ИТ-инфраструктур таким образом, чтобы доступные ресурсы были способны предоставить услуги с согласованным уровнем обслуживания экономически-эффективно и своевременно. Процесс включает планирование, оценку и постоянное управление мощностями. Процесс призван избежать ненужных инвестиций в ИКТ, не отвечающих потребностям бизнеса, и ситуаций, когда производительность труда находится под угрозой из-за нехватки или неэффективного использования существующих ИКТ, поддерживающих ИТ-услуги и/или бизнес-услуги

Процесс	Описание процесса
Управление изменениями	Отвечает за управление и мониторинг жизненного цикла и рисков всех изменений (добавление, модификация или удаление ИТ-услуг, конфигурационных образов, процессов, документации и т.д.), гарантируя, что используются нормативные методы и процедуры отслеживания изменений. Процесс необходим для обеспечения качества и непрерывности ИТ-услуг, начиная с запроса на изменение, требующего регистрации изменения, приема и классификации, утверждения и планирования, фактического осуществления изменений, оценки изменений и закрытия запроса на изменение
Управление конфигурациями	Управление конфигурационными элементами, предоставляющими ИТ-услуги или часть ИТ-услуг, на протяжении жизненного цикла элементов. Процесс используется для поддержания бизнес-услуг и их адаптации в соответствии с потребностями бизнеса и необходим для поддержания работы ИТ-сервисов и снижения затрат на ИТ-услуги и ИТ-инфраструктуру
Управление постоянством уровня услуг	Отвечает за ведение переговоров по уровню обслуживания и соблюдение требований SLA, отслеживает процессы и услуги для их подстройки к изменяющимся потребностям бизнеса. Отслеживает согласованность целей ключевых ИТ-сервисов между собой и их соответствие целям бизнеса и пользователей
Управление взаимоотношениями с заказчиками	Процесс и мероприятия, отвечающие за налаживание отношений с внутренними и внешними заказчиками ИТ-услуг. Способствует оптимизации клиент-ориентированных процессов: управление заказом сервисов, инцидентами, уровнем обслуживания. Новые технологии, такие как социальные сети, позволяют взаимодействовать с заказчиками для повышения качества предоставления и обслуживания сервисов
Управление событиями	Управление событиями на протяжении всего их жизненного цикла – от получения и протоколирования сообщений тревоги до автоматизированного разрешения. Позволяет понять, какие проблемы возникают в ИТ-инфраструктуре для последующего разрешения автоматизированным образом или через эскалацию событий экспертам предметной области
Управление инцидентами	Управление незапланированным прерыванием или снижением качества ИТ-услуг от обнаружения инцидента до восстановления функционирования. Процесс включает все проблемы, которые нарушают или могут нарушить текущие сервисы и, как правило, информирует пользователя или инструментарий управления ИТ-инфраструктурой

АНОНС

Процесс	Описание процесса
Управление жизненным циклом ИТ-активов	Управление жизненным циклом ИТ-активов от закупки до утилизации, с акцентом на расположении, стоимости и праве собственности. Отслеживание активов вручную или с использованием электронных таблиц неэффективно и чревато ошибками. Автоматизация процесса позволяет получить знания о доступных аппаратных и программных активах для принятия решений, управления мощностью и планирования. При использовании облачных технологий необходимо отслеживать инфраструктурные и программные ИТ-активы в режиме реального времени
Управление финансами	Использование традиционного бухучета и основанного на значимости инструментария и методов для обеспечения эффективного бизнес-ориентированного управления ресурсами ИТ-инфраструктуры при предоставлении клиент-ориентированных ИТ-услуг
Управление знаниями	Отвечает за сбор, анализ, хранение и распространение знаний (данные и информация, подготовленные для целевой аудитории) в рамках ИТ-подразделения. Процесс используется для распространения знаний на всех уровнях техподдержки. Повышение эффективности за счет снижения необходимости заново открыть знания
Управление портфелем услуг	Описание услуг в терминах значимости для бизнеса, сформулированных с точки зрения заказчика. Уточнение перечня ИТ-услуг. Объединение их в пакеты и определение преимуществ, предоставляемых пакетами
Управление проблемами	Управление жизненным циклом проблем для предотвращения инцидентов и минимизации последствий инцидентов, которые не могут быть предотвращены. Реактивное управление проблемами заключается в ликвидации последствий инцидента. Проактивное управление позволяет разрешить проблемы, прежде чем они окажут негативное влияние на бизнес-процессы. Процесс используется при анализе и поиске причины инцидентов и деятельности, направленной на предотвращение появления инцидентов
Управление релизами и развертыванием (предоставлением услуг)	Управление релизами процесса планирования, составление графика и контроль за перемещением релизов (наборов аппаратных средств, ПО, документации или других компонентов, как единого целого) для проверки в реальной среде. Управление развертыванием отвечает за продвижение новых или модифицированных видов оборудования, ПО, документации или процессов в реальные условия. Управление релизами используется для управления процессами, системами и функциями, необходимыми для компоновки, сборки, тестирования и развертывания релизов в производство

Архив КГТУ

Процесс	Описание процесса
Управление каталогом сервисов	Создание и поддержание каталога сервисов с указанием статуса, интерфейсов и порядка предоставления услуг. Используются для предоставления деталей и состояния существующих сервисов и бизнес-процессов, поддерживаемых ИТ-инфраструктурой, а также сервисов, находящихся в разработке или уже не используемых
Метрики сервисов и отчетность	Идентификация, сбор и представление информации о предоставлении ИТ или бизнес-услуг, определяющей типы и количество услуг, их вклад, риски и улучшение возможностей для бизнеса. Метрики используются для измерения эффективности процессов и сервисов
Управление заказом сервисов	Управление жизненным циклом запросов пользователей для получения информации о сервисах или имеющихся аналогах, советов или доступа к ИТ-услугам. Обычно входит в процессы службы поддержки
Организация времени и управление ресурсами	Распределение времени персонала и ресурсов для задач операционной деятельности, проектов и/или функций. Процесс планирования, назначения и контроля времени и ресурсов для эффективного размещения сотрудников с необходимыми навыками на высокоприоритетное сервисное обслуживание и выполнение работ

Новыми процессами, зрелость которых отражает степень перехода к клиент-ориентированным технологиям [258], являются управления: постоянством уровня услуг; взаимоотношениями с заказчиками; портфелем услуг.

Многие ИТ-специалисты не понимают принципиальной разницы между управлением инцидентами и проблемами. Инциденты – это проявление разного рода нестандартных ситуаций, а причиной появления проблем могут быть фундаментальные недостатки ИТ-инфраструктуры, которые могут приводить к массовой регистрации инцидентов. Поэтому тщательный поиск причин возникновения инцидентов может помочь в ликвидации дефектов проектирования или интеграции элементов ИТ-инфраструктуры.

Для успешного перехода от роли поставщика ИТ-услуг к роли посредника, предлагающего общие и частные ИТ-услуги, ИТ-организации должны оптимизировать внедрение ITSM процессов (табл. 1.2) и обеспечить скорейшее достижение процессами стадии постоянного совершенствования [258]. Для этого нужно стратегически правильно расставить приоритеты развертывания и развития процессов ITSM с учетом специфики и состояния ИТ-организации, понимания требований к ИТ со стороны бизнеса, осознания необходимости приведения в соответствие ИТ-услуг с бизнес-целями и потребностями пользователей.

Компании ASG, Axios Systems, BMC Software, CA Technologies, Cherwell Software, EasyVista, FrontRange Solutions, helpLine, HP, IBM, iET Solutions, LANDesk, ManageEngine, Numara Software, ServiceNow, Symantec и VMware предоставляют продукты для поддержки практически всех процессов [244, 242, 258]. В основном рынок делят межу собой CA Technologies, IBM, HP, BMC Software, Symantec и EMC [244], из которых мегавендорами являются BMC Software, CA Technologies, IBM Tivoli, HP Software [242]. В то же время отмечается снижение доли рынка этих поставщиков за счет активности новых производителей,

предлагающих инновационные или специализированные решения по управлению ИТ, ориентированные прежде всего на малый и средний бизнес.

Существенное влияние ИТ на достижение целей бизнеса не только подчеркивает значимость ИТ-сервисов, но и акцентирует необходимость управления ими. Лидирующее положение в области унификации управления ИТ-сервисами принадлежит ITSM, признанного во всем мире и повсеместно реализуемого подхода к ИТ-управлению, о чем свидетельствует появление международного стандарта ISO/IEC 20000. Стандарт ISO/IEC 20000 – первый международный стандарт управления ИТ-услугами. Он содержит: требования к руководству, документированию, компетенции, осведомленности и подготовке персонала; требования к мониторингу, измерению, оценке и улучшению процессов; принципы формирования плана управления услугами и порядка применения цикла Деминга к управлению ИТ-услугами. Стандарт внес изменения и дополнения в процессную модель, перейдя от набора процессов к созданию целостной системы управления ИТ-услугами. Выделяется 13 процессов, разделенных на пять групп, и две области управления верхнего уровня.

Оригинальная концепция управления бизнес-сервисами (BSM) направлена на улучшение решений и продуктов управления ИТ, чтобы сделать их более интеллектуальными. BSM была призвана преодолеть пропасть между видениями пользователя, который главным образом наблюдает только приложения, и администраторов, которые рассматривают элементы ИТ-инфраструктуры как совокупность отдельных ИКТ. Для этого проводится отображение приложений на элементы инфраструктуры, участвующие в работе приложений. Приложение могло быть смоделировано после учета всех зависимостей, что делает процесс управления более интеллектуальным и эффективным. Например, ИТ-подразделение может ограничить область исследований жалоб пользователя на работу приложений по элементам модели или быстро отследить влияние неполадок в инфраструктуре на критичность приложений. Но вместо продолжения направления эволюции от управления приложениями и ИТ-инфраструктурой к бизнес-сервисам фокус BSM сместился в сторону поддержки процессов в соответствии с ITIL и стал пакетом решений, посвященным операционным процессам ИТ и управлению ИТ-производством. BSM поддерживает широкий спектр ИТ-процессов управления с техническими решениями, которые используют CMS/CMDB в качестве базиса. В соответствии ITMS2 BSM не рассматривается самостоятельной категорией ПО управления ИТ, а составляет основу множества других категорий.

В рамках BSM ITSM стремится сгруппировать несколько решений, содействующих управлению интерфейсом между пользователями и услугами, таких как каталог услуг и управление заказом сервисов [280], которые должны быть построены на основе CMS/CMDB. Поэтому ITSM становится естественным продолжением BSM. В свою очередь, в настоящее время BSM включает множество категорий ПО управления ИТ, относящихся ко всем трем группам управления [239].

Требования к управлению ИТ-услугами и к системе управления определены в ISO/IEC 20000-1, а рекомендации по организации деятельности по управлению ИТ-услугами – в ISO/IEC 20000-2. В соответствии со стандартом СУ, включающая политики и структурированный подход, должна реализовывать внедрение и эффективное управление всеми ИТ-услугами. При этом рассматривается только процессное управление, а вопросы оперативного (автоматического) управления

ИТ-инфраструктурой, состояние и функционирование которой оказывает непосредственное влияние на уровень услуг, не рассматриваются. Управление качеством услуг попадает в сферу ответственности как процессного ITSM, так и оперативного управления ИТ УИТИ (рис. 1.8).

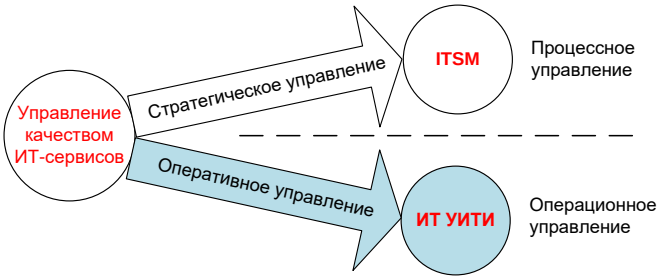


Рис. 1.8. Распределение сфер ответственности управления ИТ-сервисами между системами управления

ITSM и ИТ УИТИ имеют различные объекты управления, общим из которых являются ИТ-сервисы (рис. 1.9).

ITSM и ИТ УИТИ имеют разные цели управления. Цель ITSM – постоянное повышение качества ИТ-услуг, цель ИТ УИТИ – поддержание показателей качества ИТ-услуг на согласованном уровне. Причем ITSM не рассматривает ИТ-инфраструктуру как объект управления для поддержания заданного уровня услуг.

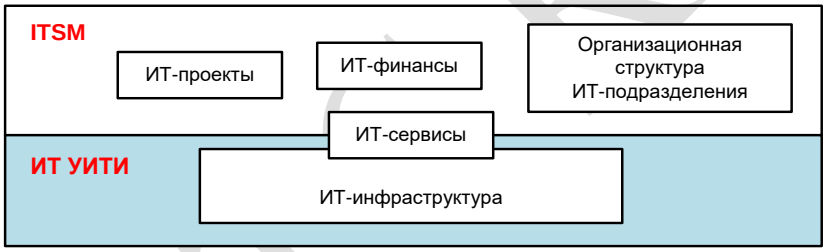


Рис. 1.9. Объекты управления ITSM и ИТ УИТИ

Из изложенного выше можно сделать вывод о том, что существующие стандарты, практики, ПО и системы управления сосредотачиваются на вопросах организационного управления ИТ-подразделением и не рассматривают детально операционное управление ИТ-инфраструктурой на всех уровнях, не предлагают модели, методы и алгоритмы рационального распределения ресурсов ИТ-инфраструктуры. Ведущие поставщики СУИ не раскрывают внутренних механизмов работы своего ПО, а ITIL и решения ITSM содержат только формализованное описание организации работы ИТ-подразделения и управление ИТ-услугами. В настоящее время отсутствуют международные стандарты, специфицирующие комплексное управление ИТ-инфраструктурой на всех иерархических уровнях. Задачи нижних уровней ИТ-инфраструктурой проработаны стандартами лишь по отдельности для сетей, серверов, хранилищ данных, отдельных ИКТ и некоторых иных компонентов, но не предлагают интегрального подхода к управлению ИТ-инфраструктурой. Не известны интегрированные СУИ, позволяющие централизованно управлять всеми элементами ИТ-инфраструктуры, а также ИУС, и представляющие инструментарий ИТ-подразделению для эффективного автоматизированного управления всеми ресурсами и приложениями.

Среди процессов ITSM и ISO/IEC 20000 отсутствуют процессы и область оперативного управления уровнем ИТ-услуг и ИТ-инфраструктурой. Это объясняется тем, что управление ИТ-услугами – сравнительно новая и активно развивающаяся область управления и если возможность и необходимость управления ИТ-услугами уже признаны (о чем свидетельствует появление стандарта), то рассмотрение ИТ-инфраструктуры в качестве объекта управления при менеджменте ИТ-услуг до сих пор в полной мере не осознано. И одной из целей авторов было доказательство того, что рассмотрение ИТ-инфраструктуры как объекта управления для поддержания приемлемого уровня качества предоставления ИТ-услуг не только возможно, но и необходимо.

1.3.4. Эволюция управления ИТ-инфраструктурой

Анализ работ, посвященных управлению ИТ-инфраструктурой, существующих СУИ, стандартов, практик, протоколов (п. 1.3.3), позволяет выделить три уровня зрелости управления ИТ-инфраструктурой (рис. 1.10): управление сетями и вычислительной техникой, управление информационными системами, управление уровнем ИТ-услуг [107].

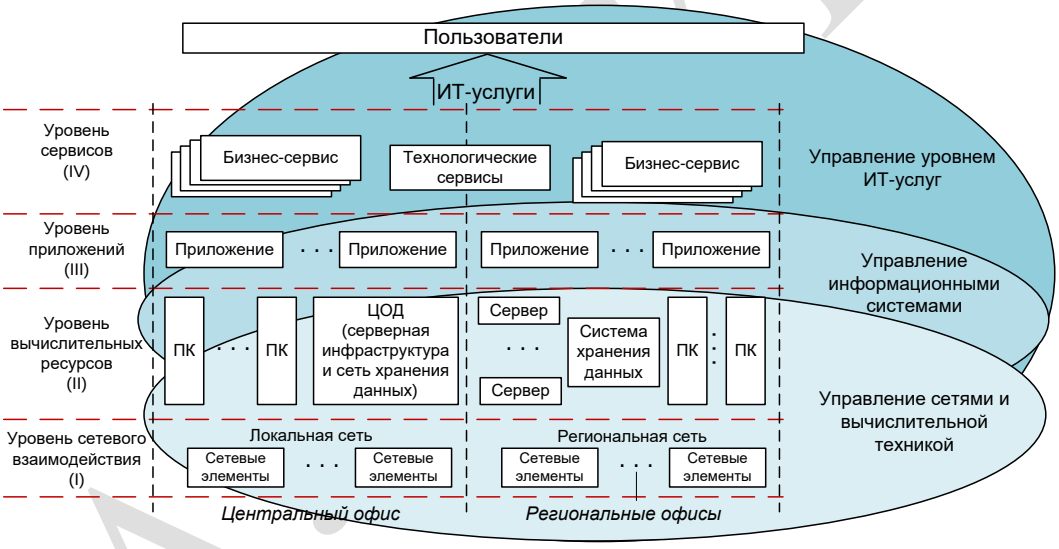


Рис. 1.10. Этапы эволюции управления ИТ-инфраструктурой

Эволюция развития средств обработки информации (см. п. 1.1.1) позволяет понять, что необходимость управления сетями и вычислительной техникой была осознана давно. Это привело к тому, что исторически первыми появились системы управления на первых двух уровнях ИТ-инфраструктуры. Вопросы сетевого управления и управления компьютерными системами очень хорошо проработаны вплоть до создания систем автоматического управления некоторыми сетями и серверами. Однако недостаточно проработаны вопросы управления ограниченными ресурсами ИТ-инфраструктуры с учетом важности бизнес-процессов, а также эффективного управления ресурсами ЦОД.

Следующим этапом эволюции управления ИТ-инфраструктурой стало управление информационными системами, которое в основном сводилось к

управлению распределенными приложениями. На этом этапе разработчики ИТ УИТИ столкнулись с отсутствием принятого ИТ-сообществом стандарта управления приложениями.

Особенность управления на этих этапах эволюции заключалась в поддержании параметров функционирования элементов и приложений на определенном, соответствующем регламенту уровне. Считалось, что если поддерживать на заданном уровне значения показателей качества функционирования ИТ-инфраструктурой, то качество предоставляемых ИТ-услуг будет соответствовать целевым значениям.

Третий этап эволюции сопровождался осознанием необходимости и возможности управления уровнем ИТ-услуг, что произошло в последние годы. Естественно, что вопросы управления уровнем ИТ-услуг в силу новизны решаемых задач исследованы и проработаны недостаточно хорошо.

Основным направлением в решении задачи управления функционированием ИТ-инфраструктуры является переход от управления сетями, компьютерными системами и приложениями как подсистемами к бизнес-ориентированному подходу, при этом ИТ-инфраструктура должна рассматриваться как основа функционирования бизнес-процессов. В связи с этим четко просматривается и очередной (четвертый) этап эволюции управления ИТ-инфраструктурой, которым должно стать управление ИТ-инфраструктурой персоналом управления бизнесом. Это станет возможным, когда степень автоматизации управления ИТ-инфраструктурой достигнет такого уровня, что управление ИТ-инфраструктурой окажется не сложнее, чем управление персональным компьютером. Это ознаменует появление новой бизнес-среды с интегрированной в нее ИТ-инфраструктурой, управление которой не будет требовать наличия специального ИТ-персонала, включенного в контур оперативного управления.

Качество предоставления ИТ-услуг обеспечивается процессами ITSM и операционным управлением ИТ-инфраструктурой. В табл. 1.3 приведен краткий анализ наличия ПО управления уровнем услуг.

Таблица 1.3. Наличие ПО управления ИТ-инфраструктурой, имеющего отношение к управлению качеством услуг

Основные функции и задачи управления качеством сервисов	Наличие ПО автоматического/автоматизированного управления	
	Процессное управление	Операционное управление
Организация работы ИТ-подразделения	+	Не рассматривается
Централизованное управление ИТ-инфраструктурой	-	-
Интегрированное управление всеми уровнями ИТ-инфраструктуры	Не рассматривается	-
Управление предоставлением услуг	+	Не рассматривается
Управление качеством ИТ-услуг	+	-

Организация сквозного процесса управления ИТ-инфраструктурой от сбора информации с объектов до управления с трансформацией в качество услуг	-	-
Анализ влияния сбоев в ИТ-инфраструктуре на качество ИТ-услуг	+/-	-
Прогноз функционирования ИТ-инфраструктуры и качества ИТ-услуг	-	-
Мониторинг качества предоставляемых сервисов	+	+/-
Поддержание качества услуг на заданном уровне	Не рассматривается	-
Управление приложениями	Не рассматривается	+/-
Взаимодействие между приложениями	Не рассматривается	+/-
Интегрированное управление приложениями, ТКС с учетом важности бизнес-процессов	Не рассматривается	-
Управление устранением неисправностей	+	-
Распространение влияния неисправностей	-	-
Управление доступом к ограниченными ресурсам	Не рассматривается	-
Распределение ресурсов ЦОД	-	-
Управление производительностью приложений	+	-
Управление потоками приложений в сети	Не рассматривается	-
Сведение метрик	-	-
Мониторинг использования приложениями ресурсов	Не рассматривается	+
Управление серверами	Не рассматривается	+
Управление сетями	Не рассматривается	+
Интегрированное управление серверами и сетями	Не рассматривается	Только мониторинг
Управление доступом пользователей к приложениям	Не рассматривается	+
Генерация отчетов о функционировании ИТ-инфраструктуры	+	+

Вопросы процессного управления уровнем услуг хорошо проработаны и стандартизованы, а вопросам управления уровнем услуг посредством оперативного управления ИТ-инфраструктурой до сих пор не уделялось достаточного внимания. Поэтому авторы, учитывая важность для бизнеса получения ИТ-услуг с высоким качеством, особое внимание уделяют именно этому аспекту управления ИТ-инфраструктурой.

АУТСКИЙ

1.3.5. Анализ подходов к построению ПО управления ИТ-инфраструктурой

Несмотря на то, что рынок ПО управления ИТ-инфраструктурой предлагает достаточно зрелые решения [242], он вопреки кризису продолжает активно развиваться. Этому способствует появление новых сфер деятельности, где необходимы соответствующие ИТ, создание новых ИТ, а также новое видение управления ИТ-инфраструктурой как производственно-экономической единицей [237]. Не удивительно, что появляются новые инструменты поддержки принятия решений на разных уровнях бизнеса. Достаточно обоснованно можно полагать, что такие сферы ИТ-управления, как управление ИТ-инфраструктурой, будут стремительно развиваться, заставляя производителей соответствующего ПО постоянно совершенствовать компоненты СУИ и делать новые предложения по расширению функционала управления [237]. К концу текущего десятилетия центр тяжести рынка ПО управления ИТ сместится в сторону глобального управления и оптимизации ИТ-ресурсов [241].

Для отслеживания всех этих изменений в области управления ИТ, ИТ-инфраструктурой и ИУС в 2009 г. аналитики Forrester предложили новую систематику категорий ПО управления ИТ, получившую название «IT Management Software 2.0» (далее – ITMS2) [239]. До этого момента широко использовалась классификация программных решений для управления ИТ, выделяющая пятнадцать основных классов, дифференцируемых по выполняемым функциям [279, 290], включая управление сетями, серверами, базами данных, событиями, хранилищами данных, работой конечных пользователей, приложениями, ресурсами ИТ-инфраструктуры, изменениями и конфигурациями ИТ-инфраструктур, производительностью ИТ-ресурсов, счетами ИТ-подразделения, персоналом ИТ-подразделения, а также службу поддержки пользователей (Service desk), планирование работ и выполнения операций, SLM и BSM.

Концепция ITMS2 предусматривает группирование основных классов управления в меньшее количество с добавлением новых классов, расширяет функции и роли ИТ-подразделения, декларирует новое видение и отношение к ИТ-подразделению.

ITMS2 основывается на следующих основных положениях:

а) ИТ-подразделение рассматривается и управляется как экономически ответственная и финансово прозрачная организация;

б) в управлении ИТ применяется ЛИН-модель;

в) организационное строение ИТ-подразделения осуществляется в соответствии с ЛИН-моделью;

г) пятнадцать классических функций управления ИТ-инфраструктурой объединяются в три группы;

д) функции управления перегруппировываются в новых категориях управления.

Экономическая ситуация заставила предприятия использовать ЛИН-технологии (Lean Thinking) [379] в управлении ИТ.

ЛИН-модель, учитывающая тенденции развития и бизнеса, и ИТ, базируется на пяти основных принципах [295] (рис. 1.11):

1. Точное определение важности ИТ-продукта или ИТ-услуги с точки зрения бизнеса. Это базовый принцип при трансформации ИТ в бизнес-технологии, который реально означает, что должен быть постоянный диалог между бизнесом и

ИТ на предмет значимости технологий или инновационного бизнеса, осуществимости и стоимости разнообразных решений. Для этого ИТ-подразделение должно обладать знаниями в технической и финансовой областях.

2. Определение потока создания ценности каждым ИТ-продуктом и услугой. Здесь учитывается знание того, как услуги построены, и стоимость услуги на каждой стадии ее жизненного цикла.

3. Обеспечение непрерывности потока значимости. Процессы следует рассматривать глобально с рациональной точки зрения для того, чтобы исключить потери, а также для того, чтобы удостовериться в том, что важность и стоимость процессов оптимизированы на всех этапах от концепции до предоставления ИТ-услуг.

4. ИТ-подразделение должно предоставлять бизнесу значимость ИТ-продуктов и услуг. Бизнес-технологии занимают на предприятии главенствующее положение, а задача ИТ заключается в автоматизации процессов с целью повышения эффективности бизнес-технологий. При этом должен соблюдаться баланс между стремлением к максимальному удовлетворению бизнес-целей и стоимостью ИТ и ресурсов, необходимых для достижения целей бизнеса. Этот баланс оценивается соотношением цена/значимость. При управлении ИТ-подразделением как предприятием заказчикам разъясняется полезность от ИТ-продуктов, которую они могут получить в рамках потенциального бюджета.

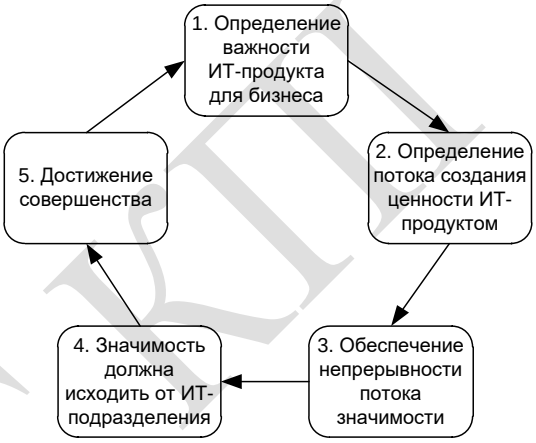


Рис. 1.11. ЛИН-модель для ИТ

5. Достижение совершенства. Качество ИТ-сервиса должно быть таким, чтобы количество неполадок при предоставлении ИТ-услуг было минимальным. Этому должно способствовать непрерывное понимание ценности каждого ИТ-процесса и требуемого качества предоставления ИТ-услуг для бизнес-технологий. В данном случае достойный ответ дают высокотехнологичная методика Six Sigma и другие статистические подходы к оценке качества выполнения бизнес-процессов, применяемые для минимизации вероятности возникновения дефектов в операционной деятельности [243].

Учитывая экспоненциальный рост сложности технологий [239], ИТ-подразделение не сможет выполнять все пять пунктов ЛИН-модели без набора базовых методов, процессов и соответствующего инструментария в виде ИТ УИТИ.

Любая инициатива, направленная на оптимизацию, требует фундаментальных знаний об элементах сервиса, основных характеристиках и структуре затрат. Структура ИТ-сервиса может оптимизироваться на глобальном, стратегическом и тактическом уровнях [239]. При этом параметры тактического уровня оптимизируются локально в ИТ-подразделении, а большая часть принципов управления и процессов имеют тактические и стратегические компоненты, что делает необходимым управление последовательностью операций этих процессов.

С точки зрения ITMS2 организационное строение ИТ-подразделения должно соответствовать подобным организационным структурам предприятия (рис. 1.12).

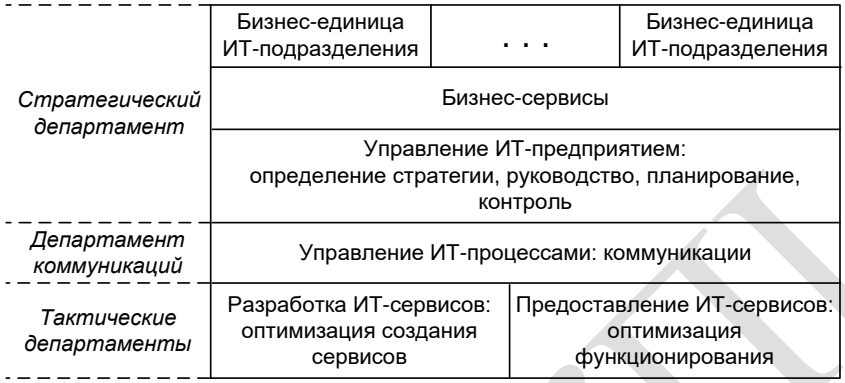


Рис. 1.12. Схема организационного строения ИТ-подразделения с точки зрения ЛИН-модели

Стратегический департамент занимает центральное место в организационном строении, решая высокоуровневые задачи и выполняя такой комплекс функций: взаимодействие с потребителями ИТ-услуг – бизнес-подразделениями (задачи маркетинга и продажи); контроль расходов и доходов (финансовые задачи); ведение переговоров с поставщиками, вынесения решений по продуктам и определение стратегии производства (задачи планирования и уровня обслуживания). Используемые методики этого департамента очевидны и подобны методикам бизнес-подразделений.

Департамент коммуникаций обеспечивает информационное взаимодействие между другими департаментами. Учитывая то, что рабочие процессы в смежных департаментах носят циклический характер, структурированный обмен информацией между ними имеет важное значение.

Как отмечалось выше, аналитики Forrester [239] предвидят повсеместный переход ИТ-подразделений от управления технологиями к предоставлению сервисов для бизнеса, что должно привести к конвергенции 15 основных категорий продуктов управления ИТ, дифференцируемых по выполняемым функциям [279, 290], в три основные группы (табл. 1.4) [290]. Поэтому ITMS2 предусматривает группирование основных категорий управления в меньшее количество и добавление новых категорий, расширяет функции ИТ-подразделения, декларирует новое видение и отношение к ИТ-подразделению. Существующая экономическая ситуация и движение к ЛИН-ИТ не изменили этой тенденции, а наоборот, добавили необходимость учета соотношения значимость/стоимость ИТ-сервисов, положительно влияющего на процесс конвергенции.

АУТСРТ

Таблица 1.4. Категории и группы ПО управления ИТ

Традиционные категории ПО управления ИТ		Группы ПО управления ИТ в соответствии с ITMS2	
Категория	Основные функции	Группа	Основные функции
1	2	3	4
Сетевое управление (Network management)	Управление сетями и межсетевым взаимодействием	Управление предоставлением и производительностью ИТ-сервисов (IT service delivery and performance management)	Управление доступностью, производительностью и событиями ИТ-инфраструктур. Сквозное управление сервисами. Решение аналитических задач и поддержка принятия решений. Автоматизация процессов и создание информационных панелей. Сопоставление результатов ИТ с результатами БТ
Управление серверами (Server management)	Управление производительностью серверов		
Управление СУБД (DBMS management)	Управление базами данных		
Управление событиями (Event management)	Мониторинг и анализ событий в ИТ-инфраструктуре		
Управление приложениями (Application management)	Управление функционированием приложений		
SLM и BSM	Управление согласованным уровнем обслуживания. Связывание бизнес-целей и приоритетов с задачами ИТ. Управление сервисами с учетом политики бизнеса		
Управление ИТ для пользователей (End user Management)	Сервис рабочих станций. Повышение эффективности работы приложений пользователей. Управление устранением проблем. Качественный доступ к ресурсам ИТ-инфраструктуры		
Управление хранилищами данных (Storage Management)	Обеспечение гарантированного предоставления емкостей в зависимости от потребностей бизнес-процессов. Оптимизация использования имеющихся емкостей		

АИСКП

Служба поддержки (Service desk)	Решение проблем пользователей. Регистрация, анализ и устранение проблем и инцидентов в ИТ-инфраструктуре	Управление ИТ сервисами и процессами (IT service management and process management)	Служба поддержки. Управление взаимоотношениями с заказчиками. Управление выполнением потока операций
Управление ИТ-активами (IT asset management)	Управление жизненным циклом вычислительных и коммуникационных ресурсов ИТ-инфраструктуры	Поддержка ИТ-сервисов и управление ресурсами (IT service support and resource management)	Среднесрочная и долгосрочная эволюция ИТ-активов. Прогнозирование ресурсов. Инициализация ресурсов и ИТ-активов. Финансовый менеджмент
Управление персоналом (Workforce management)	Управление трудовыми ресурсами ИТ-подразделения		
Управление изменениями и конфигурациями (Change and configuration management)	Организация и оптимизация процессов управления изменениями, релизами и конфигурациями		
Управление мощностями (Capacity management)	Экономически эффективное обеспечение ИТ-мощностями, удовлетворяющими текущим и перспективным потребностям бизнеса		
Управление счетами (Billing management)	Управление финансовыми вопросами ИТ-подразделения		
Планирование работ (Job scheduling)	Планирование и управление исполнением программных задач, которые необходимы как часть ИТ-услуг. Автоматизация запуска задач в определенное время дня, недели, месяца или года		

Учет значимости ИТ для бизнеса, заложенный в основу ЛИН-мышления, приводит к необходимости отображения категорий ПО управления ИТ в организационную модель ЛИН-ИТ [239, 290], позволяющую систематизировать инструментарий управления и определить основные направления развития ПО управления ИТ. На рис. 1.13 показаны группы функций и процессов управления и соответствующие им категории управления. Необходимо отметить, что ITMS2 не только проводит конвергенцию и перегруппировку существующих категорий, но и рассматривает новые категории, функции многих из которых представляют собой перегруппировку функций из существующих категорий управления.

АИС КП

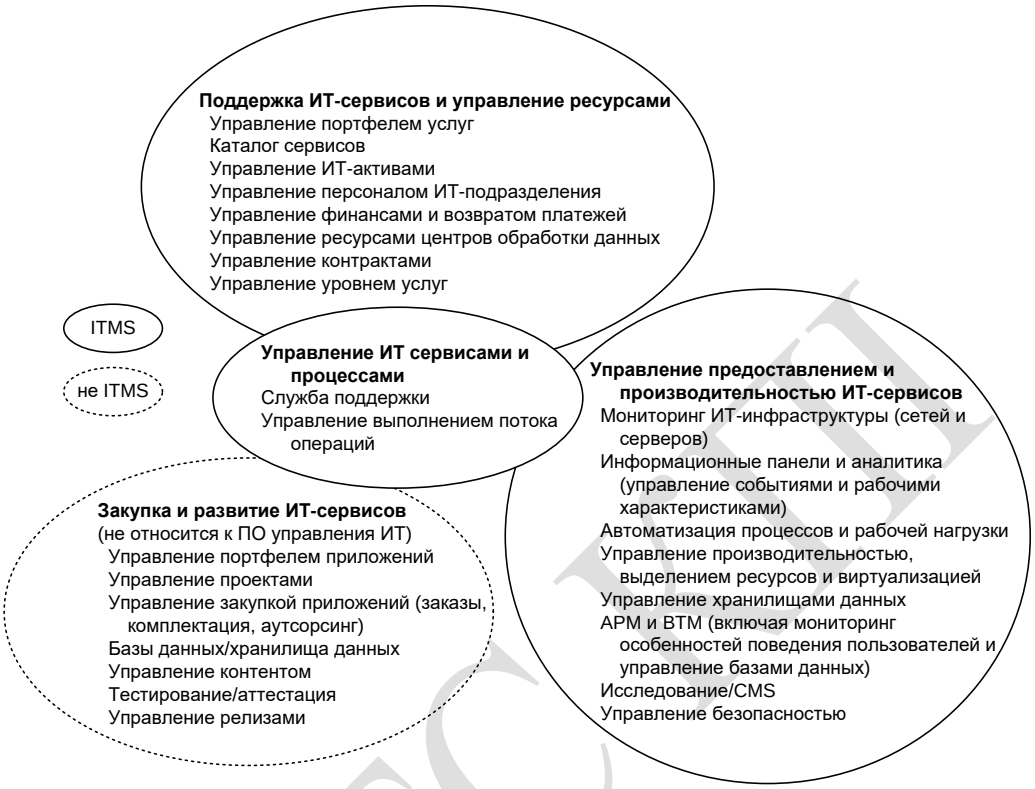


Рис. 1.13. Систематизация инструментария управления ИТ по ITMS2 с учетом ЛИН-модели

Широкомасштабное внедрение ITIL и стремление производителей ПО управления ИТ учитывать концепции ITIL привело к некоторой путанице между процессами и инструментариями и, следовательно, к путанице между инструментариями категорий ПО управления ИТ, составляющими рынок ПО управления ИТ. Этот рынок ориентирован на сервисный подход, а его инструментарий поддерживает сервисные функции. Эти функции поддерживают процессы управления ИТ, но процессами не являются, поскольку они лишь извлекают из процессов информацию и действуют в соответствии с решениями, вытекающими из процессов. Систематизация на рис. 1.13 учитывает это различие в трех главных группах ПО управления ИТ, которые должны сформироваться [239, 290]: управление предоставлением и производительностью ИТ-сервисов, управление ИТ сервисами и процессами, а также поддержки ИТ-сервисов и управление ресурсами.

Управление предоставлением и производительностью ИТ-сервисов. Эта группа призвана объединить аспекты управления ИТ-инфраструктурой, связанные с мониторингом и управлением ресурсами, с целью обеспечения заданных показателей качества функционирования ИТ-инфраструктуры. Она включает весь программный инструментарий, используемый в операционных процессах ИТ-подразделения, которые связаны с предоставлением ИТ-услуг необходимого качества, независимо от того, где располагаются аппаратно-программные средства, предоставляющие эти услуги. Поэтому ПО этой группы должно одинаково успешно управлять доставкой и характеристиками как локально расположенных

сервисов, так и SaaS или IaaS, предоставляемыми облачными технологиями. Все ПО категорий управления этой группы в процессе функционирования использует единые данные и модели сервисов, хранящиеся в CMDB, и снабжает информацией продукты SLM/BSM, которые вырабатывают команды, сигналы или изменяют политику управления ИТ-сервисами в зависимости от значимости бизнес-процессов.

Инструментарий этой группы объединен в восемь основных категорий:

1) категория «Мониторинг ИТ-инфраструктуры» представляет собой перегруппировку всего инструментария, осуществляющего сбор данных о функционировании элементов ИТ-инфраструктуры. Управление сетями и серверами объединено в одну категорию, поскольку в настоящее время стираются различия между традиционными категориями управления технологиями, а также исчезают формальные различия между управлением устранением неисправностями и управлением производительностью;

2) категория «Информационные панели и аналитика (управление событиями и рабочими характеристиками)» является перегруппировкой инструментария управления событиями, рабочими характеристиками и сигналами тревоги. Современные панели меньше ориентированы на обнаружение отказов в отдельных технологиях, а предназначены для выявления первопричин отказов в интегрированных бизнес-услугах. Они могут играть роль накопителей, которые снабжают надлежащими показателями другие инструментарии, такие как бизнес-аналитика;

3) категория «Автоматизация процессов и рабочей нагрузки» предназначена для планирования работ и автоматизации запуска вычислительных задач, которые изначально находятся в ЦОД в свернутом состоянии;

4) категория «Управление производительностью, выделением ресурсов и виртуализацией» оперирует с понятиями, которые в настоящее время рассматриваются в тесной взаимосвязи. Кроме того, широкое использование технологии виртуализации позволяет понять, что производительность является общей характеристикой всех компонентов предоставления сервисов (памяти, вычислительных ресурсов, сетей).

5) категория «Управление хранилищами данных» – неотъемлемая часть СУИ. Особую актуальность эта категория приобретает в связи с раскрывшимися преимуществами виртуализации, когда управление хранением данных становится одним из ключевых элементов управления ИТ-инфраструктурой [221].

6) категория «APM и BTM» – управление производительностью приложений (APM) и управление бизнес-транзакциями (BTM); они относятся к числу важнейших элементов оказания ИТ-услуг. Эта категория должна включать также управление комплексными распределенными приложениями, работающими на мэйнфрэймах или рассредоточенных вычислительных системах, управление обменом сообщениями и мониторинг поведенческих особенностей пользователей. Эта категория формировалась в результате перегруппирования всех компонентов APM в одну категорию, включая контроль поведения пользователей при работе с приложениями, мониторинг J2EE и .NET, мониторинг серверов, отслеживание транзакций, управление производительностью БД и пр.;

7) категория «Исследование и управление конфигурациями» является общей для всего инструментария ИТ-управления. ЛИН-подход не может быть использован без глубокого знания сервисов и их построения. Эти знания нужны не

только для управления ИТ-инфраструктурой, но и для анализа стоимости и значимости сервисов. Поскольку каждый уровень управления использует свой взгляд на активы (аппаратное обеспечение, ПО и сервисы), то, по мнению аналитиков [239], правильным решением будет создание интегрированной системы управления конфигурациями (CMS), а не использование монолитной CMDB. Заполнение CMS требует автоматического исследования приложений, зависимостей и, в конечном итоге сервисов;

8) категория «Управление безопасностью» приобретает большое значение в современных СУИ. Безопасность рассматривается как обширная дисциплина, охватывающая практически все элементы ИТ-инфраструктуры. Однако иногда вопросы безопасности могут негативно влиять на доступность и производительность сервисов. Решение этих проблем должно обеспечивать управление предоставлением сервисов.

Концепция TMN выделяет пять функциональных областей сетевого управления [320], которые вписываются в функции группы управления предоставлением и производительностью ИТ-сервисов. Этими функциональными областями управления являются управления производительностью, устранением неисправностей, конфигурацией, расчетами и безопасностью.

Управление ИТ-сервисами и процессами. Эта группа должна стать платформой коммуникаций между различными процессами управления ИТ и вывести на новый уровень средства автоматизации взаимоотношения ИТ-подразделения с заказчиками сервисов. Рабочие процессы этой категории используются не только для поддержки взаимоотношений пользователей с ИТ-подразделением, но и для поддержки процессов внутри ИТ-подразделения. Взаимодействие между процессами должно осуществляться с помощью ПО управления потоками работ и управления процессами. Служба поддержки должна в режиме реального времени выявлять проблемы в ИТ-инфраструктуре и переадресовать их в центры компетенции, ответственные за решение проблем.

Поддержка ИТ-сервисов и управление ресурсами. Третья группа сосредоточена на управлении ИТ-подразделением. На основе четкого понимания развертываемых сервисов инструментарий этой группы должен способствовать эффективному решению задач планирования сервисов и ИТ-подразделения в соответствии с финансовыми ограничениями, оптимизацией активов и управлением их жизненным циклом, контрактами и взаимоотношениями с поставщиками, оптимизацией решений по построению сервисов и их предоставлению. Критерием для оптимизации решений этих продуктов должно быть соотношение цена/значимость [294]. Данная группа призвана также объединить ПО управления планированием работ, активами, мощностями, изменениями и конфигурациями, финансами ИТ. К этой группе относится инструментарий управления сервисами, превосходящий по функциональным возможностям продукты первой группы и обеспечивающий оптимальное распределение ресурсов, рабочей нагрузки, затрат и др. в зависимости от значимости ИТ-сервисов для бизнес технологий. Например, каталог сервисов обычно содержит иерархическую структуру сервисов, построенную с учетом агрегированных требований пользователей.

Понимание структуры образования стоимости сервисов необходимо для оптимизации портфеля услуг. Оптимизация услуг проводится с учетом потребностей различных бизнес-единиц. Решения по разработке или закупке ПО, а также управление всеми ИТ должно быть основано на финансовых соображениях. Этих

же соображений ИТ-руководители должны придерживаться при управлении контрактами и взаимоотношениями с поставщиками, определяя стратегию развития, удовлетворяющую потребности бизнес-подразделений. Управление инновациями, независимо от того, исходят ли они от бизнеса или из ИТ-подразделения, должно проводиться с оценкой соотношения цена/значимость.

В третьей группе ожидаются наибольшие изменения, связанные с воплощением принципов ЛИН-мышления.

Необходимо отметить, что несмотря на перспективность подход ITMS2 до настоящего времени не получил должного широкого распространения, прежде всего по причине экономического кризиса, несколько затормозившего инвестиции в ИТ. В то же время перегруппирование функций и слияние категорий являются естественным этапом эволюции ПО управления ИТ.

1.4. ПОСТАНОВКА ЗАДАЧ УПРАВЛЕНИЯ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРОЙ

1.4.1. Постановка задачи управления ИТ-инфраструктурой, как задачи управлении уровнем услуг

Для такого объекта управления, как ИТ-инфраструктура, очень сложно определить, формализовать и решить единую задачу управления. Естественной видится декомпозиция общей задачи управления (1.3) на отдельные задачи с последующей их формализацией и комплексным решением.

Важнейшей задачей ИТ-инфраструктуры является предоставление пользователям высококачественных услуг, а эффективность управления в этом случае может быть оценена по качеству $\mathfrak{Q}$ предоставляемых сервисов и затратам на управление. При оперативном управлении ИТ-инфраструктурой задача управления качеством услуг состоит в поддержании заданного уровня качества ИТ-услуг с минимальным количеством используемых для этого ресурсов. Тогда максимальная эффективность управления достигается выбором такого управления, при котором фактический уровень услуг соответствует согласованному $\mathfrak{Q}_{\text{сог}}$ и достигается с минимальными затратами:

$$\text{Затраты}(\mathfrak{Q}(U) \to \mathfrak{Q}_{\text{сог}}) \to \min_{U \in \mathcal{U}}. \tag{1.4}$$

Качество $\mathfrak{Q}$ сервисов определяется качеством всех Q_j, $j = \overline{1,N}$ ИТ-услуг:

$$\mathfrak{Q} = f(Q_1,...,Q_N). \tag{1.5}$$

Следовательно, управляющие воздействия должны поддерживать заданный уровень качества каждой услуги с использованием для этих целей минимального количества ресурсов:

$$\text{Затраты}(Q_j(U) \to Q_j^{\text{cor}}) \to \min_{U \in \mathcal{U}}, \quad j = \overline{1,N}, \tag{1.6}$$

где Q_j^{cor}, $j = \overline{1,N}$ – согласованный уровень j-й услуги.

В отличие от процессного управления, перед которым стоит задача постоянного повышения качества услуг, операционное управление нацелено на поддержание наиболее дешевым способом качества услуг на согласованном уровне, при этом управление должно быть таким, чтобы обеспечивалось

$$q_{k,j} - q^{*}_{k,j} \to 0, \forall j,k, \tag{1.7}$$

где $q_{k,j}$ и $q^{*}_{k,j}$ – соответственно, целевое и фактическое значения *k*-го показателя качества *j*-й услуги.

С точки зрения бизнеса критерием эффективности управления ИТ-инфраструктурой при обеспечении качества *j*-й услуги может быть выбор такого управления $U \in \mathcal{U}$, при котором достигается минимальное фактическое время обработки *i*-го запроса к приложению A_j, $j = \overline{1,N}$ (*N* – количество приложений):

$$\min_{\forall i,j}(T_{\Phi_{i,j}} = (t_{R_{i,j}} - t_{A_{i,j}})), \tag{1.8}$$

где $t_{A_{i,j}}$ – время поступления в ИТ-инфраструктуру *i*-го запроса от пользователей, $t_{R_{i,j}}$ – время поступления ответа пользователю на *i*-й запрос к приложению A_j.

Что касается качества предоставления услуг, то ИТ УИТИ должна осуществлять управление, не руководствуясь критерием (1.8), поскольку невозможно достичь времени обработки запросов, равного 0, а стремиться минимизировать разницу между заданным $T_{З_j}$ и фактическим T_{Φ_j} временем выполнения запросов к *j*-му приложению, измеренным на стороне пользователя, используя наименьшее количество ресурсов:

$$\min_{\forall j}(T_{З_j} - T_{\Phi_j}) = \min_{\forall j}(\Delta T_j), \text{ при } T_{\Phi_j} > T_{З_j}. \tag{1.9}$$

В случае, когда $T_{\Phi_j} - T_{З_j} > 0$, причиной чему может быть увеличение количества запросов пользователей, выполнение критерия (1.8) или достижение (1.9) возможно путем выделения приложению A_j дополнительных ресурсов и/или приоритетного прохождения пакетов данных пользователей приложения A_j по сети. Поэтому авторами уделено должное внимание решению задач управления распределением ресурсов и сетевым трафиком.

1.4.2. Постановка задач исследования

ИТ-инфраструктура включает разнородные составляющие – информационные активы, приложения, средства взаимодействия и техническую инфраструктуру. Управление каждой из этих составляющих сопряжено с рядом специфических проблем, обусловленных индивидуальными особенностями: для информации – постоянным ростом объемов обрабатываемых данных; для приложений – необходимостью интеграции; для средств взаимодействия – необходимостью увеличения полосы пропускания и поддержки новых сервисов; для технической инфраструктуры – возрастающей сложностью. Кроме того, вся ИТ-инфраструктура пронизывается средствами обеспечения безопасности. Для управления

АУТСИ

этими составляющими ИТ-инфраструктуры создаются различные СУ, которые являются либо подсистемами СУИ, либо отдельными системами, но работающими в тесном взаимодействии и под управлением СУИ.

Первые системы административного управления создавались и продавались производителями и поставщиками сетевого оборудования. Большая часть этих систем предназначалась для работы со специфическим аппаратным и программным обеспечением. Модернизация и применение этих систем для управления оборудованием других производителей было или сильно затруднено, или практически невозможно. Во многом это положение сохраняется и по сей день, несмотря на появление дорогостоящих универсальных систем административного управления.

В настоящее время для систем связи производители коммуникационного оборудования выпускают и СУ, ориентированные на управление только отдельным типом оборудования. Такие СУ, использующие различные ноу-хау, носят закрытый характер, могут использовать собственные языки программирования и обычно содержат примитивные средства сопряжения с другими СУ, реализуемые, как правило, в виде обмена файлами с данными конфигураций и результатами мониторинга. Такая же картина имеет место и в области компьютерных технологий.

В современных ИТ-инфраструктурах и ИУС для повышения эффективности бизнес-процессов и управления ими используется большое количество разнообразных информационных и телекоммуникационных технологий и оборудования разных производителей. Эти технологии обеспечивают доступ пользователей к разным ресурсам и в тоже время потребляют часть ресурса для собственных нужд.

Все современные бизнес-технологии имеют разную потребность в разнообразных информационно-телекоммуникационных ресурсах, причем для повышения их эффективности и исходя из экономических соображений для многих бизнес-процессов задействуются одни и те же ресурсы. При этом требуемые объемы ресурса, а также необходимые и запрашиваемые условия предоставления ресурса отдельным приложением могут иметь существенную динамику, независящую от других приложений. В тоже время все ресурсы ИТ-инфраструктуры ограничены, а при одновременном обращении к общим ресурсам всегда возникает конфликт, при разрешении которых средства управления ресурсом не всегда могут отдать предпочтение наиболее важному приложению по причине отсутствия информации о значимости приложений.

Для решения этих проблем необходимо найти комплексное решение, желательно не очень сложное и громоздкое, позволяющее уменьшить проблемы взаимодействия, совместимости различных ИТ, совершенствовать процессы и методы коллективной работы администраторов, ИТ-менеджеров, менеджеров производства, повысить качество работы ИТ, организовать эффективное использование ресурсов и, следовательно, эффективность управления производственными процессами в системе управления бизнесом и эффективности функционирования ИТ-инфраструктуры.

Таким комплексным решением является создание СУИ, включающей совокупность методов, средств и мероприятий, направленных на поддержание работоспособности ИТ-инфраструктуры. СУИ является сетью, наложенной на управляемую ИТ-инфраструктуру. Эффективность работы СУИ определяется применяемыми в ней информационными, коммуникационными и компьютерными

технологиями, а сильная зависимость успешности бизнеса от ИТ требуют, чтобы управление ИТ-инфраструктурой осуществлялось с учетом актуальной важности бизнес-процессов. В настоящее время не известны интегрированные СУИ, позволяющие централизованно управлять всеми подсистемами ИТ-инфраструктуры, а также ИУС, предоставляющие инструментарий ИТ-подразделению для эффективного автоматизированного управления всеми распределенными ресурсами и приложениями.

Поэтому одна из основных задач данной работы состоит в разработке концепции, методов и средств эффективного управления ИТ-инфраструктурой с учетом значимости бизнес-процессов с последующей реализацией всего этого в соответствующей ИТ УИТИ.

Функционал СУИ огромный, ИТ-инфраструктуры могут сильно отличаться. Поэтому создание единой СУИ, реализующей весь перечень задач управления, представляет чрезвычайно сложную, трудоемкую и дорогостоящую задачу.

Приведем некоторые параметры ИТ-инфраструктуры, характеризующие сложность оценивания качества ее функционирования. Например, ИТ-инфраструктура для ОТС может содержать до $n \cdot 10000$ единиц оборудования, каждая единица оборудования может состоять из $n \cdot 100$ компонентов, предоставлять до $n \cdot 10$ сервисов, каждый из которых может быть композицией $n \cdot 10$ услуг, которыми пользуются $n \cdot 1000$ пользователей, для удовлетворения потребностей которых выделяется до $n \cdot 1000$ ресурсов. Причем все перечисленные компоненты обычно характеризуются большим набором значений параметров (до $n \cdot 1000$ параметров).

Управление ИТ-инфраструктурой нельзя рассматривать в отрыве от управления ИТ и качеством ИТ-услуг, поскольку основное предназначение ИТ-инфраструктуры заключается в предоставлении пользователям ИТ-услуг с согласованным уровнем качества. На рис. 1.14 приведены задачи инструментария управления ИТ в соответствии с ITMS2, а фоном выделены задачи оперативного управления.

Современные бизнес-технологии имеют различную потребность в ИТ-ресурсах, причем, исходя из экономических соображений, в условиях консолидации бизнес-процессы задействуют одни и те же ресурсы. Требуемые объемы ресурсов, а также необходимые и запрашиваемые условия их предоставления могут иметь существенную динамику. Ресурсы ИТ-инфраструктуры ограничены и при одновременном обращении приложений к общим ресурсам возникает конфликт, при разрешении которого СУ должны отдавать предпочтение наиболее важному приложению.

С точки зрения управления ИТ-инфраструктура представляет собой сложную систему с четко выраженным функциональным назначением. Критерии управления такими системами должны концентрироваться на эффективности выполнения функций, которых обычно несколько. С целью упрощения управления такими системами они рассматриваются на взаимосвязанных уровнях представления (см. рис. 1.2). Поэтому речь должна идти о системе векторов эффективностей выполнения отдельных функций, объединенных межуровневыми связями. Интеграцию функций управления должна обеспечить ИТ УИТИ, осуществляющая управление во всех трех функциональных областях.

Обобщение изложенного в этой главе материала позволяет сделать вывод о важности и актуальности научно-прикладной проблемы создания проблемно-ориентированной ИТ УИТИ, направленной на поддержание согласованного

уровня ИТ-услуг за счет использования минимальных объемов ресурсов ИТ-инфраструктуры в условиях виртуализации, кластеризации и консолидации вычислительных ресурсов при существенной динамике запросов пользователей.

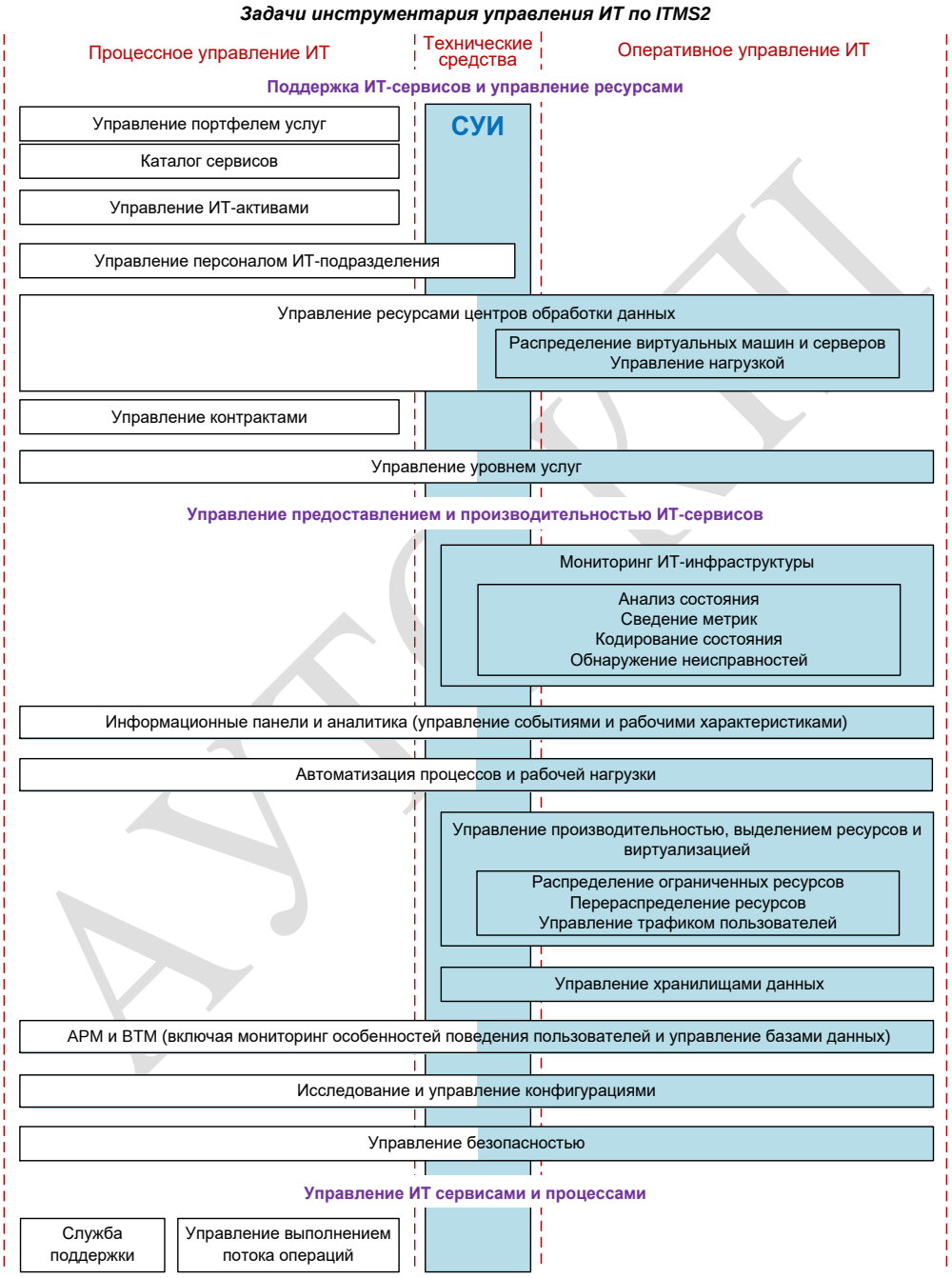


Рис. 1.14. Задачи инструментария управления ИТ-инфраструктурой

Однако создание такой ИТ УИТИ требует решения комплекса сложных задач. Первоочередными представляются следующие задачи разработки:

1) концепции управления ИТ-инфраструктурой;

2) методики управления ИТ-инфраструктурой для поддержания согласованного уровня ИТ-услуг;

3) моделей и методов анализа состояния элементов ИТ-инфраструктуры;

4) методов сведения метрик параметров функционирования ИТ-инфраструктуры;

5) методов обнаружения неисправностей и определения распространения влияния неисправностей на качество услуг;

6) моделей и методов рационального распределения ресурсов ИТ-инфраструктуры с учетом значимости бизнес-процессов;

7) алгоритмов решения задач управления распределением ресурсов, нагрузкой и оптимизации используемых ИТ-ресурсов;

8) информационной технологии управления ИТ-инфраструктурой;

9) системы управления ИТ-инфраструктурой.

Кратко изложим назначения и особенности выделенных задач.

1. Важнейшей задачей, которую нужно решить при создании ИТ УИТИ, является разработка концепции управления такой сложной системой, как корпоративная ИТ-инфраструктура. В настоящее время отсутствует ориентированная на предоставление услуг концепция управления ИТ-инфраструктурой. Учитывая то, что эффективность бизнес-процессов непосредственно зависит от эффективности управления ИТ-услугами и функционирования ИТ-инфраструктуры, разработка концепции управления ИТ-инфраструктурой является первоочередной задачей данной работы.

2. Современный этап развития ИУС характеризуется повышенными требованиями бизнеса к качеству ИТ-услуг. В настоящее время неизвестны методики оперативного управления ИТ-инфраструктур, направленные на поддержание согласованного уровня ИТ-услуг наименее ресурсоемким способом. Поэтому разработке такой методики и ее последующей реализации в ИТ УИТИ уделено особое внимание.

3. Решение задач управления, нацеленных на обеспечение функционирования ИТ-инфраструктуры в соответствии с заданным регламентом или связанных с поддержанием производительности ИТ-инфраструктуры на согласованном уровне, требует знания состояния, производительности и других характеристик ее структурных компонентов. Поэтому для оценки качества функционирования ИТ-инфраструктуры необходимо выбрать или разработать методы и средства определения и анализа состояния элементов инфраструктуры.

4. Наличие большого количества элементов, сложность связей и многоуровневость ИТ-инфраструктуры приводит к необходимости учета анализа $n \cdot 1000$ различных параметров и метрик для измерения и оценки состояния элементов ИТ-инфраструктуры и качества предоставляемых услуг. Метрики верхнего уровня часто имеют качественный характер, характеризуются неопределенностью, нечеткостью и субъективностью. Для формирования сводных метрик уровня услуг необходимо разработать соответствующие модели и методы.

5. При ориентации управления ИТ-инфраструктурой на поддержание согласованного уровня услуг наиболее дешевым способом становятся важными задача обнаружения неисправностей в ИТ-инфраструктуре и оценка их влияния на качество предоставляемых ИТ-услуг. Решение этой комплексной задачи позволит

сократить потери, связанные с неисправностями, за счет сокращения времени на локализацию и устранение неисправностей, что требует разработки соответствующих моделей и методов.

6, 7. Переход от рассредоточенной обработки данных к централизованной обработке в ЦОД требует решения ряда задач и прежде всего разработки методов и алгоритмов эффективного использования дорогостоящих вычислительных ресурсов ЦОД [376], что особенно актуально в условиях повсеместной виртуализации [288] и облачных вычислений [238].

8. Решения предшествующих задач сформируют теоретико-методологические основы создания ИТ УИТИ, обеспечивающей оперативное управление во всех трех функциональных областях с упором на управление уровнем услуг и рациональное использование ИТ-ресурсов. Однако разработка собственно ИТ УИТИ представляет собой сложную в программно-техническом и программно-технологическом аспектах задачу.

9. Разработка эффективной ИТ УИТИ составит основу для создания, внедрения и эксплуатации СУИ. Однако здесь также возникает сложная задача поддержки проектных и реализационных работ, интеграции с другими системами и развития.

ВЫВОДЫ К ГЛАВЕ 1

1. Проанализированы проблемы в области управления ИТ, позволившие определить основные требования к ИТ со стороны бизнеса, заключающиеся в необходимости: предоставления скоординированных услуг с высоким уровнем; повышении экономической отдачи от ИТ; интеграции процессов управления ИТ. Отмечена возрастающая роль ITIL при управлении ИТ, однако принципиальные недостатки не позволяют рассматривать ITIL в качестве методологической основы оперативного управления ИТ-инфраструктурой.

2. Анализ проблем управления ИТ-инфраструктурой показал сложность и многообразие проблем управления ИТ-инфраструктурой, которые обусловлены наличием множества СУ, решающих частные задачи без учета основной задачи управления ИТ-инфраструктурой. Это влечет за собой рост финансовых затрат, увеличение количества ИТ-персонала и повышение вероятности сбоев по мере роста ИТ-инфраструктуры, что отрицательно влияет на эффективность финансово-хозяйственной деятельности предприятий.

3. Определены основные факторы, под воздействием которых будет развиваться управление ИТ в ближайшие годы, среди которых наиболее весомые: консолидация вычислительных ресурсов, технологии виртуализации и облачных вычислений, конвергенции телекоммуникационных услуг. Определены основные тенденции развития управления ИТ и перспективные процессы управления ИТ. Показана необходимость перехода от накопления технологий к накоплению процессов и сервисов.

4. Анализ корпоративной ИТ-инфраструктуры как объекта управления показал, что ее декомпозицию целесообразно проводить выделяя в ней четыре иерархических уровня: сервисы, приложения, вычислительные ресурсы и сетевое взаимодействие. Предложена обобщенная схема корпоративной ИТ-инфраструктуры, определен состав уровней и показано место системы управления ИТ-инфраструктурой.

БГУИР КП

5. Выделено три функциональные области управления, покрывающие весь функционал СУИ: поддержание качества ИТ-сервисов на согласованном уровне; рациональное использование ресурсов ИТ-инфраструктуры; автоматизация операционной деятельности ИТ-подразделения. Показано, что управление ИТ-инфраструктурой, направленное на поддержание согласованного уровня ИТ-сервисов, является критически важным для обеспечения эффективного выполнения бизнес-процессов.

6. Анализ аналогов СУИ, стандартов, протоколов и практик управления ИТ-сервисами и ИТ-инфраструктурой показал, что в полном объеме не решены задачи анализа и сведения метрик качества функционирования элементов инфраструктуры, управления ограниченными ресурсами корпоративной ИТ-инфраструктуры, управления эффективным распределением ресурсов ЦОД, управления потоками, управления поддержанием согласованного уровня услуг. Отсутствие концепции управления ИТ-инфраструктурой сдерживает создание информационной технологии управления корпоративными ИТ-инфраструктурами и разработку инструментария в виде интегрированной системы управления.

7. Обоснованы перспективность и целесообразность использования концепции ПО управления ИТ – ITMS2, согласно которой ИТ-подразделение в совокупности с ИТ рассматривается и управляется как экономически ответственная и финансово прозрачная организация. В управлении ИТ применяется ЛИН-модель, организационное строение ИТ-подразделения осуществляется в соответствии с ЛИН-моделью, классические функции управления ИТ-инфраструктурой объединяются в три группы.

8. На основе выполненного анализа установлено, что консолидация и виртуализация ресурсов, появление облачных вычислений, конвергенция информационно-коммуникационных технологий, существенная зависимость бизнеса от качества услуг определяют новый уровень требований к управлению ИТ-инфраструктурой и требуют создания ИТ УИТИ, построенной на комплексном применении математических моделей и методов анализа, оценивания и управления и ориентированной на предоставление качественных услуг определенного уровня с меньшими затратами.

Аутсорсинг

ГЛАВА 2

КОНЦЕПЦИЯ УПРАВЛЕНИЯ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРОЙ

В главе изложены концепция управления ИТ-инфраструктурой, подход и методика управления уровнем услуг на основе сервисно-ресурсных моделей, комплексный подход к управлению уровнем услуг, построенный на основе интеграции процессного и оперативного управления.

2.1. СУТЬ ПРЕДЛАГАЕМОЙ КОНЦЕПЦИИ

В настоящее время отсутствует единая концепция управления такой сложной системой, какой является корпоративная ИТ-инфраструктура. Не сложилась и концепция создания и интеграции СУИ в существующие и проектируемые ИТ-инфраструктуры. Более того, нет стандартов или рекомендаций, регламентирующих построение и функционирование СУИ, а международный опыт по разработке и внедрению СУ такого типа не предоставил образцов для подражания и обобщения. Кроме того, не известны даже целостные СУ, осуществляющие управление работой функциональных и технологических подсистем ИУС и обеспечивающих при этом эффективное управление и доступ пользователей к общим ресурсам ИТ-инфраструктуры.

В то же время сформировался комплекс факторов, причем их влияние постоянно усиливается, который актуализирует задачу построения СУИ, способной увеличить эффективность использования ИТ-ресурсов и обеспечить максимальную отдачу ИТ для поддержки бизнес-процессов. К таким факторам в первую очередь следует отнести взрывной рост возможностей и номенклатуры ИТ, используемых для построения ИУС. Не меньшее значение имеет сложившаяся зависимость эффективности бизнес-процессов от эффективности управления ИТ-услугами и ИТ-подразделениями.

Поэтому разработку концепции управления ИТ-инфраструктурой и разработку архитектуры СУИ авторы считают необходимым предусловием создания, внедрения и эксплуатации эффективных СУИ. Литературные источники, посвященные этой проблематике, ценны только тем, что подсказывают как не нужно делать. Однако, не охватывая весь спектр задач, которые требуют решения при создании интегрированной ИУС, известные попытки полезны еще и накопленным опытом решения отдельных задач, рассмотрения различных аспектов управления ИТ-инфраструктурой.

В [45] концепция управления ограничивается разделением задач управления по слоям с описанием их функций. При этом выделяются три слоя управления – сетями, системами и информационными сервисами. Эти слои управления включают, соответственно, мониторинг и управление: сетевыми элементами; опера-

ционными системами, базами данных, промежуточным ПО, бизнес-приложениями, обслуживающим персоналом и пользователями; ИТ-инфраструктурой и предоставляемыми сервисами. Концепция, изложенная в [45], ограничивается разделением задач управления по слоям и описанием функций этих слоев, а ее реализация осуществляется за счет интеграции и настраивания ПО управления ИТ компании Hewlett Packard. Однако современный подход к управлению ИТ-инфраструктурой предполагает объединение функций управления сетевыми элементами и серверным оборудованием в едином инструментарии ввиду родственности функций и задач управления этими элементами ИТ-инфраструктуры. Кроме того, в [45] не решаются вопросы интегрированного взаимодействия трех контуров управления. Поэтому предлагаемая в [45] концепция управления ИТ не может быть использована в качестве методологической основы для создания перспективных ИТ УИТИ.

Основные положения предлагаемой концепции управления ИТ-инфраструктурой. Перспективная концепция управления ИТ-инфраструктурой должна давать ответы на обусловленный предшествующим анализом комплекс вопросов, связанный с:

- выделением в ИТ-инфраструктуре иерархических уровней;
- определением и группированием функций управления ИТ-инфраструктурой;
- уточнением общих для всех функций управления процессов с целью создания универсальных программных модулей ИТ УИТИ для СУИ;
- заданием иерархии команд и функций управления;
- спецификацией информационной модели объектов управления;
- построением механизмов взаимодействия ИТ УИТИ с объектами управления.

При этом концепция должна отвечать ряду требований, традиционных для теории систем:

- комплексность;
- сбалансированность;
- непротиворечивость;
- полнота рассмотрения факторов влияния;
- способность к развитию.

Этим требованиям удовлетворяет предлагаемая ниже концепция управления ИТ-инфраструктурой, ориентированная на создание среды интегрированного управления разнообразными ИТ, распределенными приложениями, ресурсами ИТ-инфраструктуры и уровнем услуг.

На концептуальном уровне управление ИТ-инфраструктурой представляется выбором наилучшего решения из набора реализуемых в пространстве возможных. В основе такого видения лежит циклический процесс: выделение задач управления – выбор методов и средств их решения – реализация и воплощение – контроль и анализ – модификация задач (при необходимости). Этот процесс является многократно повторяемым и обучаемым, причем реализуется на взаимодействующих уровнях, предполагая взаимодействие и интеграцию. Такое видение базируется на целостной системе принципов. Поэтому предлагаемая концепция управления ИТ-инфраструктурой, как система путей и методов решения задачи эффективного управления ИТ-инфраструктурой, объединяет следующие принципы:

1. Включение контура оперативного управления ИТ-инфраструктурой во внешний контур управления бизнесом и внутренний контур управления операционной деятельностью.

2. Разбиение ИТ-инфраструктуры на четыре иерархических уровня (сервисов, приложений, вычислительных ресурсов и сетевого взаимодействия).

3. Перегруппирование категорий продуктов управления ИТ в три группы (управление предоставлением и производительностью ИТ-сервисов; управление ИТ-сервисами и процессами; поддержка ИТ-сервисов и управление ресурсами).

4. Выделение трех функциональных областей управления ИТ-инфраструктурой (поддержание качества ИТ-сервисов на согласованном уровне; рациональное использование ресурсов ИТ-инфраструктуры; автоматизация операционной деятельности ИТ-подразделения).

5. Выделение пяти общих процессов при управлении ИТ-инфраструктурой (мониторинг, анализ, управление, оптимизация, планирование).

6. Выделение трех уровней управления (управление ИТ-инфраструктурой, управление приложениями, управление ресурсами).

7. Декомпозиционно-компенсационный способ организации управления уровнем услуг.

8. Интегрированное управление ИТ-инфраструктурой.

9. Учет требований бизнес-процессов.

10. Использование сервисно-ресурсных моделей при управлении уровнем услуг.

11. Внедрение процессно-оперативного подхода к управлению уровнем ИТ-услуг.

12. Использование универсального объекта мониторинга и управления (ОМУ) для систематизации и упрощения работы ИТ УИТИ с реальных (РОУ) посредством логических (ЛОУ) ОМУ.

13. Применение DALL-функций для связи РОУ и ЛОУ.

14. Использование агентского подхода.

15. Применение единого механизма взаимодействия СУИ с агентами и ОМУ.

16. Использование универсальной схемы кодирования состояния ОМУ.

Следует заметить, что некоторые из этих положений не являются новыми, но в предлагаемой концепции они впервые объединены в целостном видении управления ИТ-инфраструктурой. Кроме того, здесь не только систематизированы известные методы и технологии с последующей интеграцией их в единую ИТ УИТИ, но и предлагаются оригинальные методы и подходы как для реализации отдельных компонентов, так их интеграции. Предложенная концепция воплощена в разработанных ИТ УИТИ и СУИ.

По существу, следуя тенденциям развития ИТ, авторы предложили концепцию управления ИТ-инфраструктурой, ориентированную на создание единой универсальной среды или платформы интегрированного управления разнообразными ИТ, распределенными приложениями, ресурсами и сетевым оборудованием корпоративной ИУС и ИТ-подразделением.

Систематизация, интеграция и развитие известных и предлагаемых технологий направлены на решение еще одной важной задачи, связанной с изменением мышления разработчиков и пользователей ИТ УИТИ и СУИ. Поэтому концепция требует отказа от фрагментарного подхода, ориентированного на интеграцию разрозненных СУ, и принятие системного видения, рассматривающего СУИ как целостную систему, подчиненную целям бизнеса.

Рассмотрим более детально положения концепции, не забывая о ее систематизирующем, интеграционном и перспективном характере.

1. Включение контура оперативного управления ИТ-инфраструктурой во внешний контур управления бизнесом и внутренний контур управления операционной деятельностью. Процессы измерения значений параметров, анализа состояния ОМУ и управления ИТ-инфраструктурой в ИТ УИТИ следует формировать и рассматривать в скоординированной целостной системе процессов трех контуров управления: внешнего контура управления бизнесом, внутреннего контура управления операционной деятельностью и контура оперативного управления ИТ-инфраструктурой (см. рис. 1.3).

2. Разбиение ИТ-инфраструктуры на четыре иерархических уровня. Исходя из общности выполняемых функций и услуг предлагается в ИТ-инфраструктуре выделять четыре взаимосвязанных иерархических уровня – сервисов, приложений, вычислительных ресурсов и сетевого взаимодействия (см. рис. 1.2). Выделение уровней осуществлено не только исходя из общности выполняемых функций и сервисов, предоставляемых каждым из уровней, но и исходя из характера решаемых в данной работе задач. Иная совокупность задач может сделать целесообразным другое иерархическое представление корпоративной ИТ-инфраструктуры.

3. Перегруппирование категорий управления ИТ в три группы. Предлагается группировать продукты управления в соответствии с выполняемыми ими функциями согласно перспективной парадигме ITMS2 [239] (см. табл. 1.4 и рис. 1.13). ITMS2 вводит новые категории управления ИТ и предполагает конвергенцию используемых в настоящее время категорий управления ИТ в три группы, содержащих следующий базовый функционал управления:

- управление предоставлением и производительностью ИТ-сервисов (мониторинг ИТ-инфраструктуры (сетей и серверов); информационные панели и аналитика (управление событиями и рабочими характеристиками); автоматизация процессов и рабочей нагрузки; управление производительностью, выделением ресурсов и виртуализацией; управление хранилищами данных; управление производительностью приложений (APM) и управление бизнес-транзакциями (BTM), включая мониторинг особенностей поведения пользователей и управление базами данных; исследование/управление конфигурациями (CMS); управление безопасностью);
- управление ИТ-сервисами и процессами (служба поддержки; управление выполнением потока операций);
- поддержка ИТ-сервисов и управление ресурсами (каталог сервисов; управление: портфелем услуг; ИТ-активами; персоналом ИТ-подразделения; финансами и возвратом платежей; ресурсами ЦОД; контрактами; уровнем обслуживания).

4. Выделение трех функциональных областей управления ИТ-инфраструктуры. Все задачи и функции ИТ УИТИ в соответствии с п. 1.1.10 могут быть отнесены к одной из трех функциональных областей управления ИТ-инфраструктуры: поддержание качества ИТ-сервисов на согласованном уровне; рациональное использование ресурсов ИТ-инфраструктуры; автоматизация операционной деятельности ИТ-подразделения.

5. Выделение пяти общих процессов при управлении ИТ-инфраструктурой. Предлагается при реализации базовых функций всех трех категорий управ-

ления ИТ, сгруппированных в соответствии с ITMS2, выделять пять универсальных процессов: мониторинга, анализа, управления, оптимизации и планирования (МАУОП) [121]. Некоторые из процессов для ряда базовых функций могут иметь нулевую функциональность. Отобрав для каждого из процессов МАУОП набор методов и реализовав его в соответствующем универсальном инструментарии ИТ УИТИ, можно автоматизировать выполнение процессов в различных категориях управления, используя одни и те же методы и инструменты. Систематизация и спецификация процессов МАУОП выполнены в главе 4.

6. Выделение трех уровней управления. Выделяется три иерархических уровня управления – управление ИТ-инфраструктурой, управление приложениями и управление ресурсами (см. п. 2.2.2). Функции, команды и методы управления распределяются по этим уровням. Поэтому необходимо учесть как общие, так специфические аспекты этих уровней.

7. Декомпозиционно-компенсационный способ организации управления уровнем услуг. Декомпозиционно-компенсационный способ организации управления уровнем ИТ-услуг, предложенный и описанный в разделе 3, во-первых, объединяет преимущества децентрализованного и централизованного управления, во-вторых, предлагает схему межуровневого взаимодействия на основе декомпозиции задач управления и эскалации возникающих проблем [105].

8. Интегрированное управление ИТ-инфраструктурой. В настоящее время создание ИТ УИТИ, как правило, осуществляется путем бессистемной интеграции разрозненных приложений управления, не объединенных общими целями и задачами в организационно-технический комплекс управления ИТ-инфраструктурой из единого центра. При этом каждое ПО управления ИТ-инфраструктурой используется по своему функциональному назначению, а их совместную работу трудно подчинить главной цели – повышению эффективности бизнеса.

Перспективным представляется такой подход к построению ИТ УИТИ, когда все подсистемы управления объединяются в единую систему с общими целями и задачами. В этом случае ИТ УИТИ осуществляет не только управление отдельными устройствами, приложениями или ресурсами, а производит интегрированное управление всеми элементами и ИТ-инфраструктурой в целом исходя из единой системы целей. Под интегрированным управлением ИТ-инфраструктурой понимается общее управление различными ИКТ, средствами вычислительной техники, приложениями, ИТ-сервисами, ИТ-персоналом и другим компонентами, исходя из единых целей, определяемых значимостью бизнес-процессов предприятия.

9. Учет требований бизнес процессов. Создание эффективных технологий бизнес-анализа с проекцией на управление ИУС можно рассматривать как своеобразную глобальную обратную связь [171]. Организацию интегрированного управления ИТ-инфраструктурой в комплексе с управлением выполнением бизнес-процессов целесообразно осуществлять на основе систем управления производительностью бизнеса (BpM). Системы BpM призваны автоматизировать бизнес-процессы и анализировать метрики выполнения бизнеса с целью оптимизации производительности его выполнения путем эффективного использования всех финансовых, людских и материальных ресурсов [171]. Процессами BpM обеспечивается обратная связь, когда в результате анализа значений KPI и KQI, а также метрик, характеризующих управление ИТ-услугами, администраторы

ИТ УИТИ и менеджеры бизнеса корректируют работу ИТ-инфраструктуры и выполнение бизнес-процессов. Для этого BpM осуществляет: мониторинг бизнес-процессов; бизнес-анализ для выявления несоответствия показателей бизнес-процессов с показателями эффективного выполнения бизнеса; инициализацию управляющих воздействий для коррекции результативности бизнес-процессов.

Одна из основных задач, решаемых BpM, заключается в координации целей и задач бизнеса со средствами выработки управляющих воздействий на всех иерархических уровнях СУ предприятия. Инструментарий BpM дает возможность бизнесу ставить стратегические и тактические цели, после чего, измеряя основные показатели, характеризующие выполнение бизнес-процессов, производить менеджмент для достижения бизнес-целей. Для этого BpM использует технологии финансового и операционного планирования, моделирования бизнес-процессов, бизнес-аналитики, интеграции приложений, управления и автоматизации выполнения бизнес-процессов. При изменении бизнес-целей или значимости бизнес-процессов ИТ УИТИ должна оперативно реагировать на эти изменения, например, выделением дополнительных ресурсов наиболее важным бизнес-процессам. Для этого средства BpM и ИТ УИТИ должны тесно и оперативно взаимодействовать друг с другом.

10. Использование сервисно-ресурсных моделей при управлении уровнем услуг. Предлагается при управлении уровнем ИТ-услуг использовать сервисно-ресурсные модели, связывающие ресурсы и сервисы и, тем самым, позволяющие задачи управления бизнесом через управление бизнес-процессами свести к задачам управления ИТ-инфраструктурой. Более детально подход описан в п. 2.3.

11. Внедрение процессно-оперативного подхода к управлению уровнем ИТ-услуг. Комплексное управление уровнем услуг в сочетании процессного и оперативного (автоматического или автоматизированного) управлений позволяет снизить число нарушений SLA, обеспечивая поддержание согласованного уровня услуг.

12. Использование универсального ОМУ для систематизации и упрощения работы ИТ УИТИ с РОУ посредством ЛОУ. В данной работе введено понятие универсального ОМУ, а взаимодействие СУИ проводится не с реальными (РОУ), а с логическими ОМУ (ЛОУ). ЛОУ привязываются и ассоциируются не только с РОУ, но и с функциями, состояниями процессов деятельности, подсистемами ИТ-инфраструктуры, процессами деятельности, бизнес-процессами и пр., а также со связями в ИТ-инфраструктуре, физическими соединениями, с администраторами и другим персоналом, принимающим участие в управлении ИТ-инфраструктурой. ЛОУ является моделью РОУ, отражающей свойства, которые относятся к управлению ИТ-инфраструктурой. В качестве ЛОУ могут быть абстрактные объекты, не имеющие РОУ, введенные в информационную модель для обобщения информации мониторинга нескольких ЛОУ. Такой подход позволяет использовать общие универсальные методы работы ИТ УИТИ с ЛОУ, а также универсальные методы оценки состояния ЛОУ.

Использование шаблонов ОМУ при отнесении ОМУ и ЛОУ к классам. При конфигурации предлагаемой СУИ используется механизм шаблонов ОМУ и ЛОУ, позволяющий настраивать поведение и структуру шаблонного ЛОУ, а при создании новых ЛОУ использовать настройки шаблона, при необходимости уточняя и дополняя их. При этом можно легко изменять поведение и структуру группы ЛОУ, изменяя настройки их шаблона.

13. Применение DALL-функций для связи РОУ с ЛОУ. Для связи ЛОУ и РОУ используется предлагаемый механизм DALL-функций, основанный на рефлексии типов. DALL-функции являются передаточными звеньями при взаимодействии ЛОУ и РОУ. Путем вызова соответствующих DALL-функций мониторинга проводится синхронизация состояния ЛОУ с состоянием РОУ, а в случае оказания на ЛОУ управляющих воздействий эти воздействия передаются РОУ путем вызова соответствующих управляющих DALL-функций. Механизм DALL-функций описан в разделе 4.1.

14. Использование агентского подхода. Предлагаемая ИТ УИТИ построена на агентской технологии. Агентскую технологию можно рассматривать как промежуточный этап при переходе от централизованного к перспективному децентрализованному управлению ИТ-инфраструктурой. Одним из основных и кардинальных способов сокращения сетевой нагрузки, создаваемой ИТ УИТИ при мониторинге и управлении ИТ-инфраструктурой, является применение агентской технологии с использованием агентов и метаагентов [14, 117, 164, 234]. Особенности реализации агентского подхода рассмотрены в главе 4.

15. Применение единого механизма взаимодействия СУИ с агентами и ОМУ. Для взаимодействия СУИ с распределенными подсистемами СУИ, ОМУ и агентами целесообразно использовать единые методы и протоколы. Например, для обмена информацией между модулями СУИ в среде Windows целесообразно задействовать компоненты .NET Remoting или пришедший им на смену WCF – программный фреймворк, используемый для обмена данными между приложениями и входящий в состав .NET Framework 3.0.

16. Использование универсальной схемы кодирования состояния ОМУ. Предложен универсальный метод кодирования состояния ОМУ, позволяющий представлять оценку состояния ОМУ символами кода для возможности измерения качественных свойств ОМУ и упрощения последующей обработки при сведении метрик. Предлагаемый метод кодирования описан в главе 5.

На основании предложенной концепции создана ИТ УИТИ и СУИ, применение которой позволяет увеличить эффективность и надежность использования ИКТ и ИТ-ресурсов, а также повысить производительность ИТ-инфраструктуры, что положительно влияет на выполнение бизнес-процессов предприятия. Приведенный ниже материал поясняет, раскрывает или дополняет суть предложенной концепции управления ИТ-инфраструктурой.

2.2. РАЗВИТИЕ ПАРАДИГМЫ ITMS2

2.2.1. Особенности выделения пяти общих процессов при управлении ИТ-инфраструктурой

Повышение конкурентоспособности предприятия за счет владения ИТ-инфраструктурой с высокой производительностью и низкими операционными затратами можно обеспечить внедрением ИТ УИТИ. При разработке ПО управления ИТ-инфраструктурой целесообразно руководствоваться парадигмой ITMS2 [239], принятой на вооружение многими производителями ПО управления ИТ [244].

По сути, ITMS2 предполагает объединение существующих категорий ПО управления ИТ в три группы с перегруппировкой функций этих категорий с добавлением новых перспективных категорий. Необходимо отметить, что конвергенция категорий управления с последующим перегруппированием выполняемых функций категорий ПО управления ИТ по признаку однородности является одной из характерных черт современного этапа эволюции управления ИТ-инфраструктурой. Так, категории сетевого управления и управления серверами объединены в категорию мониторинга ИТ-инфраструктуры, которая должна осуществлять управление не только всем парком сетевого и серверного оборудования, но и других элементов ИТ-инфраструктуры. Такое группирование достаточно разнородных элементов вполне оправданно, поскольку для управления этими элементами применяются не только одни и те же подходы к управлению, но и одни и те же механизмы и технологии управления.

Кроме того, это предопределяет использование универсальных протоколов управления, таких как SNMP, который создавался как протокол сетевого управления, но позже стал использоваться и для управления самыми разнородными устройствами. В этом случае удаленное управление любыми аппаратно-программными элементами, подключенными к сети, становится возможным после обеспечения поддержки ими SNMP и создания соответствующих библиотек – MIB.

Еще одним фактором конвергенции категорий управления, способствующим созданию универсального инструментария, является повсеместное внедрение ИТ для автоматизации управления ИТ-инфраструктурой.

Конвергенция категорий управления с последующим перегруппированием выполняемых функций категорий ПО управления ИТ по признаку однородности представляется вполне естественной и подталкивает на поиск схожих концептуальных решений, которые могут стать основой создания перспективных подходов к управлению ИТ-инфраструктурой.

Учитывая, что каждая категория управления содержит одинаковые процессы или операции, например, мониторинг или анализ, приходим к идее выделить и объединить одинаковые операции родственных процессов из всех категорий управления всех групп с последующей реализацией этих операций в едином инструментарии, который будет использоваться для всех категорий ITMS2. Такой подход кажется естественным продолжением тенденций развития ПО управления ИТ-инфраструктуры, рассмотренных в ITMS2.

Эти процессы должны удовлетворять следующим требованиям и свойствам:

- давать ответы на следующие вопросы, касающиеся состояния ИТ-инфраструктуры и ее компонентов: «Что сейчас происходит в ИТ-инфраструктуре?», «Почему это происходит?», «Что нужно сделать, чтобы сохранить штатное состояние?»; «Что можно сделать, что бы ИТ-инфраструктура функционировала эффективнее?», «Как развивать ИТ-инфраструктуру?»;
- обладать способностью работы в реальном или близком к реальному масштабу времени и пр.

При рассмотрении вопросов ИТ-управления выделяют общие циклы управления или вводят в рассмотрение жизненные циклы процессов управления. При процессном подходе часто используется цикл Шухарта-Деминга [102], представляющий собой повторяющийся процесс принятия решения и обозначаемый PDCA (Plan-Do-Check-Act). Цикл PDCA является последовательностью действий

для достижения целей процесса. Цикл включает этапы: планирования работ и ресурсов, необходимых для достижения целей; выполнения; проверки; воздействия с целью устранения причин отклонений от запланированного результата. После чего повторяется этап планирования с изменения планов и распределения ресурсов. Цикл PDCA многократно выполняется с разной периодичностью, обычно совпадающей с периодичностью циклов отчетности ИТ-подразделения. При выполнении корректирующих действий длительность цикла может быть сокращена в зависимости от мероприятий по устранению причин отклонения. Цикл PDCA относится к организационному управлению, применим для управления качеством продукции, но совершенно не пригоден для оперативного управления ИТ-инфраструктурой.

Методология управления, контроля и аудита информационных систем COBIT [215] делит всю деятельность ИТ-подразделения на четыре сферы, включающие 34 процесса, каждый из которых относится к ИТ-цели. Каждая ИТ-цель привязана к цели бизнеса. Каждый процесс содержит метрики и систему оценки. Стандарт выделяет четыре группы процессов: планирования и организации; приобретения и внедрения; эксплуатации и сопровождения; мониторинга и оценки. COBIT больше подходит для аудита деятельности ИТ-подразделения, чем для организации управления ИТ-инфраструктурой.

Международный процессно-ориентированный стандарт по управлению ИТ-сервисами ISO 20000 [262, 263] определяет требования к системе управления ИТ, содержащей политики и подходы обеспечения эффективного управления и внедрения ИТ-сервисов. Стандарт определяет требования к 13 процессам, объединенным в пять групп: предоставления сервисов (управление уровнем сервиса, отчетность по предоставлению сервисов, управление непрерывностью и доступностью; бюджетирование и учет затрат для ИТ-сервисов, управление мощностями, управление информационной безопасностью); взаимодействия (управление взаимодействием с бизнесом, управление подрядчиками); разрешения (управление проблемами, управление инцидентами); контроля (управление конфигурациями, управление изменениями); управления релизами. Процессы стандарта ISO 20000 относятся в основном к организационному управлению ИТ-подразделением и только косвенно затрагивают процессы оперативного управления ИКТ.

Методика Six Sigma [170] при реализации проектов использует последовательность этапов DMAIC (define, measure, analyze, improve, control – выявить, измерить, проанализировать, усовершенствовать, проконтролировать). Этапы DMAIC предназначены для: определения целей проекта и запросов потребителей, как внутренних, так и внешних; измерения процессов для определения хода выполнения; анализа и определения первопричин дефектов; улучшения процесса сокращением дефектов; контроля дальнейшего протекания процесса. Данная методика совершенствования бизнес-процессов в основном предназначена для минимизации вероятности возникновения дефектов в операционной деятельности при производстве продукции и предоставлении услуг. Этапы DMAIC можно использовать в операционной деятельности ИТ-подразделения и лишь частично при управлении ИТ-инфраструктурой.

Ни один из проанализированных выше наборов общих процессов различных стандартов не удовлетворяет требованиям к общим процессам управления ИТ-инфраструктурой. Поэтому при реализации базовых функций всех трех катего-

рий управления ИТ, сгруппированных в соответствии с ITMS2, предлагается выделить пять универсальных процессов: мониторинга, анализа, управления, оптимизации и планирования (МАУОП) (рис. 2.1).

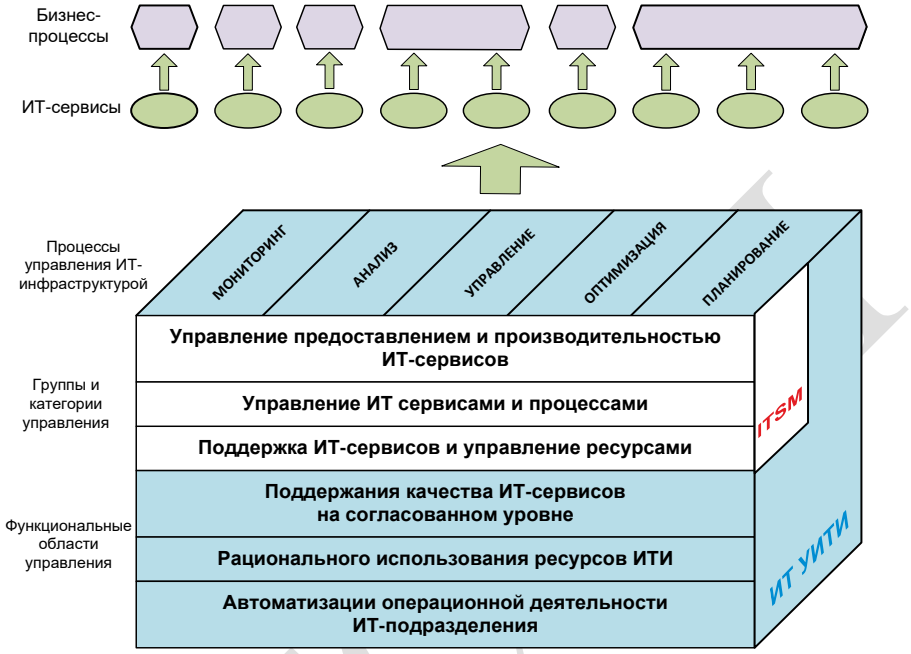


Рис. 2.1. Универсальные процессы управления ИТ-инфраструктурой в ИТ УИТИ

Отобрав или разработав для каждого из процессов МАУОП набор универсальных методов и реализовав его в соответствующем инструментарии ИТ УИТИ, можно автоматизировать выполнение процессов в различных категориях управления, используя одни и те же методы и инструменты.

Такой подход, предполагающий выделения всего пяти ключевых процессов, позволит постепенно наращивать функциональность, не включая каждую категорию управления полностью, а используя только актуальные функции. Это даст возможность отследить только ключевые для организации ИТ-ориентиры и сократить затраты на содержание ИТ-инфраструктуры.

Объединение не функций, как это сделано в ITMS2, а методов, операций, процессов или видов деятельности, таких как планирование, позволит при практической реализации всех категорий ПО управления ИТ использовать готовые наборы универсальных моделей, алгоритмов, методов, модулей и механизмов МАУОП.

Процессы МАУОП, приведенные на рис. 2.2, можно определить следующим образом:

1. Мониторинг – сбор, измерение, первичная обработка и сравнение информации, отчетность. Отвечает на вопросы: «Что сейчас происходит в ИТ-системе?» и «Что происходило?». Мониторинг для всех категорий целесообразно осуществлять единым инструментарием. Единый инструментарий мониторинга

для всех категорий (например, агент) может следить сразу за всеми ИКТ, элементами ИТ-инфраструктуры, работой приложений и другими составляющими, находящихся в зоне его ответственности, и осуществлять управление ими.

2. Анализ – анализ результатов мониторинга, состояния компонентов и причин возникновения неисправностей. Отвечает на вопрос «Почему это происходит?». Набор различных методов анализа, реализованный в универсальном инструментарии, позволит выявлять тенденции поведения разнообразных элементов ИТ-инфраструктуры, упреждать неисправности в ИТ-инфраструктуре, реализовать проактивное управление.

3. Управление – включает все аспекты управления элементами ИТ-инфраструктуры, ИКТ, ТКС, предоставлением ИТ-сервисов и пр. Отвечает на вопрос: «Что нужно сделать, чтобы все работало хорошо, работало лучше или хотя бы не хуже?»

4. Оптимизация – оптимизация использования ресурсов ИТ-инфраструктурой. Отвечает на вопрос: «Что можно сделать, чтобы стало лучше?»

5. Планирование – прогнозирование поведения ИТ-системы, планирование развития ИТ-инфраструктуры. Отвечает на вопросы: «Что произойдет?», «Что мы хотим получить или достичь?» и «Как развивать ИТ-инфраструктуру?». Инструментарий планирования также целесообразно объединить для разных процессов. Например, планирование при расширении предприятия вызовет необходимость расширения сети, вычислительных мощностей, привлечение дополнительного ИТ-персонала и пр.

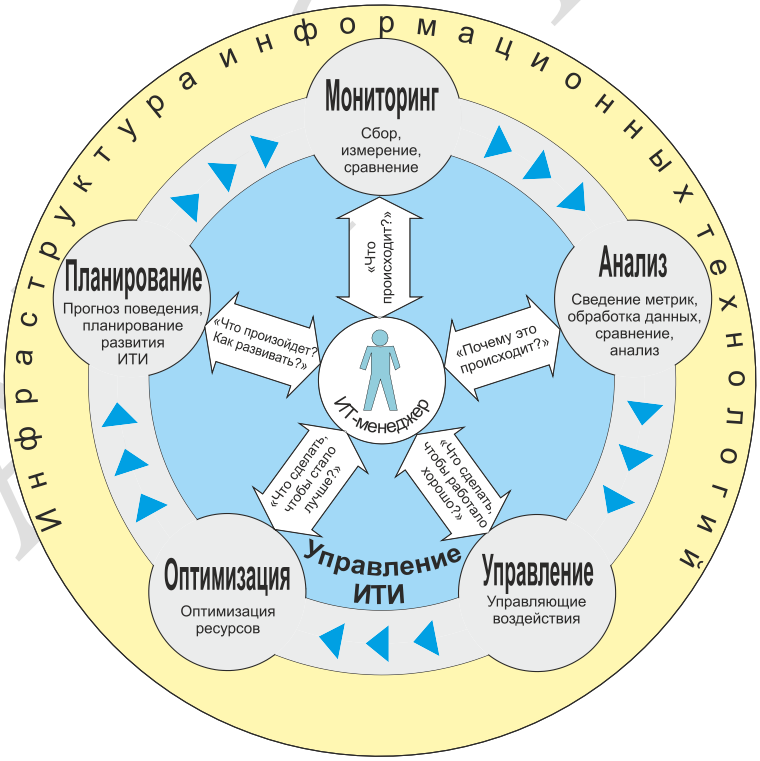


Рис. 2.2. Последовательность выполнения процессов МАУОП

Процессы МАУОП применяются во всех трех функциональных областях управления ИТ-инфраструктурой (рис. 2.3).

Необходимо учитывать, что при управлении ИТ-подразделением процессы МАУОП часто выполняются вручную, как, например, большая часть процессов управления документацией [10], которая в соответствии с ITIL использует такой показатель, как процент документов, не просматривавшихся в течение года, и другие показатели, измеряемые с периодичностью квартал, полугодие или год. Автоматизация таких процессов часто оказывается нецелесообразной прежде всего по экономическим соображениям. В таких случаях руководство ИТ-подразделения выполняет мысленно процессы МАУОП или часть этих процессов, в результате которых выдаются указания на пересмотр политики работы с документацией, приводящей к тому, что, например, создаются дополнительные копии документов, документы подвергаются существенному редактированию или некоторые категории документов убираются далеко в архив или удаляются.

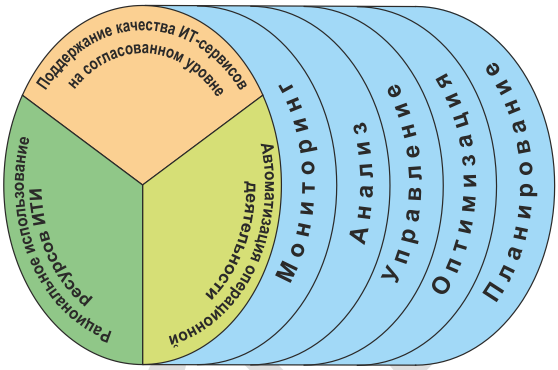
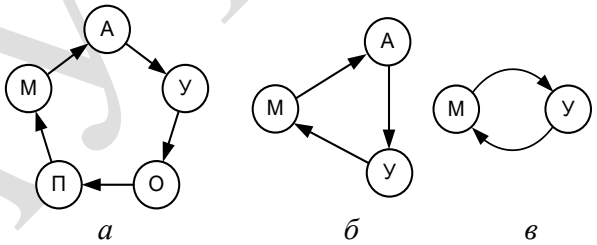


Рис. 2.3. Место процессов МАУОП в функциональных областях управления ИТ-инфраструктурой

Естественно, что не для всех элементов ИТ-инфраструктуры и процессов управления необходимо выполнять все процессы МАУОП (рис. 2.4). Так, при управлении серверами и сетевым оборудованием цикл управления обычно включает в себя только два процесса – мониторинга и управления (МУ) (рис. 2.4,*в*), а управление большинством аппаратно-программных средств ограничивается только циклом – мониторинг, анализ и управление (МАУ) (рис. 2.4,*б*).



а *б* *в*

Рис. 2.4. Жизненные циклы управления: *а* – полный цикл; *б* и *в* – сокращенные циклы

Сохраняя общность, будем считать, что жизненный цикл управления ИТ-инфраструктурой всегда содержит пять процессов МАУОП, часть из них не используется или может выполняться только время от времени. Когда некоторые из процессов не используются, то считается, что эти процессы все равно выполняются, но имеют нулевую функциональность.

2.2.2. Выделение трех иерархических уровней управления

Поскольку ИТ-инфраструктура имеет явно выраженную иерархическую структуру (см. рис. 1.2), то средства управления таким сложным и громоздким объектом управления также целесообразно строить по иерархическому принципу. Для этого необходимо выделить иерархические уровни управления, рационально распределить функции управления, осуществить дифференциацию команд и методов управления. В ИТ УИТИ иерархических уровней управления может быть больше или меньше, чем иерархических уровней ИТ-инфраструктуры, кроме того, уровни управления могут не совпадать с иерархическими уровнями ИТ-инфраструктуры.

В данной работе используется подход, предполагающий рассмотрение в тесной взаимосвязи управления аппаратно-программными средствами компонентов уровней вычислительных ресурсов и сетевого взаимодействия. Это подразумевает, что команды управления сетевым оборудованием и серверами будут относится к одной категории несмотря на то, что эти компоненты относятся к разным уровням иерархической структуры ИТ-инфраструктуры.

Компания Cisco использует модель, предполагающую выделение трех уровней управления – сервисами, сетью и сетевыми элементами [214]. Концепция TMN специфицирует управление сетевыми элементами, сетью, услугами и бизнес-процессами ОТС [320]. В [45] выделяется три слоя управления ИТ-инфраструктурой – сетями, системами и сервисами. Парадигма ITMS2 предполагает создание единого инструментария для управления элементами уровней вычислительных ресурсов и сетевого взаимодействия. Это означает, что команды управления сетевым оборудованием и серверами будут относится к одной категории, несмотря на то, что эти компоненты относятся к разным уровням иерархии ИТ-инфраструктуры. Подобным образом, ввиду родственности решаемых задач, следует объединить управление сервисами и приложениями.

При управлении корпоративными ИТ-инфраструктурами целесообразно выделить три уровня управления – ИТ-инфраструктурой, приложениями и ресурсами.

Множество $\hat{\mathcal{U}}$ выполняемых ИТ УИТИ команд управления, множество используемых в ИТ УИТИ $\hat{\mathfrak{M}}$ методов управления, а также множество $\hat{\mathcal{F}}$ всех функций управления необходимо отнести к этим уровням управления (табл. 2.1).

Управление осуществляется с помощью методов управления, которые образуют множество

$$\hat{\mathfrak{M}} = \{\hat{M}_c, \hat{M}_n, \hat{M}_p\}, \tag{2.1}$$

где $\hat{M}_c, \hat{M}_n, \hat{M}_p$ – методы управления соответственно ИТ-инфраструктурой, приложениями и ресурсами.

Архитект

Таблица 2.1. Уровни управления ИТ-инфраструктурой

Уровень управления	Функции управления	Команды	Методы
Управление ИТ-инфраструктурой	$\hat{F}_c$: Управление функционированием ИТ-инфраструктуры. Организационное управление ИТ-подразделением. Управление СУИ. Координация нижних уровней	$\hat{U}_c$	$\hat{M}_c$
Управление приложениями	$\hat{F}_n$: Управление приложениями и их взаимодействием. Оперативное управление ИТ-инфраструктурой для поддержания заданного уровня услуг. Управление пользователями	$\hat{U}_n$	$\hat{M}_n$
Управление ресурсами	$\hat{F}_p$: Управление распределением и доступом к ресурсам. Управление компонентами подсистем, доступом пользователей к ресурсам и службам, управление элементами ИТ-инфраструктуры	$\hat{U}_p$	$\hat{M}_p$

Стратификация функций, методов и команд управления ИТ-инфраструктурой по трем иерархическим уровням проведена в п. 1.1.5.

Соотношение между уровнями управления, уровнями иерархии ИТ-инфраструктуры и управляемыми компонентами ИУС представлено на рис. 2.5.

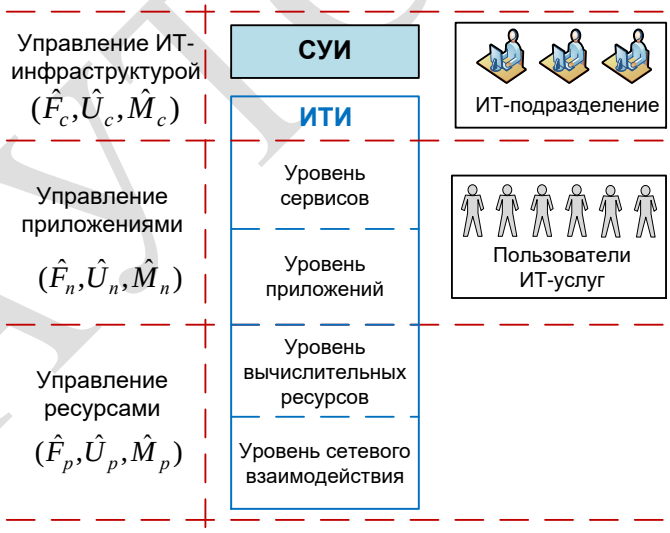


Рис. 2.5. Соотношение между уровнями управления ИТ-инфраструктурой

Функции множества $\hat{F}_c \in \hat{\mathfrak{F}}$ и команды управления множества $\hat{U}_c$ уровня управления ИТ-инфраструктурой обеспечивают поддержание функционирования ИТ-инфраструктурой в соответствии с заданным регламентом, управление

АУТСКИ

функциональными подсистемами, бизнес-процессами ИТ-подразделения и организационное управление ИТ-инфраструктурой, координацию, т. е. согласованное управление работой нижних уровней управления для достижений общей цели функционирования ИУС. К этому уровню относится персонал, принимающий концептуальные и стратегические решения. На этом уровне общие цели и задачи, стоящие перед ИУС, преобразуются в задачи и целевые значения для нижних уровней, перераспределяются ресурсы ИТ-инфраструктуры и принимаются решения в нештатных ситуациях. На этом уровне команды управления могут реализовываться путем выработки соответствующих политик управления.

Множество $\hat{U}_n$ составляют команды управления приложениями и взаимодействием между распределенными приложениями, услугами и поддержанием согласованного уровня предоставления услуг, отдельными типами ТКС и пользователями. Здесь используется программное управление в соответствии с целевыми значениями, задаваемыми уровнем управления ИТ-инфраструктурой.

Команды из множества $\hat{U}_p$ управляют вычислительными ресурсами и сетевым взаимодействием, компонентами приложений, доступом пользователей к информационным и телекоммуникационных ресурсам, службами, серверами, ПК, оборудованием ТКС, вспомогательным оборудованием, ОС и другими элементами. Управление сводится к программному изменению параметров ОМУ в соответствии со значениями, задаваемыми вышерасположенными уровнями управления.

Реализация команд расположенного выше уровня, как правило, осуществляется формированием наборов команд нижележащих уровней:

$$\hat{U}_c \rightarrow \hat{U}_n \rightarrow \hat{U}_p. \qquad (2.2)$$

Например, функции управления, определенные для ТКС стандартом X.700 и включающие управление устранением неисправностей, учетом, конфигурациями, производительностью и безопасностью, реализовываются командами управления всех трех уровней с учетом специфики управляемых ОМУ.

На рис. 2.6 показано взаимоотношение между функциональными областями управления, уровнями и процессами управления. Как видно из рисунка, для достижения цели создания ИТ УИТИ необходимо осуществлять управление во всех функциональных областях. Причем управлять нужно на всех уровнях иерархии ИТ-инфраструктуры, выполняя процессы МАУОП.

Каждый уровень управления должен быть реализован не просто отдельной СУ, а представлять собой часть интегрированной платформы управления ИТ УИТИ, позволяющей наращивать как собственные возможности, так и легко включать в себя ИТ, предназначенные для решения отдельных задач управления. Например, для управления ресурсами могут быть задействованы СУ сетевыми устройствами, разработанные производителями сетевого оборудования, ПО, созданное для управления СУБД, серверами и пр. Для решения задач МАУОП может быть использовано ПО СУ, номенклатура которого определяется характером бизнеса, масштабом предприятия, используемым оборудованием, ИТ и другими факторами. Так, функции мониторинга и управления на нижних уровнях иерархии ИТ-инфраструктуры могут выполняться универсальными СУ, такими как HP Operations Manager, IBM Tivoli, Microsoft System Center Operations Manager (SCOM) и др., или же СУ, разработанными и поставляемыми производителями

ресурсов, например, Oracle, или разработчиками решений в одной из функциональных областей ИТ.

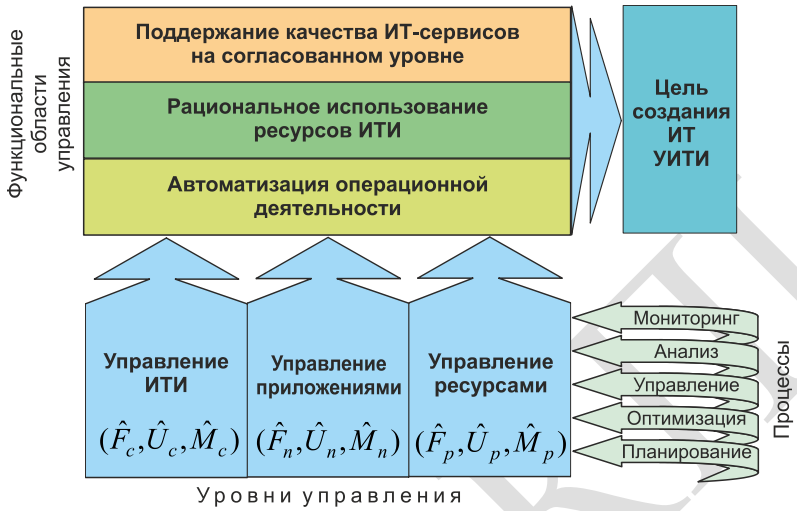


Рис. 2.6. Взаимосвязь между функциональными областями, уровнями и процессами управления для достижения цели создания ИТ УИТИ

Команды управления из $\hat{\mathcal{U}}$ применяются для реализации процессов МАУОП и могут вырабатываться автоматически ИТ УИТИ с использованием универсальных механизмов, алгоритмов и протоколов, автоматизировано – посредством СУИ, либо вручную администраторами.

Самые сложные задачи управления решаются при выполнении функций управления из множеств $\hat{F}_c$ и $\hat{F}_n$, реализуемых выполнением команд управления $\hat{U}_c$ и $\hat{U}_n$. Централизация управления распределенными приложениями и ресурсами ИТ-инфраструктуры с целью поддержания их корректной совместной работы, направленной на обеспечение максимальной эффективности бизнес-процессов, предполагает автоматизацию решения множества проблем из трех функциональных областей управления ИТ-инфраструктурой. При этом фактически безграничная сфера применения и деятельности ИУС и ИТ-инфраструктуры, а также многообразие используемых в них технологий, ресурсов, оборудования, приложений и других элементов, осуществляющих поддержку множества бизнес-процессов, делают невозможным создание единой универсальной СУ абсолютно всеми ИТ-инфраструктурами. Поэтому целесообразно разрабатывать СУИ, областью применения которых являются достаточно широкие классы ИУС и ИТ-инфраструктур с родственной функциональностью, имеющие в своем арсенале средства, обеспечивающие взаимодействие между отдельными СУ и характеризующиеся простотой наращивания функциональности.

В настоящее время отсутствует ПО, покрывающее весь функционал управления $\hat{\mathcal{F}}$. Из такой ситуации возможны два выхода: разработка оригинальной ИТ УИТИ или закупка ПО, решающего отдельные задачи управления или части задач, с их последующей интеграцией в ИТ УИТИ.

2.2.3. Расширенная базовая модель управления ИТ-инфраструктурой

Расширенная базовая модель ИТ-инфраструктуры, уточняющая и развивающая модель, изображенную на рис. 1.4, приведена на рис. 2.7. Она иллюстрирует процесс выбора команд управления в традиционных для теории управления представляющих.

На вход ИТ-инфраструктуры поступает поток запросов $\tilde{\Omega}$, в результате которого на выходе создается поток результатов $\tilde{\Re}$. Информация о значениях параметров, характеризующих свойства потока запросов $\tilde{\Omega}$ и показателей потока результатов $\tilde{\Re}$, в виде потоков значений q и $\tilde{t}$, соответственно, поступает в СУИ. На ИТ-инфраструктуру воздействуют возмущающие воздействия $\tilde{\Xi}$, некоторые параметры которых могут фиксироваться СУИ (на рис. 2.7 не показано). В СУИ анализируется поток данных $\tilde{\mathcal{G}}$ о состоянии ОМУ, а также потоки q и $\tilde{t}$, в результате чего вырабатывается поток команд управления $\tilde{\mathcal{U}}$, поступающий на ИТ-инфраструктуру для поддержания параметров функционирования на заданном уровне.

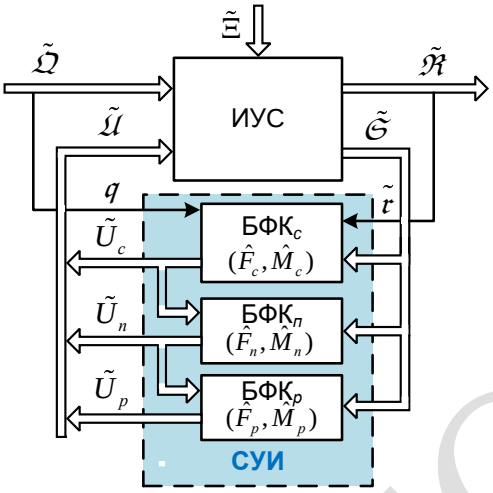


Рис. 2.7. Расширенная базовая модель управления ИТ-инфраструктурой

Поскольку поток данных о состоянии ОМУ $\tilde{\mathcal{G}}$ содержит сведения об ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, и в СУИ реализована иерархия команд, причем выработка команд верхних уровней влечет за собой выбор команд нижних уровней, поток команд управления $\tilde{\mathcal{U}}$ будет состоять из суммы потоков команд в общем случае всех категорий управления:

$$\tilde{\mathcal{U}} = (\tilde{U}_c + \tilde{U}_n + \tilde{U}_p), \qquad (2.3)$$

где $\tilde{U}_c, \tilde{U}_n, \tilde{U}_p$ – соответственно потоки команд управления ИТ-инфраструктурой, приложениями и ресурсами.

Каждый из потоков команд $\tilde{U}_c, \tilde{U}_n, \tilde{U}_p$ вырабатывается соответствующим блоком формирования команд БФК$_c$, БФК$_n$ или БФК$_p$, каждый из которых ориентирован на реализацию соответствующих функций управления $\hat{F}_c, \hat{F}_n$ или $\hat{F}_p$ и использует соответствующие методы управления $\hat{M}_c, \hat{M}_n$ или $\hat{M}_p$.

Команды управления, формирующие поток $\tilde{\mathcal{U}}$, выбираются блоками БФК$_c$, БФК$_n$ и БФК$_p$ из множества $\hat{\mathcal{U}}$ на основании анализа данных о состояния элементов ИТ-инфраструктуры; эти данные содержатся в потоке $\tilde{\mathcal{G}}$ и с учетом значений параметров потоков q и $\tilde{t}$

$$\tilde{\mathcal{U}} = \mathcal{F}(\tilde{\mathcal{S}}, q, \tilde{r}). \quad (2.4)$$

Потоки команд управления нижних уровней вырабатываются под воздействием потоков команд управления расположенных выше уровней, при этом дополнительно учитывается состояние всех ОМУ $O_{i,j}$, $i = \overline{1, I_j}$, $j = \overline{1, J}$.

Таким образом, поток команд расположенных выше уровней инициирует дополнительный поток команд расположенных ниже уровней иерархии. В то же время команды из множества $\tilde{U}_p$ могут вырабатываться самостоятельно, как реакция СУИ на изменение состояния ОМУ.

2.3. МЕТОДИКА УПРАВЛЕНИЯ УРОВНЕМ ИТ-УСЛУГ НА ОСНОВЕ СЕРВИСНО-РЕСУРСНЫХ МОДЕЛЕЙ

В процессе эксплуатации ИУС возникают нештатные ситуации, обусловленные конечной надежностью элементов ИТ-инфраструктуры, нестационарностью процессов подключения пользователей и, соответственно, генерируемого ими потока запросов, а также другими факторами. Для поддержания качества ИТ-сервисов на заданном уровне необходимо автоматически формировать соответствующие управляющие воздействия или задействовать оперативное вмешательство персонала ИТ-подразделения. Каждая нештатная ситуация, которая в теории управления техническими системами рассматривается как возмущающее воздействие, порождает соответствующую прикладную задачу, решением которой является определение вида управляющего воздействия, которое подается на ИТ-инфраструктуру для восстановления качества предоставляемых ИТ-услуг. Решение таких задач осуществляется ИТ УИТИ, используемой в СУИ. Функционал такой ИТ позволяет работать СУИ в автоматическом или автоматизированном режиме. В последнем случае СУИ рассматривается как система поддержки принятия решений, предоставляющая рекомендации администраторам систем ИТ-инфраструктуры по необходимым оперативным управляющим воздействиям.

Для решения задач оперативного управления ИТ-инфраструктурой и создания соответствующей ИТ необходимо разработать математические модели процессов управления ИТ-инфраструктурой и ее компонентами, а также методики управления. Количество решаемых задач управления ИТ-инфраструктурой ввиду ее громоздкости и сложности столь велико, что не позволяет смоделировать все ситуации, которые могут возникнуть в ИТ-инфраструктуре. Поэтому при автоматизации процессов управления выделяют наиболее актуальные задачи управления и на их решении делается акцент при создании ИТ управления ИТ-инфраструктурой. Одной из самых важных задач управления является обеспечение поддержания качества ИТ-услуг на заданном, согласованном с бизнес-подразделениями уровне при условии рационального использования ее ресурсов. Ниже предлагается методика решения этой задачи на основе сервисно-ресурсных моделей.

2.3.1. Композиция ИТ-сервиса

При оперативном управлении уровнем ИТ-услуг должна быть известна структура предоставления ИТ-услуг, определены факторы, влияющие на качество предоставления услуг, известно, как можно измерять и влиять на значения параметров качества, должны быть организованы процессы управления услугами.

Жизненный цикл сервиса состоит из этапов «планирования», «внедрения», «предоставления» и «поддержки» [28]. С композиционной точки зрения сервис является предоставляемой пользователю, управляемой и поддерживаемой ИТ-системой с согласованным уровнем качества [13].

Композиция сервиса определяется комбинацией из трех элементов – ИТ-системы, требований к качеству услуг и поддержки. В свою очередь, в соответствии с ITIL ИТ-система включает в себя активы: Люди – Процессы – Технологии – Информация (рис. 2.8).

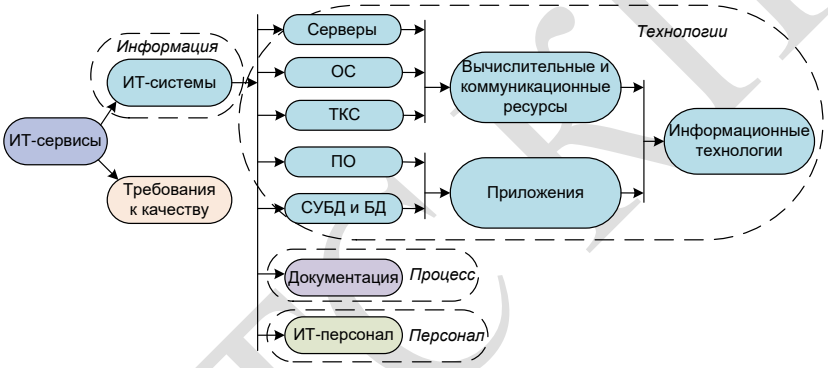


Рис. 2.8. Композиция ИТ-сервисов с точки зрения ITIL

При управлении уровнем ИТ-услуг в ИТ УИТИ целесообразно использовать сервисно-ресурсные модели (СРМ) [28, 208]. В СРМ, представленной в виде иерархического графа (рис. 2.9), узлами являются ЛОУ (L_1–L_4) – ресурсы или сервисы, а ребрами – связи между ними. СРМ создаются на основе системной методологии в рамках определенного процесса управления ИТ-инфраструктурой. Вид модели зависит от целей процесса, для которого создается модель. Одна из основных задач создания СРМ заключается в мониторинге параметров качества ИТ-услуг. При этом качество услуг определяется комплексом измеримых характеристик [13, 28].

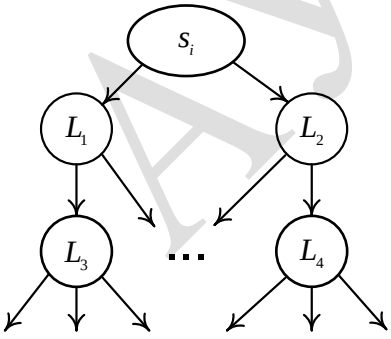


Рис. 2.9. Граф сервисно-ресурсной модели

СРМ широко используется в процессном управлении, однако применение модели в операционной деятельности при оперативном управлении ИТ-инфраструктурой пока не получило широкого распространения [28].

2.3.2. Методика управления уровнем услуг на основе CPM

Предлагаемая методика управления уровнем услуг на основе CPM состоит в реализации такой последовательности этапов:

1. Формирование каталога ИТ-сервисов.
2. Определение параметров качества услуг.
3. Проектирование CPM.
4. Определение методов получения значений показателей качества услуг.
5. Мониторинг и анализ значений параметров качества услуг.
6. Управление ИТ-инфраструктурой по поддержанию согласованного уровня услуг.

Подробнее суть этапов заключается в следующем [112].

Этап 1. Формирование каталога ИТ-сервисов. В ITSM инициализируется процесс «Планирование и реализация новых или измененных услуг», целью которого является гарантирование предоставления новых/измененных услуг и их менеджмента в соответствии с согласованными затратами и качеством услуг.

Бизнес-процесс разделяется на бизнес-операции или функции, для которых определяются бизнес-услуги и технологические услуги. Запускается согласовательный процесс между бизнес-менеджерами и ИТ-дирекцией (см. рис. 3.3). Бизнес выдвигает предложения относительно новых $s_i^{\text{н}}$ или изменяемых $s_i^{\text{м}}$ услуг, руководствуясь ожидаемыми результатами коммерческого воздействия.

ИТ-дирекция рассматривает различные варианты представления новой услуги $s_i^{\text{н}}$: комбинация или модификация имеющихся услуг $\mathcal{G} = \{s_i\}$, $i = \overline{1,K}$, с учетом стоимости модификации $C_i^{\text{м}}$, поддержки $C_i^{\text{п}}$ и обслуживания $C_i^{\text{о}}$; самостоятельно разрабатываемое ПО с учетом стоимости разработки $C_i^{\text{р}}$ и сроков; закупка ПО с анализом стоимости лицензии $C_i^{\text{л}}$, сроков запуска, востребованности; поставляемые с использованием облачных вычислений с учетом стоимости $C_i^{\text{Об}}$ и надежности. Понимание структуры образования стоимости услуг необходимо для оптимизации портфеля услуг. Пути оптимизации услуг выбираются с учетом критерия $\min\limits_{C}\{(C_i^{\text{м}} + C_i^{\text{п}} + C_i^{o}); (C_i^{\text{р}} + C_i^{\text{п}} + C_i^{o}); (C_i^{\text{л}} + C_i^{\text{п}} + C_i^{o}); (C_i^{\text{Об}})\}$.

Новая услуга заносится в каталог, содержащий информацию о наименовании услуги и значении целевых показателей уровня услуги.

Этап 2. Определение параметров качества ИТ-услуг. На этом этапе определяются параметры качества всех услуг s_i, $i = \overline{1,K}$, входящих в каталог $\mathcal{G}$. Этап реализуется в рамках процесса «Управление уровнем услуг (SLM)», обеспечивающего согласование уровня качества и рисков для всех услуг s_i, $i = \overline{1,K}$. Каждый сервис S_b, $b = \overline{1,D}$, где D – количество сервисов, состоит из совокупности услуг $s_{b,i} \in \mathcal{G}$, $i = \overline{1,K}$, $b = \overline{1,D}$. Согласованные показатели качества $q_{b,ki}$, $b = \overline{1,D}$, $k = \overline{1,M_i}$, $i = \overline{1,K}$, документируются в SLA. Здесь $q_{b,ki}$ – значение k-го показателя качества i-й услуги, когда она предоставляется вместе с b-м сервисом. SLA находится под управлением процесса «Управление изменениями». Примерный перечень показателей приведен на рис. 2.10.

Из всех показателей качества услуг СУИ оперирует только с измеримыми объективными показателями $q_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$ (рис. 2.10). Пример фрагмента SLA с показателями, находящимися в сфере оперативного управления ИТ-инфраструктурой, приведен в табл. 2.2.

Рис. 2.10. Примерный перечень показателей ИТ-сервисов

Функции мониторинга и анализа ИТ УИТИ при управлении уровнем услуг сводятся к определению фактических значений $q^{*}_{b,ki}$ показателей уровня услуг и сравнения их с целевыми $q_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, из табл. 2.2.

Таблица 2.2. Пример фрагмента SLA с показателями уровня услуг

ИТ-сервис	ИТ-услуги	Целевые показатели уровня услуг и их граничные значения					
		Показатели производительности	Значение	Показатели надежности и доступности	Значение	…	…
S_1	$S_{1,1}$	Время открытия формы, не более (с)	1	Доступность, не хуже	99,99	…	…
	$S_{1,т1}$	Загрузка данных, не более (с)	0,5	Доступность, не хуже	99,99	…	…
	$S_{1,т2}$	Печать документа, не более (с)	20	Доступность, не хуже	99,99	…	…
S_2	$S_{2,2}$	Скорость доступа (Мбит/с)	100	Доступность, не хуже	99,9	…	…
	…	…	…	…	…	…	…
	$S_{2,i}$	Время отклика, не более (с)	1	Доступность, не хуже	99,9	…	…
…	…	…	…	…	…	…	…

Этап 3. Проектирование CPM. Несмотря на то, что реализации проектов ITSM часто осуществляются с использованием CPM, практически отсутствуют методики построения и применения CPM. В [28] сформулированы некоторые методологические аспекты построения CPM, которые, однако, не затрагивают процессы оперативного управления.

При согласовании проекта сервиса S_b выделяются его компоненты $s_{b,i} \in \mathcal{G}$, связи между компонентами и за счет создания CPM проводится отображение услуг на ПО и ресурсы ИТ-инфраструктуры.

При оперативном (автоматическом или автоматизированном) управлении услугами целесообразно создавать СРМ только для критически важных услуг из $\mathcal{G}$, относительно которых осуществляется управление по поддержанию согласованного уровня услуг.

При создании СРМ важно придерживаться следующих правил:

- модели проектируются по принципу «сверху вниз» – от сервисов к каналам связи;
- в СРМ выделяются уровни, соответствующие выделенным уровням ИТ-инфраструктуры;
- связи в модели должны отображать влияние состояний расположенных ниже по иерархии ЛОУ на значения параметров качества расположенных выше ЛОУ;
- связи не должны содержать петель;
- определяются параметры, измеряемые в каждом элементе СРМ.

Пример схемы СРМ для сервиса S_1 приведен на рис. 2.11.

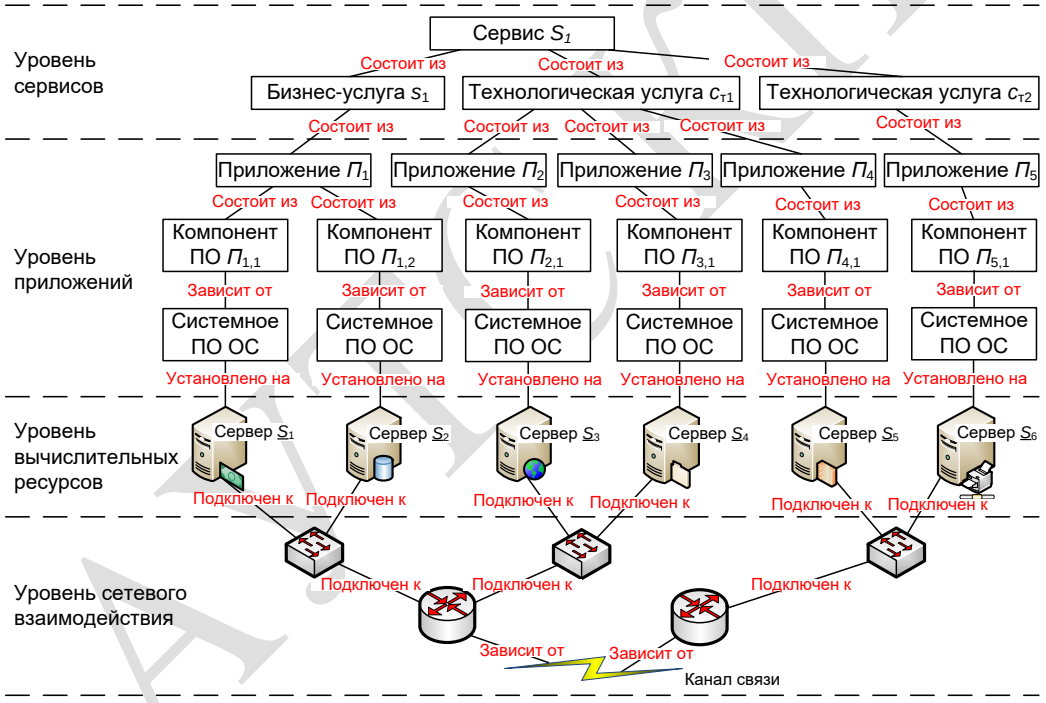


Рис. 2.11. Пример схемы СРМ

На основе СРМ и требований к параметрам сервиса определяются потребности в ресурсах ИТ-инфраструктуры. Кроме того, СРМ используются для оперативного контроля функционирования сервисов. Для сервисов, функционирование которых оценивается по принципу «работает – не работает» [28], СРМ не содержит ответвлений.

Этап 4. Определение методов получения значений показателей качества сервисов. Определение степени снижения уровня услуг $w(\Delta q_{b,ki})$ проводится на

основе фактических значений $q^*_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$. Измерение значений $q^*_{b,ki}$ для всех пользователей технически реализуемо, но трудоемко, поэтому широко используется вычисление $q^*_{b,ki}$ на основании косвенных переменных, характеризующих функционирование ОМУ, которые участвуют в предоставлении услуги. Для этого решается задача сведения метрик, после чего сведенные значения $q^*_{b,ki}$ сравниваются с целевыми значениями $q_{b,ki}$ показателей уровня услуг.

Рассмотрим в качестве примера приведенную на рис. 2.11 СРМ, сконструированную для сервиса S_1 (см. табл. 2.2). Сервис состоит из услуг $s_{1,1}$, $s_{т,1}$ и $s_{т,2}$. Покажем, как СРМ используется для расчета показателя производительности на примере параметра «Время открытия формы» услуги $s_{1,1}$. Услуга $s_{1,1}$ зависит от: приложения $П_1$ (компоненты ПО $П_{1,1}$ и $П_{1,2}$), сервера приложений $\underline{S}_1$, СУБД и БД на сервере $\underline{S}_2$, программных компонентов серверов и ТКС. На значение показателя $q_{1,1,1}$ «Время открытия формы», оказывает влияние производительность сервера $\underline{S}_1$ с установленной на нем ПО и ТКС. Из всех параметров, характеризующих функционирование этих элементов ИТ-инфраструктуры, выделяются параметры, влияющие на показатель $q_{1,1,1}$: время Δt_{S_1} отклика сервера $\underline{S}_1$ на запрос; время обслуживания запроса $\Delta t_{П_1}$ компонентом ПО $П_{1,1}$; задержка $\Delta t_{ТКС}$ прохождения данных по ТКС. Эти данные измеряются подсистемой мониторинга, которая также постоянно отслеживает количество a_{S_1} открытых пользовательских сессий для контроля доступности услуги $s_{1,1}$ (рис. 2.12).

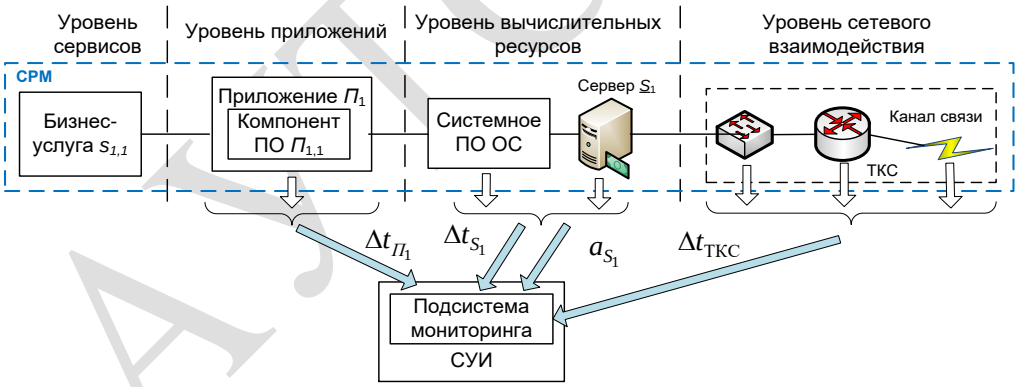


Рис. 2.12. Измерение значений параметров функционирования ИТ-инфраструктуры, оказывающих влияние на значение показателей уровня услуги $s_{1,1}$

Этап 5. Мониторинг и анализ значений параметров качества услуг осуществляется для установления соответствия текущих значений показателей уровня услуг целевым значениям, т. е. текущего контроля уровня услуг. Выделяются два вида контроля услуг – оперативный и аналитический [28, 112]. Первый необходим для оперативного управления, направленного на поддержание услуг на согласованном уровне. Аналитический контроль используется при процессном управлении ITSM для накопления статистики при коррекции целевых значений уровней услуг.

АУТСКИЙ

При оперативном контроле измеряются значения q^*_{ki}, $k=\overline{1,M_i}$, $i=\overline{1,K}$, которые сводятся к обобщенному показателю. Потом значение этого показателя сравнивается с целевым значением показателя качества услуг. Например, для услуги $s_{1,1}$ обобщенным показателем будет суммарная задержка передачи данных $\Delta t_{1,1}$:

$$\Delta t_{1,1} = \Delta t_{П_1} + \Delta t_{S_1} + \Delta t_{\text{ТКС}}. \quad (2.5)$$

Значение (2.5), умноженное на поправочный коэффициент, соответствует фактическому значению $q^*_{1,1,1}$ уровня услуги $c_{1,1}$:

$$q^*_{1,1,1} = k_{1,1,1} \cdot \Delta t_{1,1}. \quad (2.6)$$

Фактическое значение $q^*_{1,1,1}$ уровня услуги $s_{1,1}$ сравнивается с целевым значением $q_{1,1,1}$ и в случае несоответствия СУИ генерирует управляющее воздействие.

Этап 6. Управление ИТ-инфраструктурой по поддержанию уровня услуг. Для поддержания уровня услуг фактическое значение $q^*_{b,ki}$ сравнивается с целевым значением $q_{b,ki}$ для всех или определенных показателей качества. Если значение $q^*_{b,ki}$ превышает предельные значения $q_{b,ki}$, то администратор проводит действия, направленные на то, чтобы показатели качества услуг вернулись к нормативным значениям.

На рис. 2.13 в качестве примера рассмотрена реализация управления с использованием CPM для подержания услуги $s_{1,1}$ на согласованном уровне с помощью определения значений составляющих выражения (2.5).

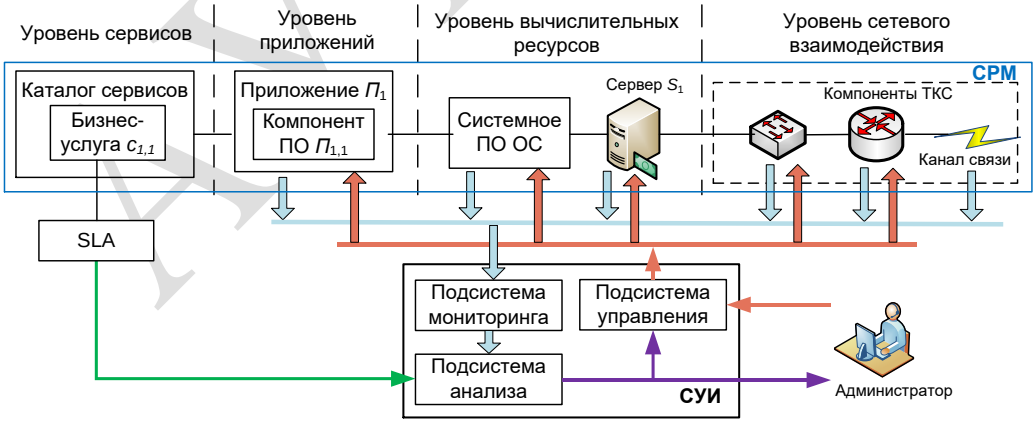


Рис. 2.13. Реализация оперативного управления для поддержания уровня услуг

АУТСКД

2.3.3. Процессно-оперативный подход к управлению уровнем ИТ-услуг

Управление уровнем услуг осуществляется как средствами процессного управления ITSM, так и оперативного, реализуемого СУИ (см. рис. 1.8). Интерес представляет комплексный подход, предполагающий тесную взаимосвязь процессного и оперативного управления уровнем услуг. В этом случае связующим звеном для интеграции управления является СУИ (рис. 2.14), которая, кроме автоматизации управления ИТ-инфраструктурой и оперативного (операционного) управления по поддержанию согласованного уровня услуг, выполняет ряд функций процессного управления ITSM наряду с обеспечением взаимодействия администраторов SLM и ИТ-инфраструктуры для управления уровнем услуг и решения проблем, возникающих в ИТ-инфраструктуре.

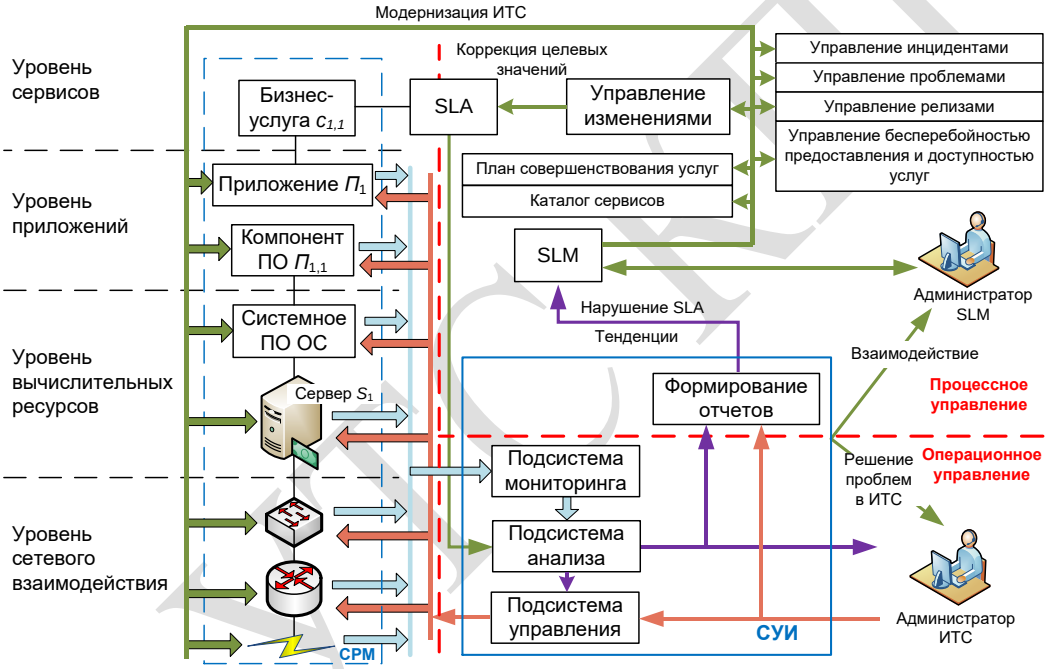


Рис. 2.14. Схема реализации комплексного подхода к управлению уровнем услуг

Процессное управление по результатам аналитического контроля и с учетом финансовых затрат в рамках процесса SLM проводит коррекцию целевых значений $q_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, показателей качества услуг и стимулирует решение проблем в ИТ-инфраструктуре, влияющих на снижение качества (см. рис. 3.2). Целью SLM является повышение уровня услуг. Для достижения цели выполняется цикл определения, согласования, утверждения, контроля и оценки уровня предоставляемых ИТ-услуг, а также активации действий по повышению качества услуг. Для этого определяются точные и измеримые показатели $q_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, качества всех услуг. После чего фактический уровень услуг $q^{*}_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, определяемый СУИ, контролируется, а

нарушения SLA фиксируются в отчетах, наряду со сведениями о тенденциях изменения уровня услуг и причинах несоответствия. Все действия по улучшению услуг вносятся в план совершенствования услуг [262]. SLM должен содействовать реализации превентивного подхода к управлению уровнем услуг. Качество услуг зависит от того, как организована деятельность по их управлению. При процессном управлении используется простая модель управления качеством в виде цикла Деминга [13].

Поставщик и заказчик услуг регулярно корректируют уровень услуг $q_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, запуская процесс SLM. Среди параметров, характеризующих качество ИТ-услуг, имеется показатель удовлетворенности пользователей (QoE), который рассматривается как субъективная составляющая качества услуги, а целевые показатели уровня услуг – как объективные параметры. Объективные параметры, значения которых зависят от состояния и функционирования ИТ-инфраструктуры, являются предметом контроля со стороны СУИ при операционном управлении ИТ-инфраструктурой. Для этого СУИ осуществляет непрерывный контроль выполнения целевых показателей уровня услуги. Кроме того при оперативном управлении проводятся анализ рисков возникновения сбоев, планирование роста/сокращения численности пользователей или объема подлежащих хранению данных, компенсация пиков загруженности ИТ-инфраструктуры и всех изменений условий предоставления услуг. В этих случаях оперативное управление реализуется перераспределением ресурсов с учетом приоритетов бизнеса, критичности услуги и прогноза будущих требований к доступности с обеспечением приемлемого уровня услуг.

Комплексное управление уровнем услуг позволяет снизить число нарушений SLA и степень влияния ухудшения качества на бизнес, обеспечивая поддержание согласованного уровня услуг, достаточного для удовлетворения потребностей бизнеса.

ВЫВОДЫ К ГЛАВЕ 2

1. Разработана концепция управления ИТ-инфраструктурой корпоративного уровня, отличающаяся отказом от объединения разрозненных систем управления и переходом к рассмотрению СУИ как целостной системы, подчиненной целям бизнеса. Реализация концепции требует разработки соответствующей ИТ УИТИ и инструментария управления. Применение такой ИТ УИТИ при разработке СУИ позволит поддерживать согласованный уровень услуг с одновременным увеличением эффективности использования ресурсов ИТ-инфраструктуры, что окажет положительное влияние на выполнение бизнес-процессов и повышение конкурентоспособности компаний.

2. Детализированы основные положения концепции управления ИТ-инфраструктурой, что позволило ввести определения и формализмы, необходимые для формулировки критериев и ограничений задач управления в последующих главах.

3. Получила дальнейшее развитие парадигма ITMS2. Предложено выделить пять универсальных процессов, общих для всех категорий команд управления ИТ-инфраструктурой: мониторинга, анализа, управления, оптимизации и планирования. В отличие от ITMS2 проведено объединение не функций, а процессов,

что позволяет в разработанной информационной технологии использовать наборы универсальных моделей, алгоритмов, методов, модулей и механизмов.

4. Предложена методика управления уровнем услуг с использованием сервисно-ресурсных моделей, создаваемых на основе системной методологии в рамках процесса управления ИТ-инфраструктурой. Это позволяет не только легко выявлять факторы, приведшие к снижению уровня услуг, но и определять рациональные компенсирующие воздействия для восстановления согласованного уровня услуг.

5. Предложен комплексный подход к управлению уровнем услуг на основе сервисно-ресурсных моделей, отличающийся интеграцией процессного и оперативного управления. Подход позволяет обеспечивать не только оперативное поддержание согласованного уровня услуг, но и стратегическое обоснование согласованного уровня услуг в зависимости от имеющихся ресурсов ИТ-инфраструктуры.

6. Разработана расширенная базовая модель управления ИТ-инфраструктурой, которую отличает декомпозиция команд управления ИТ-инфраструктурой в потоки команд управления приложениями и ресурсами.

АУТСКИЙ

ГЛАВА 3

УПРАВЛЕНИЕ УРОВНЕМ УСЛУГ В КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРЕ

В главе предложен декомпозиционно-компенсационный способ организации управления уровнем услуг, основанный на декомпозиции задачи управления и компенсации отрицательного влияния отдельных факторов за счет выделения дополнительных ресурсов критичным приложениям. Разработана модель СУИ в виде двухуровневой системы управления с координатором. Определены принципы управления и координации.

3.1. ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ УРОВНЕМ ИТ-УСЛУГ

3.1.1. Декомпозиционно-компенсационный способ организации управления уровнем услуг

Основная цель бизнеса – получение максимальной прибыли. Максимальная прибыль за счет ИТ достигается тогда, когда бизнесу предоставляется множество $\mathcal{G} = \{s_i\}$, $i = \overline{1,K}$, необходимых ИТ-услуг с максимальным качеством $\mathcal{Q}$ и минимальными затратами $\mathcal{C}$.

Управление уровнем услуг в ИТ-инфраструктуре корпоративного уровня предлагается осуществлять интегрированным взаимодействием трех процессов [105, 325, 326, 327]: согласования уровня услуг, планирования ресурсов и управления уровнем услуг (рис. 3.1).

Процесс согласования уровня услуг запускается по инициативе бизнес-менеджеров и заканчивается формированием или обновлением элементов множества $\mathcal{G}$ и матрицы $Q = \|q_{ki}\|$, элемент q_{ki}, $k = \overline{1,M_i}$, $i = \overline{1,K}$, которой соответствует значению k-го показателя качества i-й услуги. Бизнес выделяет услугам $\mathcal{G}$ количество r ресурсов. Полученное в результате ресурсное обеспечение в виде системы

$$< Q, r > \tag{3.1}$$

является основой для решения задач на расположенном ниже уровне.

Процесс планирования заключается в выделении и закреплении за каждой услугой s_i, $i = \overline{1,K}$, части ресурсов ИТ-инфраструктуры из ресурсов $R_1,...,R_m$, выделенных для поддержания услуг. При этом $r_1,...,r_m$ – количество ресурсов $R_1,...,R_m$, а $c_1,...,c_m$ – стоимость единицы ресурса $R_1,...,R_m$ соответственно.

Тогда количество всех ресурсов вычисляется как

$$r = \sum_{j=1}^{m} r_j \,, \tag{3.2}$$

АУТСКИ

а стоимость c ресурсов определяется следующим образом:

$$c = \sum_{j=1}^{m} r_j \cdot c_j \ . \tag{3.3}$$

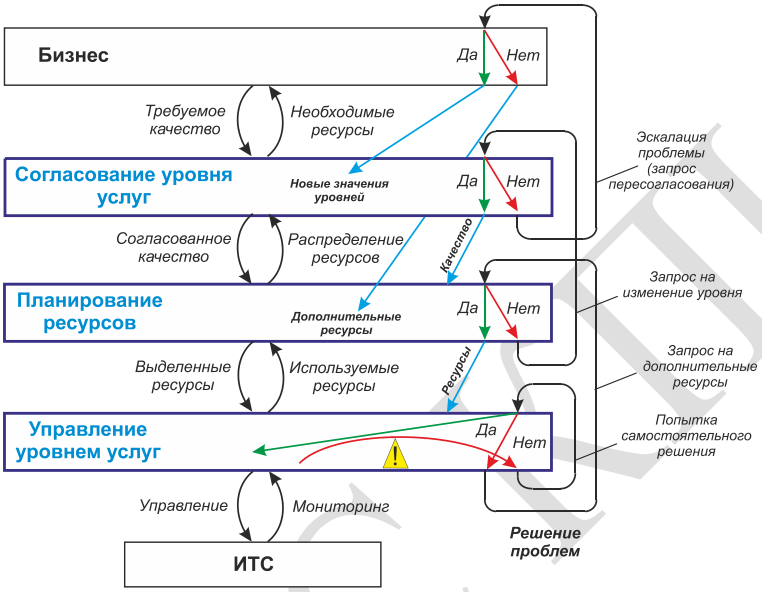


Рис. 3.1. Взаимодействие процессов при управлении уровнем услуг

Использование услугами ресурсов задается матрицей $\mathrm{P} = \left\| \rho_{ij} \right\|$, где ρ_{ij} равно количеству выделенного услуге s_i ресурса R_j, $j = \overline{1,m}$ или 0, если ресурс не выделяется.

Процесс управления уровнем услуг осуществляет управление ИТ-инфраструктурой так, чтобы фактические значения q_{ki}^*, $k = \overline{1,M_i}$, $i = \overline{1,K}$, показателей качества услуг соответствовали согласованным значениям из матрицы Q, т. е. чтобы выполнялось равенство

$$q_{ki} - q_{ki}^* = 0, \ \ k = \overline{1,M_i}, i = \overline{1,K} \ . \tag{3.4}$$

Предлагается способ организации управления уровнем услуг, построенный на межуровневом взаимодействии [325].

При невыполнении условия (3.4) выявляются элементы матрицы фактических значений показателей качества $Q^* = \left\| q_{ki}^* \right\|$, для которых $q_{ki}^* < q_{ki}$, $k = \overline{1,M_i}$, $i = \overline{1,K}$. ИТ УИТИ пытается решить проблему на нижнем уровне (см. рис. 3.1), изменяя значения параметров функционирования ИТ-инфраструктуры либо перераспределяя ресурсы между приложениями так, чтобы увеличить значение q_{ki}^*. Если в результате восстановительных мероприятий удалось обеспечить выполнение равенства (3.4), то функционирование ИТ-инфраструктуры продолжается

с новыми настройками. Если полномочий нижнего уровня недостаточно для достижения (3.4), то осуществляется эскалация проблемы на уровень планирования ресурсов.

Процесс планирования ресурсов пытается решить проблему выделением дополнительных ресурсов $R_1,...,R_m$ услуге s_i, для которой выполняется условие $q_{ki}^* < q_{ki}$. Если дополнительные ресурсы выделяются, то формируется матрица $\mathrm{P}' = \left\| \rho'_{ij} \right\|$ с новыми значениями элементов, причем $\rho'_{ij} > \rho_{ij}$, $j = \overline{1,m}$ или 0, если j-й ресурс не требуется. Если дополнительные ресурсы отсутствуют, то уровень планирования ресурсов пытается провести перераспределение ресурсов между услугами, отдавая ресурсы более важным услугам за счет менее важных. Если проблема решается, то значения матрицы $\mathrm{P}' = \left\| \rho'_{ij} \right\|$ с новым планом закрепления ресурсов поступают на нижний уровень. При невозможности разрешения проблемы на уровне планирования ресурсов проводится эскалация проблемы на расположенный выше уровень [327].

Процесс согласования уровня услуг по инициативе процесса планирования осуществляет пересмотр сначала значения q_{ki}, для которого выполняется условие $q_{ki}^* < q_{ki}$, а затем, возможно, и значений всех элементов q_{ki}, $k = \overline{1,M_i}$, $i = \overline{1,K}$, матрицы качества услуг Q в сторону уменьшения. Если удается сформировать матрицу $Q' = \left\| q'_{ki} \right\|$ с новыми значениями показателей качества услуг, то она передается на уровень ниже, где проводится высвобождение ресурсов и выделение их услугам, для которых выполняется условие $q_{ki}^* < q_{ki}$, $k = \overline{1,M_i}$, $i = \overline{1,K}$. Если процесс согласования уровня услуг не имеет полномочий для формирования матрицы $Q' = \left\| q'_{ki} \right\|$, то проводится эскалация проблемы на уровень бизнеса, который вынужден либо сгенерировать матрицу $Q' = \left\| q'_{ki} \right\|$ с новыми значениями, либо увеличить общий объем ресурсов, что приводит к увеличению значений $r_1,...,r_m$.

Рассмотрим подробнее процессы, реализуемые при согласовании уровня услуг, планирования ресурсов и управления уровнем услуг с дальнейшей формализацией решаемой в работе проблемы.

3.1.2. Согласование уровня услуг в корпоративных ИТ-инфраструктурах

Требования максимизации ($\max \mathfrak{Q}$) качества услуг и минимизации связанных с этим затрат ($\min \mathfrak{C}$) находятся в естественном противоречии, что вынуждает каждое предприятие установить экономически обоснованный уровень услуг с учетом как своих возможностей, так и достигнутого отраслью уровня и ожиданий клиентов.

Если $\hat{D}$ – доходы бизнеса от ИТ, а $\hat{C}$ – затраты на ИТ-инфраструктуру или ИУС, то бизнес совместно с процессом согласования уровня услуг пытается добиться

$$\max(\hat{D} - \hat{C}). \qquad (3.5)$$

Стремясь к повышению качества, бизнес- и ИТ-подразделения должны регулярно в рамках процессного управления ITSM выполнять аналитическую оценку

зависимостей качества сервиса от стоимости ресурсов $\hat{Q} = f_1(c)$, потерь бизнеса $\hat{L}$ от некачественного сервиса с учетом рисков M_0 и неопределенностей N_0 (см. рис. 3.2). После этого с учетом уровня качества, достигнутого отраслью, определяются допустимые значения уровня услуг $Q = \|q_{ki}\|$, которые еще не приводят к потерям при выполнении бизнес-операций. Таким образом, система

$$< \hat{D}, \hat{C}, \hat{L}, \hat{Q}, M_o, N_0 > \qquad (3.6)$$

определяет задачу по формированию пары (3.1).

Если полученные с учетом рисков значения уровня услуг превышают достигнутый отраслью уровень, т. е. повод для оптимизма и развития бизнеса (см. рис. 3.2). Если не превышают, то это свидетельствует о проблемах и является основой для сворачивания бизнеса в соответствующих направлениях.

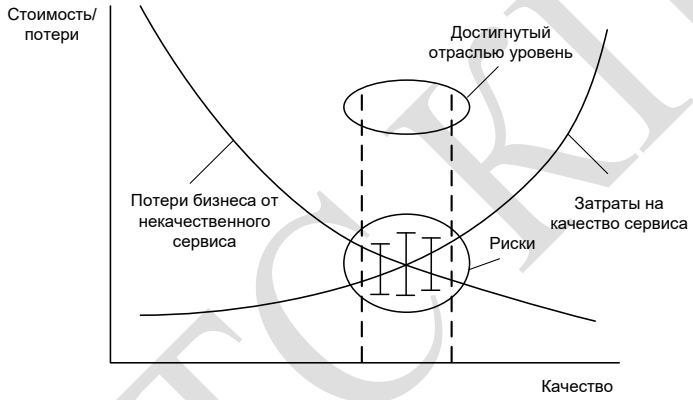


Рис. 3.2. Нахождение оптимального соотношения уровня качества сервиса и стоимости достижения этого качества

Варианты реализации услуги со стоимостными, техническими, организационными и временными показателями предоставляются бизнесу. Бизнес определяет оптимальное отношение доходность/качество ИТ-услуг. Для этого оцениваются потери в зависимости от падения уровня услуг и стоимость предоставления услуг высокого качества, после чего находится точка минимально допустимого качества (рис. 3.3). Затем бизнес одобряет один из вариантов или корректирует требования к новой ИТ-услуге, исходя из значимости сервиса, бенчмаркинга, опыта и пр.

План внедрения новых услуг ИТ-дирекция готовит с учетом финансового и ресурсного обеспечения их предоставления и управления. При этом используются методы эффективного использования ресурсов ИТ-инфраструктуры, предложенные в главах 7 и 8.

Согласование уровня услуг проводится в условиях избытка или дефицита ресурсов. При дефиците ресурсов запускается итеративная процедура согласования уровня услуг, когда на основании (3.2) процесс планирования определяет значения q_{ki}, $k = \overline{1, M_i}$, $i = \overline{1, K}$, затем полученные показатели в виде матрицы Q предъявляются бизнесу, и если значения элементов матрицы Q не удовлетворяют бизнес, то проводится перераспределение ресурсов между услугами и запускается

новый цикл определения значений q_{ki}, $k = \overline{1, M_i}$, когда для заданного плана закрепления ресурсов $R_1,...,R_m$ за услугами s_i, $i = \overline{1, K}$, определяется ожидаемое качество Q.

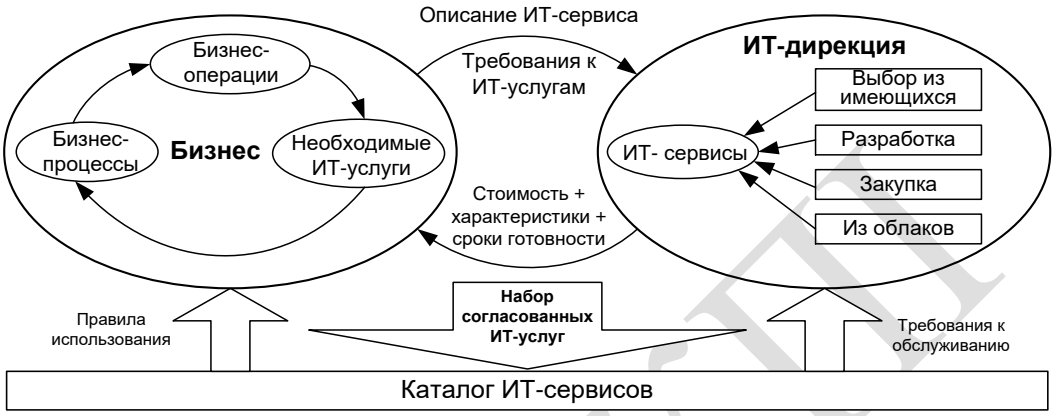


Рис. 3.3. Процесс формирования каталога ИТ-сервисов

При решении задач согласования уровня услуг могут быть использованы методы математического программирования, исследования операций, теории принятия решений, методы теории игр, методы решения многокритериальных задач, принятия решений в условиях неопределенности и риска, методы искусственного интеллекта, балансовые модели и др.

3.1.3. Планирование ресурсов

На уровне планирования ресурсов решаются задачи в интересах процесса согласования уровня услуг, проводится определение необходимого количества ресурсов, распределение и закрепление ресурсов за услугами s_i, $i = \overline{1, K}$.

Услуги s_i, $i = \overline{1, K}$, поддерживаются приложениями из множества $A = \{A_l\}$, $l = \overline{1, I}$, где I – количество приложений, причем каждая услуга s_i, $i = \overline{1, K}$, поддерживается одним или несколькими приложениями, в то время как каждое приложение A_l, $l = \overline{1, I}$, поддерживает одну или несколько услуг.

Введем понятие поддержки услуги. Под поддержкой услуги s_i будем понимать взаимодействие нескольких приложений из множества A, обеспеченных ресурсами, направленное на достижение единого результата – работоспособность услуги s_i. Пусть ресурсы R_j, $j = \overline{1, m}$, используются для поддержки услуг s_i, $i = \overline{1, K}$. Введем булевую переменную x_i, $i = \overline{1, K}$, определяющую степень поддержки i-й услуги и принимающую значения:

$$x_i = \begin{cases} 1, \text{если } i\text{-я услуга поддерживается в полном объеме;} \\ 0, \text{если } i\text{-я услуга не поддерживается.} \end{cases} \quad (3.7)$$

Пусть пользователи услуги s_i формируют в среднем a_{li} запросов к приложению A_l за единицу времени. Тогда количество клиентских запросов приложения A_l определяется следующим образом:

$$a_l = \sum_{i=1}^{K} a_{li} \cdot x_i \,, \tag{3.8}$$

а количество запросов к приложениям A_l, $l = \overline{1,I}$, представляется вектором $\hat{a} = \{a_l, l = \overline{1,I}\}$.

Потребности приложения A_l в ресурсах типа j определяются выражением

$$r_{jl} = b_{jl}\, a_l + d_{jl}, \tag{3.9}$$

где b_{jl} – среднее количество ресурсов типа j, используемых приложением A_l для обработки одного клиентского запроса; d_{jl} – количество ресурсов типа j, используемых приложением A_l независимо от количества клиентских запросов.

Тогда общее количество ресурсов r, необходимое для поддержания всех услуг из $\mathcal{G}$, определяется с помощью выражения

$$r = \sum_{j=1}^{m}\sum_{l=1}^{I} r_{jl} = \sum_{j=1}^{m}\sum_{l=1}^{I} (b_{jl} a_l + d_{jl}) \,. \tag{3.10}$$

Задачи планирования существенно зависят от ограничений на бюджет ИТ-подразделения. При отсутствии финансовых ограничений, когда критерий $\min \mathcal{C}$ не учитывается, планируется и проектируется ИТ-инфраструктура, ресурсы r которой будут достаточными для поддержания приложений $\{A_l\}$ с требуемыми значениями показателей качества услуг q_{ki}, $k = \overline{1,M_i}$, $i = \overline{1,K}$, при максимальных допустимых значениях вектора $\hat{a}$. Кроме того, наиболее важные ресурсы дублируются или резервируются ($r_r > 0$), а также учитывается возможное запредельное увеличение количества запросов a_{li}.

При ограниченном бюджете ИТ-подразделения на создание и развитие ИТ-инфраструктуры дефицит ресурсов может закладываться уже на стадии проектирования или возникать в процессе эксплуатации, и сложившаяся ИТ-инфраструктура в принципе не может обеспечить выполнение условия (3.4). Несмотря на это, ИУС, спроектированная с дефицитом ресурсов, способна эффективно предоставлять услуги, если априорно обеспечивается поддержка наиболее важных приложений. Для этого ИТ УИТИ перераспределяет ресурсы в процессе работы в соответствии с определенным регламентом использования ресурсов. Решению подобных задач посвящена глава 7.

Таким образом, задачи планирования и управления ресурсами необходимо решать в условиях как избытка, так и дефицита ресурсов. В каждом конкретном случае необходимо учесть некоторые специфические особенности распределения ресурсов.

После расчета потребности ресурса на основе выражения (3.10) и ее сравнения с имеющимися ресурсами получим задачу управления уровнем сервисов при дефиците ресурсов, если нештатные ситуации в ИТ-инфраструктуре, увеличение

интенсивности клиентских запросов $\hat{a}$ и другие факторы приводят к невыполнению равенства (3.4).

В этом случае для принятия решений при распределении ресурсов необходимо учитывать дополнительную информацию.

Введем понятие важности w_i услуги s_i, $i = \overline{1,K}$, которое будем использовать при решении задачи согласования уровня услуг в условиях дефицита ресурсов.

Задача согласования уровня услуг *при дефиците ресурсов* сводится к определению значений матрицы Q при заданном количестве ресурсов:

$$Q = \mathfrak{F}_1(\mathcal{G}, r, W_p, Z_s), \tag{3.11}$$

где $W_p = \{w_i \mid i = \overline{1,K}\}$; $Z_s = \{z_i \mid i = \overline{1,K}\}$; z_i – планируемая степень поддержки i-й услуги. Здесь нормативное значение количества клиентских запросов учитывается в величине r.

Если же после расчета потребности в ресурсах и ее сравнения с имеющимися ресурсами последних оказывается больше, то получим задачу управления уровнем услуг *при избытке ресурсов*. В частности, может быть выделен резервный объем r_r ресурсов, и количество ресурсов $\hat{r}$, выделяемое для поддержания всех услуг из $\mathcal{G}$ и определяемое по выражению

$$\hat{r} = r + r_r, \tag{3.12}$$

возрастает. Тогда объем ресурсов r определяется, исходя из значений элементов матрицы Q, а величина r_r – исходя из вероятности возникновения нештатных ситуаций и предельного значения величины a_{li}.

В этом случае выделяются две задачи согласования уровня услуг. Задача первого рода аналогична задаче (3.11) и состоит в определении значений показателей качества при известном объеме выделенных ресурсов:

$$Q = \mathfrak{F}_2(\mathcal{G}, \hat{r}, r_r, \mathcal{C}). \tag{3.13}$$

Задача второго рода состоит в определении необходимого объема ресурсов для обеспечения заданных значений показателей качества:

$$r = \mathfrak{F}_3(\mathcal{G}, Q, \mathcal{C}). \tag{3.14}$$

При решении задач (3.11), (3.13) и (3.14) могут быть использованы методы теории массового обслуживания [51, 52], теория надежности [79], теории фракталов, аналитическое и имитационное моделирование, в частности с применением методов теории массового обслуживания и искусственного интеллекта.

3.1.4. Процесс управления уровнем услуг

После согласования уровня услуг и планирования ресурсов процесс управления осуществляется так, чтобы выполнялись критерии

$$\min(q_{ki} - q_{ki}^*), \quad k = \overline{1,M_i}, i = \overline{1,K}, \text{ при } q_{ki}^* < q_{ki} \tag{3.15}$$

или

АУЭС РФ

$$\min \hat{C},\ \text{при}\ q_{ki}^{*} > q_{ki},\quad k=\overline{1,M_i},\ i=\overline{1,K}. \tag{3.16}$$

В последнем случае для экономии затрат проводится сокращение выделенных приложениям $\{A_l\}$ ресурсов, а незадействованные ресурсы отключаются. Такие задачи решаются в главе 4.

Критерии (3.15) и (3.16) применяются только тогда, кода для всех ресурсов $k=\overline{1,M_i}$ и услуг $i=\overline{1,K}$ при сравнении значений q_{ki}^{*} и q_{ki} выполняется условие «только не больше» или «только не меньше». В противном случае ресурсы между приложениями из $\{A_l\}$ могут быть перераспределены так, чтобы приложениям, для которых выполняется условие $q_{ki}^{*} < q_{ki}$, были выделены ресурсы за счет приложений, для которых выполняется условие $q_{ki}^{*} > q_{ki}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$.

Если средствами нижнего уровня не удается обеспечить равенство $q_{ki}^{*} = q_{ki}$ при $q_{ki}^{*} < q_{ki}$, то запускается итеративная процедура, в которой задействуются расположенные выше уровни (см. рис. 3.1). В этом случае на расположенных выше уровнях приложению $A_l^{*} \in A$, для которого выполняется условие $q_{ki}^{*} < q_{ki}$, дополнительно выделяется квант Δr ресурсов. После чего проверяется выполнение условия (3.4). Если по-прежнему выполняется условие $q_{ki}^{*} < q_{ki}$, то выделяется еще один квант ресурсов Δr. Процедура повторяется до тех пор, пока не начнет выполняться условие (3.4). При отсутствии в ИТ-инфраструктуре запаса ресурсов при управлении уровнем услуг возможны две ситуации:

1) пересматриваются значения матицы *Q*;

2) выделяются кванты ресурса Δr за счет приложений из множества $\{A_l\}$ с учетом важности W_p услуг.

Зависимость значений показателей качества q_{ki}, $k=\overline{1,M_i}$, $i=\overline{1,K}$, от объема ресурсов *r* без потери общности можно представить в виде

$$q = f_{qr}(r), \tag{3.17}$$

где *q* – показатель качества услуги. Для увеличения значения *q* соответствующему приложению из множества $\{A_l\}$ необходимо выделить дополнительные ресурсы. Тогда

$$q' = f_{qr}(r+\Delta r). \tag{3.18}$$

Если $\Delta r > 0$, то $q' \geq q$, что позволяет сделать предположение о монотонном характере функции f_{qr}.

Аналогичным образом можно предположить, что функция

$$q = f_{qa}(\hat{a}) \tag{3.19}$$

также будет монотонной.

Тогда, если функции (3.17) и (3.19) монотонные, то функция

Аутсорс

$$q = f_q(r, \hat{a}) \tag{3.20}$$

также будет монотонной [68].

Пусть управление $u^+ \in U$, где U – множество управляющих воздействий, заключается в выделении дополнительных ресурсов приложению $A_l^* \in A$, для которого фактическое качество $q_{\text{ф}}$ хуже целевого $q_{\text{ц}}$, $q_{\text{ф}} < q_{\text{ц}}$, а $u^- \in U$ – управление, изымающее ресурсы у приложения $A_l^* \in A$, если $q_{\text{ф}} > q_{\text{ц}}$.

Учитывая монотонный характер зависимости q_{ki}, $k = \overline{1, M_i}$, $i = \overline{1, K}$, от r, докажем необходимые утверждения, предварительно сделав следующие выкладки.

Использование ресурсов приложениями зададим матрицей $\hat{\mathrm{P}} = \left\| \hat{\rho}_{lj} \right\|$, $l = \overline{1, I}$, $j = \overline{1, m}$, причем

$$\hat{\rho}_{lj} = \begin{cases} n_{lj} \Delta r_j, \text{если } l\text{-е приложение использует } j\text{-й ресурс;} \\ 0, \text{если } l\text{-е приложение не использует } j\text{-й ресурс,} \end{cases} \tag{3.21}$$

где n_{lj} – количество квантов j-го ресурса, выделенного l-му приложению; Δr_j – размер кванта j-го ресурса. При этом должны выполняться такие ограничения:

$$\Delta r_j \sum_{l=1}^{I} n_{lj} \le r_j, \quad \forall j = \overline{1, m}. \tag{3.22}$$

Тогда можно определить следующее отображение:

$$\hat{\Omega} = U \times \mathrm{P} \times \bar{\hat{a}} \to Q, \tag{3.23}$$

где $\bar{\hat{a}} = \{\hat{a}\}$ – множество векторов $\hat{a} \in \bar{\hat{a}}$.

В свою очередь,

$$\hat{\mathcal{U}} = Q^* \times \mathrm{P} \times \bar{\hat{a}} \to U. \tag{3.24}$$

Утверждение 1. Для заданных значений q_{ki}, $k = \overline{1, M_i}$, $i = \overline{1, K}$, в случае, когда выполняется условие $q_{ki}^* < q_{ki}$, существует управление $u^+ \in U$, позволяющее обеспечить равенство $q_{ki}^* = q_{ki}$ при минимальных ресурсах $(\min r)$ на поддержание уровня услуг.

Доказательство следует из монотонности функций (3.17)–(3.20), конечности множеств Q^*, P и $\bar{\hat{a}}$ и сопоставимости целей процессов на рис. 3.1.

Управление $u^+ \in U$ находится итеративно.

Для доказательства следующего утверждения введем отображение

$$\hat{\mathcal{F}} = Q \times \bar{\hat{a}} \to R. \tag{3.25}$$

АУТСКИ

Утверждение 2. При выполнении условий (3.12) и $q^*_{ki} < q_{ki}$, если известны значения $\hat{a}$, то управление $u^+ \in U$, позволяющее восстановить равенство $q^*_{ki} = q_{ki}$, $k = \overline{1, M_i}$, $i = \overline{1, K}$, может быть найдено без использования итерационных процедур.

Доказательство. Если значения вектора $\hat{a}$ фиксированы, то, исходя из (3.25), можно утверждать, что существуют зависимости $q_{\text{ц}} \to r$, $q_1 \to r_1$ и $q_2 \to r_2$. Тогда, по аналогии с $q_1 - q_2 \to r_1 - r_2 = \Delta r_{12}$, при выполнении условия $q^*_{ki} < q_{ki}$ для обеспечения выполнения равенства $q^*_{ki} = q_{ki}$ необходимо *l*-му приложению, поддерживающему *i*-ю услугу с учетом (3.21), выделить дополнительно ресурс $\Delta\hat{\rho}_{lj}$, $i = \overline{1, K}$, $l = \overline{1, I}$, $j = \overline{1, m}$, причем величина $\Delta\hat{\rho}_{lj}$ может быть определена на основе соотношения $\Delta q_{ki} = q_{ki} - q^*_{ki} \to \Delta\hat{\rho}_{lj}$. В общем случае выделение дополнительных ресурсов с учетом ограничения (3.22) без итераций может быть осуществлено только при выполнении условия (3.12), что и доказывает утверждение.

Следствие. Если выполняется условие $q^*_{ki} < q_{ki}$ и *i*-я услуга поддерживается приложением $A^*_l \in A$, то за счет приложений из множества *A*, для которых выполняется условие $q_{\text{ф}} > q_{\text{ц}}$, можно высвободить объем ресурсов $\Delta\hat{\rho}_{в}$. Причем выполняется условие $\Delta\hat{\rho}_{в} \geq \Delta\hat{\rho}_{lj}$, где $\Delta\hat{\rho}_{lj}$ – дополнительный объем ресурсов, который нужно выделить *l*-му приложению для того, чтобы обеспечить выполнение условия $q^*_{ki} = q_{ki}$. Тогда управление $u^+ \in U$ в условиях дефицита ресурсов позволяет восстановить уровень *i*-й услуги, поддерживаемой *l*-м приложением за один проход.

Реализацию процедуры принятия решений на этом уровне целесообразно осуществлять на основе системы с координатором (см. п. 3.2).

Исследование предложенного декомпозиционно-компенсационного способа организации управления уровнем услуг не только подтвердило его работоспособность, но и выявило способность повышения эффективности использования ресурсов. Предлагаемый способ сравнивался с методом полного резервирования ресурсов и методом наращивания нод, описанных ниже, по эффективности использования ресурсов $q_{\text{Э}}$, определяемой как

$$q_{\text{Э}} = \frac{r_{и}}{r_{в}}, \tag{3.26}$$

где $r_{и}$ – количество фактически используемых ресурсов, а $r_{в}$ – количество зарезервированных и выделенных ресурсов.

Наиболее распространенный на практике метод полного резервирования предполагает определение количества ресурсов $r_{\max,l}$ на основе выражения (3.9) для максимального количества $a_{\max,l}$ пользователей *l*-й услуги, оговоренного в SLA. Объем ресурсов $r_{\max,l}$ закрепляется за приложением A_l и в процессе работы

АУТСКИ

не изменяется. Если фактическое количество клиентских запросов меньше максимального, т. е. выполняется условие $a_l < a_{\max,l}$, то ресурсы используются неэффективно. Наоборот, при выполнении условия $a_l > a_{\max,l}$ качество услуг снижается, управление для повышения уровня услуг не осуществляется, и пользователи довольствуются фактическим качеством услуг.

Метод поддержания качества услуг наращиванием нод при горизонтальном масштабировании отслеживает не уровень услуг, а процент утилизации выделенных ресурсов. При достижении задействованности отдельных ресурсов на 95% проводится увеличение на размер ноды всех видов ресурсов независимо от реальной потребности увеличения объемов только некоторых из них. При этом незадействованные ресурсы не могут быть использованы другими приложениями.

Зависимость эффективности использования ресурсов $q_Э$ от соотношения $a_l / a_{\max,l}$ приведена на графиках рис. 3.4 для предлагаемого метода (кривая 1), метода полного резервирования ресурсов (кривая 2) и метода наращивания нод (кривая 3) при разном количестве d_l ресурсов, используемых приложением A_l, независимо от количества клиентских запросов. Для проведения исследований величина кванта ресурсов устанавливалась на уровне 10% от размера ноды.

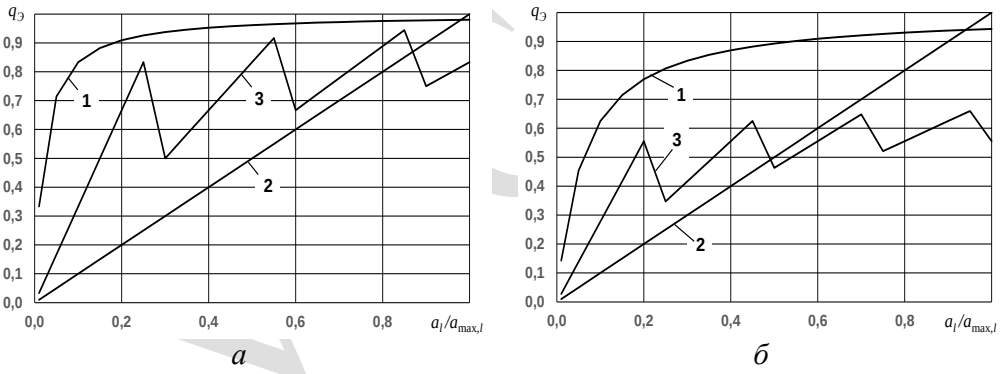


Рис. 3.4. Зависимости $q_Э$ от $a_l / a_{\max,l}$ для пропорциональных (*а*) и непропорциональных (*б*) потребностей в ресурсах (1 – предлагаемый метод; 2 – метод полного резервирования ресурсов; 3 – метод наращивания нод)

Анализ графиков, приведенных на рис. 3.4, подтверждает, что предлагаемый метод гораздо эффективнее использует ресурсы ИТ-инфраструктуры, причем эффективность использования ресурсов ИТ-инфраструктуры увеличивается по мере уменьшения значения соотношения $a_l / a_{\max,l}$.

Новый класс СУ – многообъектные многоуровневые СУ – породил новые проблемы и задачи управления, выходящие за рамки традиционных систем автоматического управления и регулирования, среди которых наиболее актуальными являются задачи координации и принятия решения. Анализу СУИ, как двухуровневой СУ с координатором, посвящен п. 3.2.

3.2. ПРЕДСТАВЛЕНИЕ СУИ В ВИДЕ ДВУХУРОВНЕВОЙ СИСТЕМЫ С КООРДИНАТОРОМ

3.2.1. Модель двухуровневой СУИ

В связи с тем, что в настоящее время исследования теории управления заострены на многообъектности, распределенности, большой размерности, возникает потребность выделения специального класса многообъектных распределенных систем управления [2, 73], к которым относятся СУИ. Одним из важнейших вопросов, решаемых при проектировании и эксплуатации таких СУ, кроме разработки архитектуры является задача принятия решений. Сложность принятия решений в иерархических СУ, таких как СУИ, обусловлена тем, что решения принимаются на разных уровнях иерархии за ограниченное время. В [73] задачи принятия решений с ограничением времени называются задачами принятия оперативных решений. В [68] предложено такие системы рассматривать как двухуровневые СУ с координатором. Основанием для выделения в СУИ двух уровней является то, что при управлении уровнем услуг СУИ функционирует в различных режимах в условиях неопределенности, неполноты и недостоверности информации, наличия факторов риска, множества конфликтующих критериев и целей подсистем СУИ. От таких СУ требуется не достижение оптимального функционирования ИТ-инфраструктуры, что практически невозможно, а улучшение качественных характеристик работы ИТ-инфраструктуры. В таких случаях оправдано построение двухуровневых систем с координатором, когда координатор согласовывает самостоятельные решения и действия подсистем СУИ для улучшения работы ИТ-инфраструктуры в целом с точки зрения качества предоставляемых услуг. При этом действия координатора должны быть направлены на улучшение глобальной функции качества предоставления услуг, а принятие им решений осуществляется в условиях неопределенности.

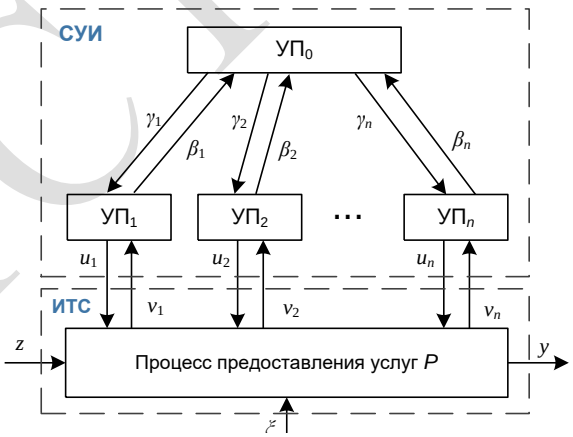


Рис. 3.5. Представление СУИ в виде двух уровневой системы

На рис. 3.5 приведена модель СУИ, осуществляющей управление уровнем услуг, в виде двухуровневой системы с координатором [2, 68].

Расположение управляющих подсистем (УП) отображает иерархическую структуру СУИ. Модель состоит из расположенной выше по иерархии управляющей подсистемы ($УП_0$), n расположенных ниже управляющих подсистем ($УП_1,\ldots,УП_n$) и управляемого процесса P.

Взаимодействие УП по вертикали осуществляется следующим образом. Команды, сигналы, воздействия или вмешательства (входы) $\gamma_1,...,\gamma_n$, передаваемые от $УП_0$ к $УП_1,\ldots,УП_n$, являются *координирующими*. Командные сигналы или воздействия (входы) $(u_1,...,u_n)$ от $УП_1,\ldots,УП_n$ к процессу P являются *управляющими*.

Снизу вверх поступают сигналы обратной связи или информационные сигналы $v_1,...,v_n$ и $\beta_1,...,\beta_n$. Описание двухуровневой СУИ может быть осуществлено посредством терминальных переменных (входов и выходов). В этом случае УП описываются как функциональные подсистемы, выходы которых однозначно определяются входами [68].

Процесс *P* можно описывать как управляемую подсистему, на которую воздействуют управляющие сигналы *u* от $УП_1,\ldots,УП_n$, $u \in U$, U – множество управляющих воздействий; поступают входные сигналы *z*, $z \in Z$, представляющие собой запросы пользователей, и сигналы ξ, $\xi \in \Xi$, являющиеся возмущающими воздействиями. К возмущающим воздействиям Ξ относятся неисправности в ИТ-инфраструктуре, функциональные отказы РОУ, запросы пользователей, которые, являясь помехой для запросов рассматриваемых пользователей, затрудняют достижение целей управления. Выходом процесса *P* является *y*, $y \in Y$, где Y – множество выходов процесса *P*, ответы ИТ-инфраструктуры на запросы пользователей.

Процесс *P* можно представить в виде следующего отображения на основе декартова произведения:

$$P : U \times Z \times \Xi \to Y . \tag{3.27}$$

Множество *U* управляющих сигналов, воздействующих на процесс *P* со стороны $УП_1,\ldots,УП_n$, удобно представлять в виде декартова произведения *n* множеств [2, 68]:

$$U = U_1 \times U_2 \times ... \times U_n. \tag{3.28}$$

При этом каждая *g*-я управляющая подсистема $УП_g$, $g = \overline{1,n}$, обладает полномочиями выбора *g*-го компонента u_g управляющего воздействия *u* для оказания непосредственного влияния на процесс *P*.

На входы каждой *g*-й управляющей подсистемы $УП_g$, $g = \overline{1,n}$, поступают два сигнала: координирующий сигнал $\gamma_g \in \Gamma$ от $УП_0$ и информационный сигнал обратной связи v_g в виде данных мониторинга. Управляющим выходом $УП_g$, $g = \overline{1,n}$, является воздействие u_g, выбираемое $УП_g$ из множества U_g. Предположим, что каждая из $УП_g$, $g = \overline{1,n}$ реализует отображение C_g, $g = \overline{1,n}$, такое, что

$$C_g : \Gamma \times V_g \to U_g , \tag{3.29}$$

где V_g – множество данных мониторинга v_g, поступающих в СУИ от ИТ-инфраструктуры, $v_g \in V_g$. Данные мониторинга V_g, $g = \overline{1,n}$, являются сигналами обратной связи для локального контура управления на основе $УП_g$, $g = \overline{1,n}$.

Сигналы обратной связи v_g, поступающие на вход $УП_g$, $g = \overline{1,n}$, получены в результате мониторинга ИТ-инфраструктуры. Они содержат информацию относительно протекания процесса *P*. Естественно, эти сигналы функционально зависят от управляющих сигналов *u*, входов *z*, возмущений ξ и выходов *y*. Эту зависимость можно представить отображением [2, 68]

АУТСКИЙ

$$f_g : U \times Z \times \Xi \times Y \to V_g . \quad (3.30)$$

Управляющая подсистема $УП_0$ является координатором и вырабатывает координирующие сигналы $\gamma_g \in \Gamma$, $g = \overline{1,n}$, причем сигнал γ_g с g-го выхода $УП_0$ поступает только на g-й вход расположенной ниже управляющей подсистемы $УП_g$, $g = \overline{1,n}$. Координатор $УП_0$ вырабатывает сигнал на основе анализа информации, поступающей на его вход от $УП_g$, $g = \overline{1,n}$, и представляющей собой сигналы обратной связи и обобщенную информацию о состоянии и функционировании ИТ-инфраструктуры. В этом случае можно считать, что в координаторе реализуется отображение C_0 такое, что

$$C_0 : \mathrm{B} \to \Gamma , \quad (3.31)$$

где B – множество информационных сигналов β, реализующих обратную связь. Причем $\beta = (\beta_1,...,\beta_n)$ представляет собой совокупность сигналов β_g, $g = \overline{1,n}$, обратной связи, поступающих в координатор $УП_0$ от подсистем $УП_g$, $g = \overline{1,n}$.

Аналогично (3.30) сигнал обратной связи β, поступающий в $УП_0$, содержит информацию о состоянии всех расположенных ниже подсистем, поэтому он определяется отображением

$$f_0 : \Gamma \times V \times U \to \mathrm{B} , \quad (3.32)$$

где $V = V_1 \times ... \times V_n$. Таким образом, B является функцией координирующих сигналов γ_g, $g = \overline{1,n}$, сигналов обратной связи $v = (v_1,...,v_n)$, поступающих в $УП_g$, $g = \overline{1,n}$, и управляющих воздействий $u = (u_1,...,u_n)$.

На модели, изображенной на рис. 3.5, не показано в явном виде взаимодействие между подсистемами $УП_1,\ldots,УП_n$, так же, как не показано непосредственное влияние $УП_0$ на функционирование ИТ-инфраструктуры и получение координатором $УП_0$ сигналов обратной связи непосредственно от элементов ИТ-инфраструктуры, что имеет место в реальных СУИ.

В соответствии с [68] координация заключается в воздействии на подсистемы управления $УП_g$, $g = \overline{1,n}$, заставляющем действовать их согласованно, подчиняя действия УП единой политике, ориентированной на достижение глобальной цели системы, несмотря на то, что эта цель может противоречить локальным целям подсистем. Координацию осуществляет $УП_0$, причем именно координатор должен преодолеть противоречия между локальными целями подсистем $УП_g$, $g = \overline{1,n}$.

Успешность деятельности координатора по организации согласованных действий $УП_g$, $g = \overline{1,n}$, оценивается тем, насколько успешно достигается глобальная цель управления ИТ-инфраструктурой. Достижение цели координатором можно рассматривать как решение задачи, которая формализуется как задача принятия решения и заключается в оценке результативности координации. Поскольку эта задача определяется относительно всех подсистем, включая процесс P, то она называется глобальной решаемой задачей [2, 68].

АУТСКИ

Для двухуровневых систем должны быть обеспечены координируемость по отношению к задаче, решаемой $УП_0$, и координируемость по отношению к глобальной задаче [68]. Первое означает, что сигналы $УП_0$ оказывают координирующее воздействие на задачи, решаемые $УП_g$, $g=\overline{1,n}$, а второе, что координатор способен влиять на $УП_g$, $g=\overline{1,n}$, так, что их совместное воздействие на процесс P направлено на решение глобальной задачи.

Успешное функционирование СУИ, отвечающей двухуровневой модели, может быть обеспечено только тогда, когда цели подсистем согласованы между собой и согласованы с глобальной целью системы [2, 68]. В двухуровневой системе выделяют три типа целей (задач): глобальную цель, цель координатора $УП_0$ и цели $УП_g$, $g=\overline{1,n}$. Необходимость совместимости целей или задач вытекает из следующих особенностей. На процесс P непосредственно воздействуют только $УП_g$, $g=\overline{1,n}$, поэтому глобальная цель может быть достигнута только опосредованно через действия $УП_g$, $g=\overline{1,n}$, которые должны быть скоординированы относительно глобальной цели, а также цели координатора. Глобальная цель – повышение эффективности выполнения бизнес-процессов – выходит за рамки непосредственной деятельности двухуровневой системы, приведенной на рис. 3.5, и ни одна из подсистем $УП_g$, $g=\overline{1,n}$ не ориентирована на достижение глобальной цели или решение глобальной задачи. Глобальная задача может быть решена только совместными воздействиями всех управляющих подсистем $УП_g$, $g=\overline{1,n}$.

Глобальная цель. Учитывая тот факт, что ИТ-инфраструктуры создаются для повышения эффективности производственных и бизнес-процессов, глобальную цель СУИ можно определить как обеспечение максимального качества $\mathfrak{Q}$ ИТ-сервисов с минимальными затратами $\mathfrak{C}$. Так, целью процессного управления в соответствии с ITSM и ISO является постоянное повышение уровня ИТ-услуг [137, 262, 263], что формально можно записать как $\max \mathfrak{Q}$.

Максимальное качество предоставления услуг в ИТ-инфраструктуре будет достигаться в том случае, когда

$$\max \mathfrak{Q} \Leftrightarrow \max Q_i, i=\overline{1,K} \Leftrightarrow \max q_{ki}, i=\overline{1,K},\ k=\overline{1,M_i}, \tag{3.33}$$

где $Q_i, i=\overline{1,K}$ – качество i-й услуги; $q_{ki}, k=\overline{1,M_i}$, – значение k-го показателя качества i-й услуги.

Для достижения цели процессного управления необходимо непрерывно наращивать ресурсы ИТ-инфраструктуры, что неприемлемо прежде всего с экономической точки зрения. Однако повышение экономической эффективности ведения бизнеса требует сокращения затрат на ИТ-инфраструктуру, т. е. действий, нацеленных на достижение $\min \mathfrak{C}$. Поддержание качества услуг на этом уровне является основной задачей координатора.

Цель координатора. Цель координатора – поддержание качества $\mathfrak{Q}$ услуг на согласованном уровне с минимальными затратами $\mathfrak{C}$ на задействуемые ресурсы. Цель координатора можно формализовать следующим образом:

$$\mathfrak{Q} = \text{const}\big|_{\min \mathfrak{C}}. \tag{3.34}$$

АУПСКИ

Выражение (3.34) означает, что координатор из всех возможных вмешательств будет выбирать такие, которые требуют минимальной стоимости реализации.

Требование поддержания согласованного уровня услуг касается всех услуг и отдельных показателей качества услуг:

$$\mathfrak{Q} = \text{const} \Leftrightarrow Q_i = \text{const}, \; i = \overline{1,K} \Leftrightarrow q_{ki} = \text{const}, \; k = \overline{1,M_i}, \; i = \overline{1,K}. \qquad (3.35)$$

Здесь необходимо оговорить следующее обстоятельство. Основным способом повышения качества i-й услуги, $i = \overline{1,K}$, является выделение дополнительных ресурсов приложениям, поддерживающих работу i-й услуги. При превышении уровнем i-й услуги целевого значения проводится сокращение ресурсов, выделенных соответствующим приложениям, как этого требует критерий $\min \mathfrak{C}$. В то же время последний сервер, предоставляющий i-ю услугу, не может быть отключен, несмотря на то, что качество этой услуги по-прежнему выше требуемого, поскольку это приведет к полному прекращению предоставления услуги. Таким образом, всегда будет какой-то фиксированный минимум затрат $\mathfrak{C}^*$, после чего дальнейшее сокращение затрат будет невозможно.

Локальные цели. Целью локального управления является поддержание заданных значений параметров функционирования ИТ-инфраструктуры с минимальными затратами. В модели СУИ, приведенной на рис. 3.5, управляющие подсистемы УП_g, $g = \overline{1,n}$ могут иметь собственные различающиеся цели функционирования.

3.2.2. Определение принципа координации и синтез координатора

Выполнение требований координируемости и совместимости выступает ограничением при определении стратегий, которыми может руководствоваться координатор. Предложенные в [2, 68] принципы координации, основанные на постулате совместимости, не могут быть использованы в СУИ, поскольку они предполагают либо получение и использование точного прогноза значений параметров процесса P, либо требуют знания вида функций или аналитических выражений для решения задачи координации. Для решения проблемы координации необходимо после декомпозиции глобальной задачи произвести синтез координатора и определиться с методами, процедурами или алгоритмами координации.

Перепишем цель (3.35) координатора в следующем виде [122]

$$\min \Delta Q_i = \min(Q_i - Q_i^*), \Leftrightarrow \min \Delta q_{ki} = \min(q_{ki} - q_{ki}^*), \; k = \overline{1,M_i}, \; i = \overline{1,K}, \qquad (3.36)$$

где Q_i и Q_i^* – целевое и фактическое значение качества i-й услуги; q_{ki} и q_{ki}^* – целевое и фактическое значение показателя качества i-й услуги, причем фактическое качество считается хуже требуемого при $Q_i > Q_i^*$ и, соответственно, при $q_{ki} > q_{ki}^*$.

На входы процесса P поступают управляющие и возмущающие воздействия, а задача СУИ сводится к выбору управления, противодействующего возмущению. Исходя из (3.36), координатор должен сравнивать текущие значения показателей качества q_{ki}^*, $k = \overline{1,M_i}$, $i = \overline{1,K}$ услуг, предоставляемых процессом P поль-

зователям, с целевыми значениями $q_{ki}, k=\overline{1,M_i}, i=\overline{1,K}$ и вырабатывать координирующие сигналы, минимизирующие отклонение. При управлении, направленном на поддержание согласованного уровня, естественным является использование принципа управления по отклонению [7]. В этом случае выходы процесса P после соответствующих преобразований, заключающихся в сведении метрик для определения значений $q_{ki}, k=\overline{1,M_i}, i=\overline{1,K}$, по цепи обратной связи поступают в координатор, где сравниваются с целевыми значениями. На основании отклонения $\Delta q_{ki}, k=\overline{1,M_i}, i=\overline{1,K},$ вырабатываются координирующие сигналы для УП$_g$, $g=\overline{1,n}$.

В СУИ могут быть измерены основные возмущающие воздействия, к которым относится количество $\hat{a}=\{a_l, l=\overline{1,I}\}$ пользователей услуг, влияние неисправностей на качество сервиса, загруженность каналов связи и др. Это позволяет совместно с управлением по отклонению использовать принцип управления по возмущению и реализовать в СУИ комбинированное управление, при этом больший вес имеет управление по отклонению. На рис. 3.6 приведен результат декомпозиции координатора, реализующего комбинированный принцип управления уровнем услуг.

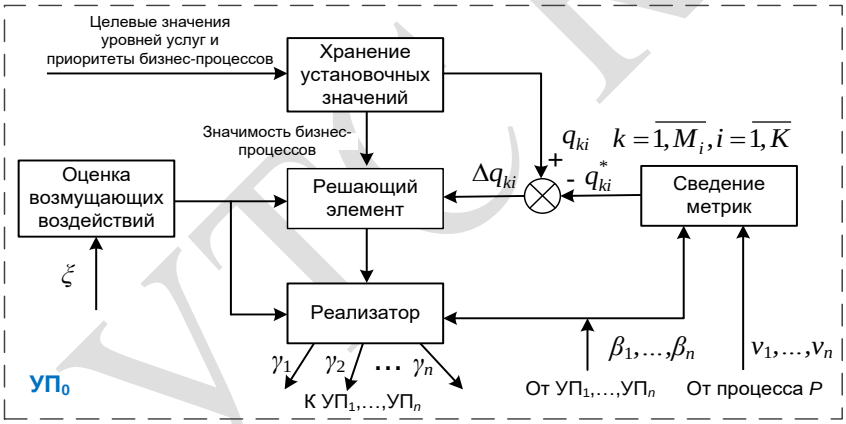


Рис. 3.6. Структура координатора, использующего обратную связь и учитывающего возмущающие воздействия

Координатор, приведенный на рис. 3.6, содержит контур отрицательной обратной связи и цепи для компенсации возмущающих воздействий $\xi \in \Xi$. Компенсационный контур оценивает основные возмущения, которые учитываются при выборе корректирующих сигналов.

Глобальная цель управления ИТ-инфраструктурой может изменяться, что проявляется для координатора изменением приоритетов Pr приложений $\{A_l\}$ и целевых значений $Q_i, i=\overline{1,K},$ уровня услуг. В этом случае выражение (3.31) примет вид

$$C_0 : Pr \times Q_i \times \mathrm{B} \times V \times \Xi \to \Gamma, i=\overline{1,K}. \qquad (3.37)$$

После определения принципа управления необходимо определить стратегию и правила координатора, а также условия применения стратегии или правил.

Выбор координирующего воздействия определяет система

$$< Pr, \Delta\hat{q}, \mathrm{B}, V, \Xi, \mathcal{N} >, \tag{3.38}$$

где $\Delta\hat{q}$ – вектор отклонения; $\mathcal{N}$ – ситуационная неопределенность.

Рассмотрим систему (3.38). Приоритеты Pr приложениям задаются системой управления выполнением бизнеса BPM и принимают значения из множества $\{1,2,...,Pr_m\}$, где Pr_m – максимальное значение приоритета. В процессе функционирования ИТ-инфраструктуры приоритеты приложений Pr изменяются, отслеживая изменения значимости бизнес-процессов.

Для характеристики степени отклонения значений элементов вектора $\Delta\hat{q} = (\Delta q_{1,1},...,\Delta q_{k,i},...,\Delta q_{M_K,K})$ от целевых значений по аналогии с [43] введем функцию $w(\Delta q_{ki})$, $k = \overline{1,M_i}$, $i = \overline{1,K}$, принимающую значения на отрезке [-1,1] и определяющую степень близости фактического уровня качества к целевым значениям

$$w(\Delta q_{k,i}) = (q_{ki} - q_{ki}^{*}) / q_{\mathrm{кр}ki}, \tag{3.39}$$

где $q_{\mathrm{кр}_{k,i}}$ – критическое значение показателя качества i-й услуги, при котором качество считается неудовлетворительным. Причем при $w(\Delta q_{k,i}) \in (0,1]$ фактическое значение показателя качества i-й услуги лучше, чем требуемое, при $w(\Delta q_{k,i}) \in [-1,0)$ качество услуги хуже согласованного.

Из всех видов неопределенностей [42] наиболее характерной для ИТ-инфраструктуры или ИУС являются ситуационная неопределенность $\mathcal{N}$, характеризуемая непредвиденными действиями пользователей, непрогнозируемостью нештатных ситуаций, трудностью определения реакции ресурсов на комбинацию воздействующих факторов. При этом возникает задача выработки координирующих $\gamma = (\gamma_1,...,\gamma_n)$ и управляющих $u = (u_1,...,u_n)$ воздействий по сигналам обратной связи $v = (v_1,...,v_n)$ и $\beta = (\beta_1,...,\beta_n)$ при воздействии возмущений $\xi \in \Xi$ в условии неопределенности $\mathcal{N}$. Поскольку в аналитическом виде определить соответствующее отображение не представляется возможным, то выходом из ситуации является использование итеративной процедуры координации [122], предполагающей участие всех процессов, реализующих управление уровнем услуг, приведенных на схеме на рис. 3.1. Применение итеративной процедуры позволяет выработать приемлемые координирующие воздействия ввиду монотонности функций (3.17)–(3.20), а также монотонности влияния ситуационной неопределенности $\mathcal{N}$ на уровень услуг. Таким образом, в данной работе раскрытие неопределенности осуществляется использованием итеративных процедур управления.

Основная функция координатора заключается в согласовании деятельности УП$_g$, $g = \overline{1,n}$, при генерации ими собственных решений так, чтобы повысить суммарный эффект от их совместных действий. Поэтому решения, принимаемые координатором, оказывают влияние на выбор координирующих, а не управляющих

воздействий [68]. Для выбора координирующих воздействий необходимо определить принцип координации. В СУИ координирующие воздействия указывают на то, какому из УП$_g$, $g=\overline{1,n}$, отдается предпочтение при восстановлении качества услуг и какие методы целесообразно использовать (рис. 3.7). Например, при перегрузке каналов связи выделение дополнительных вычислительных ресурсов приложениям из множества $\{A_l\}$ не восстановит уровень услуг. Поэтому координатор УП$_0$ сообщает УП$_g$, отвечающей за управление потоками в сети, о необходимости ограничения исходящего трафика приложений из $\{A_l\}$, имеющих низший приоритет. Такую координацию можно реализовать, например, путем использования основанного на системе продукций принципа координации. Правила в нотации Бекуса – Наура [304] имеют следующий вид:

$$
\begin{aligned}
&< \textit{система\_правил} >::=[< \textit{продукция} >]\\
&< \textit{продукция} >::=< \textit{условие} >\rightarrow< \textit{следствие} >\\
&< \textit{условие} >::=[< \textit{простое\_условие} >]\\
&< \textit{простое\_условие} >::=< \textit{объект} >< \textit{атрибут} >< \textit{предикат} >< \textit{значение} >\\
&< \textit{объект} >::= \textit{УП}_g,\ i=\overline{1,n}\\
&< \textit{следствие} >::=[< \textit{указание} >|< \textit{формула} >|< \textit{программа} >]\\
&< \textit{предикат} >::=|\neq|<|>|\leq|\geq\\
&< \textit{атрибут} >::=< \textit{приложение} >|< \textit{приоритет\_приложения} >|< \textit{услуга} >|\\
&|< \textit{бизнес\_процесс} >|< \textit{важность\_процесса} >|< \textit{параметр} >|< \textit{статус\_ИТС} >
\end{aligned}
\qquad (3.40)
$$

Применение основанного на продукциях принципа координации оправдано в случаях, когда ставится цель улучшения качества предоставления услуг, а не достижения оптимальных показателей функционирования ИТ-инфраструктуры в условиях недостаточности информации о факторах, влияющих на результаты координирующих и управляющих воздействий.

На рис. 3.7 продемонстрированы возможные реакции СУИ на изменение качества предоставляемых услуг. СУИ компенсирует незначительное ухудшение качества услуги, выделяя дополнительные объемы ресурсов. При существенном ухудшении качества разворачиваются дополнительные экземпляры приложения. Если фактическое качество услуги лучше согласованного, то осуществляется постепенное уменьшение объемов вычислительных и коммуникационных ресурсов, выделяемых этой услуге.

Отображение (3.31) может иметь очень сложный вид, а для системы (3.38) результирующую взаимосвязь между координирующими $\gamma=(\gamma_1,...,\gamma_n)$, управляющим $u=(u_1,...,u_n)$ воздействиями и выходом процесса P аналитически получить невозможно. В этом случае можно использовать программное управление [76].

Основными процедурами координации являются использование итеративных процедур по улучшению координирующих сигналов на основании анализа результатов координации либо использование обратной связи для коррекции координирующего сигнала [122]. В данной работе применяются оба типа процедур. В обоих случаях для определения сигнала ошибки при оценке воздействия необходимо провести сведение метрик, измеряемых на уровне процесса P, к метрикам, которыми оперирует координатор.

АИС КП

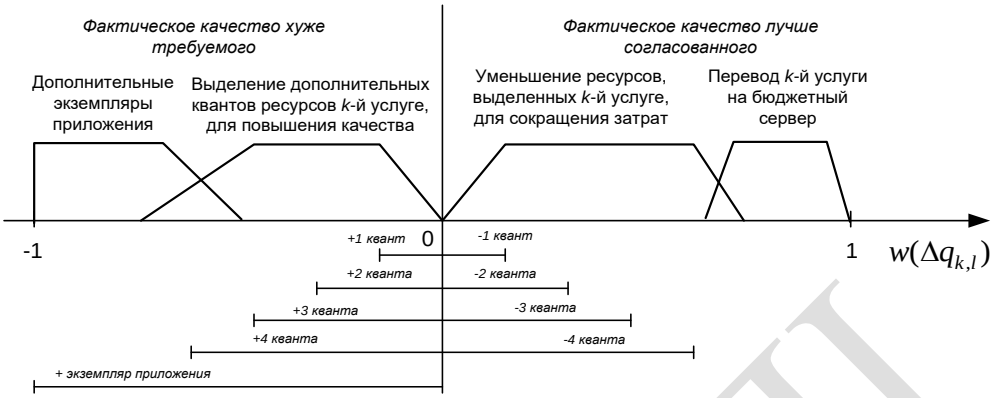


Рис. 3.7. Пример выбора действий УП$_g$, $g=\overline{1,n}$, в зависимости от значения функции $w(\Delta q_{ki})$

Необходимо отметить, что координатор используется при автоматическом режиме управления уровнем услуг, а при автоматизированном управлении роль координатора выполняет СППР.

3.3. Особенности декомпозиционно-компенсационного способа организации управления уровнем услуг при использовании сервисно-ресурсных моделей

Предложенный декомпозиционно-компенсационный способ организации управления уровнем услуг с использованием СРМ имеет ряд особенностей.

Во-первых, выделяются услуги $\mathcal{G}$, влияющие на бизнес-сервисы, и определяется перечень параметров q_{ki}, $k=\overline{1,M_i}$, $i=\overline{1,K}$, определяющих качество этих услуг. Во-вторых, на основании СРМ выделяются характеристики функционирования ИТ-инфраструктуры, которые влияют на качество сервиса и которые можно измерять и контролировать, а также проводится сбор и анализ значений этих характеристик. В-третьих, значения показателей функционирования ИТ-инфраструктуры сводятся к обобщенным показателям $q^*_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, которые сравниваются с целевыми значениями $q_{b,ki}$, $b=\overline{1,D}$, $k=\overline{1,M_i}$, $i=\overline{1,K}$, показателей уровня услуг. При нарушении SLA подсистема анализа уведомляет об этом администратора ИТ-инфраструктуры. В-четвертых, администратор посредством СУИ выбирает решение, позволяющее компенсировать снижение уровня услуги s_i, произошедшее из-за воздействия ряда факторов, например, увеличения количества пользователей a_l услуги A_l выделением дополнительных ресурсов $\Delta\hat{\rho}_{lj}$, $i=\overline{1,K}$, $l=\overline{1,I}$, $j=\overline{1,m}$, приложению A_l, поддерживающему услугу s_i. Для этого администратор задействует подсистему оптимизации, которая, в свою очередь, использует информацию из базы знаний СУИ.

В качестве примера рассмотрим реализацию управления по поддержанию услуги $s_{1,1}$ на уровне $q_{1,1,1}$ (рис. 3.8) с использованием СРМ (см. табл. 2.2).

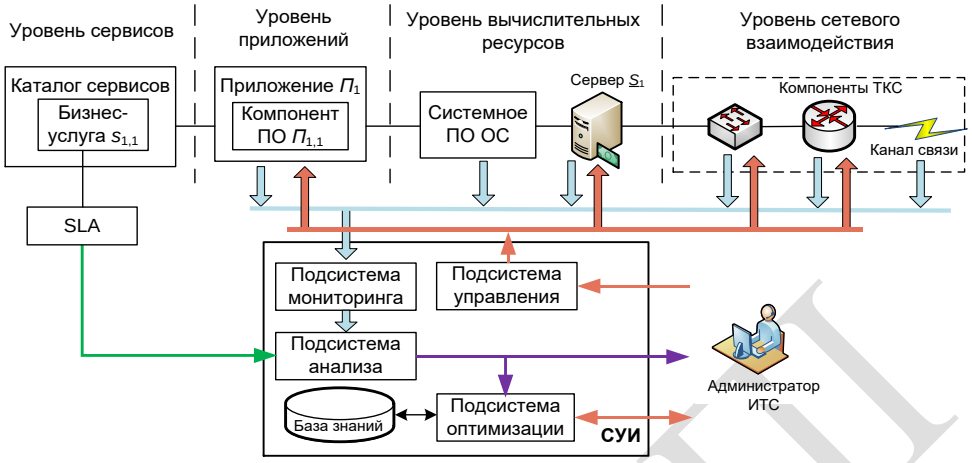


Рис. 3.8. Схема управление уровнем услуги $s_{1,1}$ с использованием СРМ

Исходя из п. 2.3.2, где приведен пример SLA с показателями уровня услуг, можно утверждать, что ключевым показателем $q_{1,1,1}$ качества услуги $s_{1,1}$ является показатель производительности «Время открытия формы». Фактическое значение $q^*_{1,1,1}$ зависит от значений параметров правой части выражения (2.5). При $a_1^* > a_{\max,1}$, где a_1^* – количества открытых сессий услуги $s_{1,1}$, $a_{\max,1}$ – максимальное количество сессий, оговоренных в SLA, увеличивается время Δt_{S_1} отклика сервера $\underline{S}_1$ на запросы пользователей, а также происходит увеличение других составляющих $\Delta t_{1,1}$, что приводит к снижению качества предоставления услуги $s_{1,1}$. Ограничение a_1^* до значения $a_{\max,1}$ позволит восстановить уровень услуги $s_{1,1}$ до целевого значения $q_{1,1,1}$, но при этом пострадают интересы бизнеса из-за отключения части пользователей от $s_{1,1}$.

Применение предлагаемого способа организации управления позволяет сохранить как количество открытых сессий при $a_1^* > a_{\max,1}$, так и качество услуги $s_{1,1}$ на уровне $q_{1,1,1}$. В этом случае СУИ предложит администратору выбрать один из трех методов восстановления уровня услуги $s_{1,1}$, приведенных в табл. 3.1.

Таблица 3.1. Методы компенсации снижения уровня услуги $s_{1,1}$

№ п/п	Обозначение	Суть метода	Возможности	Затраты
1	M_1	Кластеризация приложения $П_1$	Большие	Существенные
2	M_2	Выделение компоненту ПО $П_{1,1}$ дополнительных ресурсов на сервере $\underline{S}_1$	Большие	Умеренные
3	M_3	Увеличение полосы пропускания ТКС для данных приложения $П_1$	Ограниченные	Незначительные

АУТСКИ

Для каждого из методов компенсации имеются графики зависимости степени повышения производительности (уменьшения задержки) от приведенных затрат (рис. 3.9). Графики могут иметь вид, подобный приведенному на рис. 3.2, если представляется возможность связать потери бизнеса со снижением качества услуги.

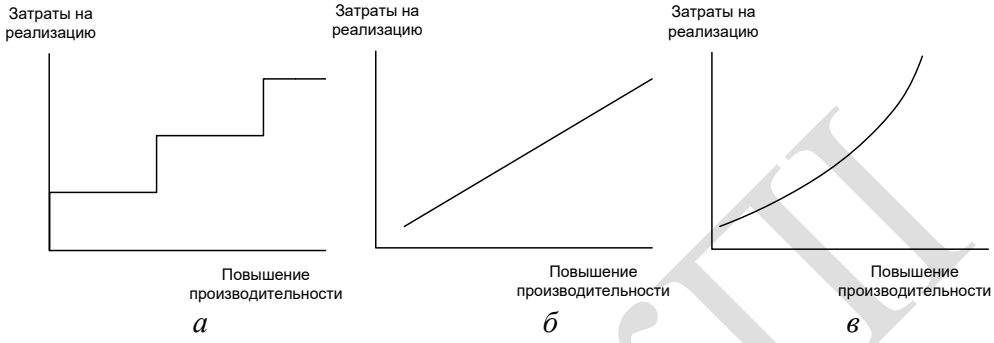


Рис. 3.9. Кривые зависимости повышения производительности от приведенных затрат для методов: *а* – M_1; *б* – M_2; *в* – M_3

Кривая затрат для метода M_1 (рис. 3.9, *а*) имеет ступенчатый вид, объясняемый дискретным увеличением затрат на выделение ресурсов $\Delta\hat{\rho}_{1j}$, $j=\overline{1,m}$, для размещения дополнительного экземпляра приложения $П_1$. Повышение производительности методом M_2 (рис. 3.9,*б*) пропорционально зависит от приведенных затрат, что объясняется примерно пропорциональным повышением производительности компонента ПО $П_{1,1}$ от выделенных для него ресурсов на сервере $\underline{S}_1$.

Для выбора одного из этих методов компенсации СУИ приводит графики к единому масштабу и при заданной степени повышения производительности $w(\Delta q_{ki})$, достаточной для приведения уровня услуги $s_{1,1}$ к уровню $q_{1,1,1}$, позволит администратору выбрать метод, требующий меньших затрат для своей реализации с учетом рисков (рис. 3.10). Так, для повышения производительности до точки A_1 следует выбирать метод M_3, до A_2 – M_2, до A_3 – M_1.

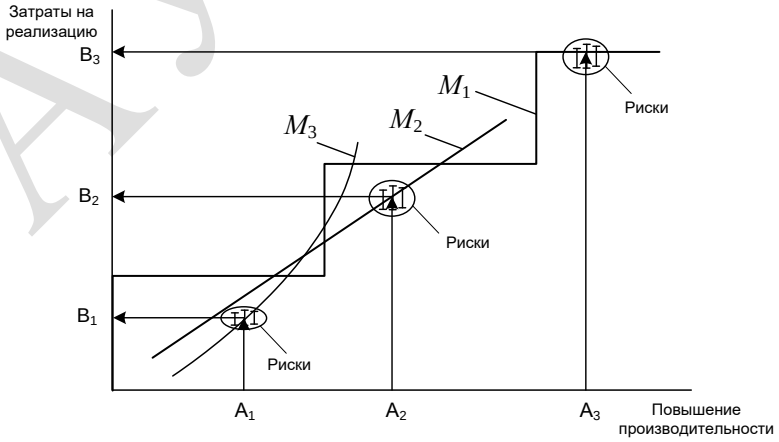


Рис. 3.10. Выбор метода компенсации для услуги $c_{1,1}$

Кроме учета приведенных затрат при выборе метода компенсации ИТ УИТИ руководствуется следующими соображениями: при соизмеримых затратах отдается предпочтение методу, компенсирующему увеличение соответствующей составляющей в правой части выражения (2.5); при выборе метода учитывается загруженность ТКС, поскольку при перегруженной ТКС методы компенсации M_1 и M_2 не дадут желаемого результата, а при незагруженной ТКС выбор метода M_3 не повлияет на повышение уровня сервиса. Реализация управления осуществляется посредством правил, подобных (3.40).

На рис. 3.11 изображен фрагмент СРМ с замкнутым контуром управления поддержания согласованного уровня услуг. В этом случае СУИ вместе с управлением по отклонению использует принцип управления по возмущению.

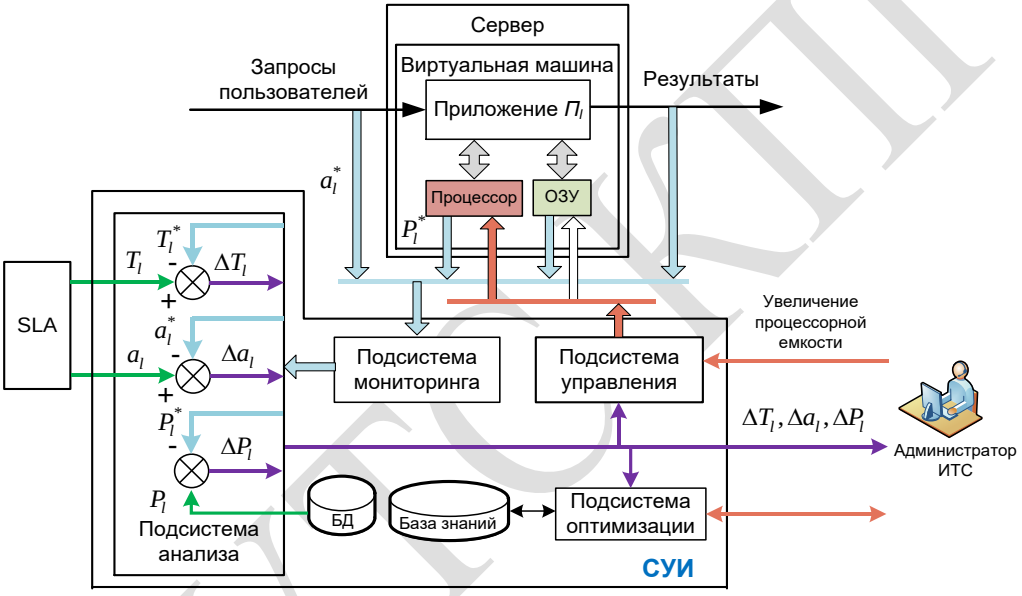


Рис. 3.11. Замкнутый контур поддержания услуг на согласованном уровне с комбинированным принципом управления

Подсистема мониторинга измеряет параметры, оказывающие влияние на качество услуги s_i, предоставляемой приложением $П_l$. Подсистема анализа сравнивает фактическое значение T_l^* показателя производительности услуги s_i с целевым значением T_l. При $\Delta T_l = T_l^* - T_l > 0$ подсистема анализа выдает сообщение о несоответствии уровня услуги s_i, оговоренному в SLA. В ходе дальнейшего анализа сравнивается фактическое количество a_l^* пользователей услуги s_i со значением $a_{\max,1}$. Кроме этого, проверяется загруженность процессора P_l^* и используемая память ВМ с приложением $П_j$. Если $\Delta a_l = a_l^* - a_l > 0$ и $\Delta P_l = P_l^* - P_l > 0$, то администратор, анализируя степень ухудшения сервиса по значению ΔT_l, превышение допустимого количества пользователей Δa_l услуги s_i и загруженность процессора ΔP_l ВМ, принимает решение об увеличении процессорной емкости приложению $П_l$.

Управление на уровне ресурсов может осуществляться автоматически (с использованием координатора), когда сигнал от подсистемы анализа инициирует автоматическое выделение дополнительной процессорной емкости приложению $П_l$.

Для реализации декомпозиционно-компенсационного способа организации управления уровнем услуг должны быть разработаны математические модели и алгоритмы кластеризации приложений, распределения и перераспределения ресурсов и управления потоками в ТКС. Решению этих вопросов посвящены главы 7, 8 и 9.

ВЫВОДЫ К ГЛАВЕ 3

1. Предложен декомпозиционно-компенсационный способ организации управления уровнем услуг в корпоративных ИТ-инфраструктурах. Способ отличается тем, что управление уровнем услуг осуществляется взаимодействием трех иерархических процессов: согласования уровня услуг, планирования ресурсов и управления уровнем услуг. Это позволяет создать иерархию решений и поддерживать согласованный уровень услуг с учетом существующих ограничений, поднимаясь вверх по иерархии для выбора управления при невозможности реализации управления с полномочиями нижних уровней.

2. Определены и формализованы все процессы предлагаемого декомпозиционно-компенсационного способа организации управления уровнем услуг с учетом целей управления на каждом уровне, основных показателей, оказывающих влияние на качество услуг, ресурсных ограничений, что позволяет создавать разноплановые модели управления уровнем услуг в корпоративных ИТ-инфраструктурах.

3. Сформулированы задачи управления ИТ-инфраструктурой при дефиците и избытке ресурсов ИТ-инфраструктуры, что позволило определить варианты реализации процессов управления, позволяющие оптимизировать использование ресурсов при управлении уровнем услуг.

4. Доказана целесообразность представления модели СУИ при управлении уровнем услуг в виде двухуровневой системы с координатором. Комбинированный принцип управления на основе обратной связи с учетом возмущающего воздействия и основанный на правилах принцип координации позволяют поддерживать согласованный уровень услуг в автоматическом режиме работы СУИ.

5. Показано, что монотонность функций, выражающих зависимость качества услуг от объема выделенных ресурсов, дает возможность применить эффективную итеративную процедуру управления.

6. Рассмотрены особенности реализации декомпозиционно-компенсационного способа организации управления уровнем услуг при использовании сервисно-ресурсных моделей.

БГУИР

ГЛАВА 4

МЕТОДЫ МОНИТОРИНГА ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ

Для формулирования требований к СУИ и определения принципов ее построения необходимо провести анализ объектов мониторинга и управления (ОМУ), которыми в ИТ-инфраструктуре служат элементы, представленные на обобщенной схеме корпоративной ИТ-инфраструктуры (см. рис. 1.2). Поэтому данная глава посвящена разработке концепции логического объекта мониторинга и управления и анализу методов мониторинга элементов ИТ-инфраструктуры.

4.1. РЕАЛЬНЫЕ И ЛОГИЧЕСКИЕ ОБЪЕКТЫ УПРАВЛЕНИЯ В ИТ-ИНФРАСТРУКТУРЕ

4.1.1. Концепция логического объекта мониторинга и управления

Предлагаемая концепция управления ИТ-инфраструктурой успешно и эффективно реализована в ИТ УИТИ благодаря тому, что процессы МАУОП оперируют с универсальными ОМУ. В качестве таких ОМУ использованы логические ОМУ [120].

Логические ОМУ (ЛОУ) – универсальное и обобщенное понятие, элемент логической модели ИТ-инфраструктуры, основная функциональная единица представления элементов и ресурсов ИТ-инфраструктуры, в которой инкапсулированы те свойства и состояния элементов ИТ-инфраструктуры, которые имеют отношение к процессам управления ИТ-инфраструктурой. Таким образом, ЛОУ – это абстрактное представление ресурсов и элементов ИТ-инфраструктуры, которые проявляют свои свойства с точки зрения управления ИТ-инфраструктурой.

ЛОУ специфицируется в понятиях параметров, атрибутов и их принадлежности, выполняемых над ними действий, выдаваемых ими сообщений, их взаимосвязей и взаимоотношений с другими ЛОУ [124, 120]. Относительно ЛОУ определяются набор функций и параметры мониторинга, функции и команды управления, методы оценки состояния и механизмы выявления неисправностей.

ЛОУ могут быть взаимосвязаны и ассоциируются не только с РОУ, но и с функциями, состояниями процессов деятельности, системами и подсистемами ИУС, процессами деятельности, бизнес-процессами и другими компонентами бизнес среды, а также со взаимосвязями в ИТ-инфраструктуре, физическими соединениями и людьми – администраторами и другим персоналом, принимающими участие в управлении ИТ-инфраструктурой. В этом случае ЛОУ является моделью РОУ, отражающей те его свойства, которые относятся к управлению ИТ-инфраструктурой. Также в качестве ЛОУ могут выступать абстрактные объекты, не имеющие РОУ, введенные в информационную модель, например, для обобщения информации мониторинга нескольких ЛОУ.

Модель ИТ-инфраструктуры имеет иерархическую строение, поэтому с точки зрения процессов МАУОП ИТ-инфраструктуру и связи ЛОУ удобно представлять в виде иерархического графа или совокупности графов, узлами которых являются ЛОУ, а связи отображают функциональные и информационные зависимости. При этом иерархических уровней ЛОУ может быть гораздо больше, чем уровней ИТ-инфраструктуры, и каждому из уровней модели ИТ-инфраструктуры соответствует несколько уровней иерархии ЛОУ. Каждый ЛОУ содержит сведения о конфигурации и текущем состоянии отдельного РОУ. Свойства, наиболее значимые с точки зрения мониторинга и оценки элементов ИТ-инфраструктуры, составляют набор параметров ЛОУ.

Каждый элемент ИТ-инфраструктуры содержит собственное дерево ЛОУ, листья которого – сенсорная часть в виде различных датчиков, а остальные вершины – родительские ЛОУ. Пример дерева зависимостей ЛОУ приведен на рис. 4.1.

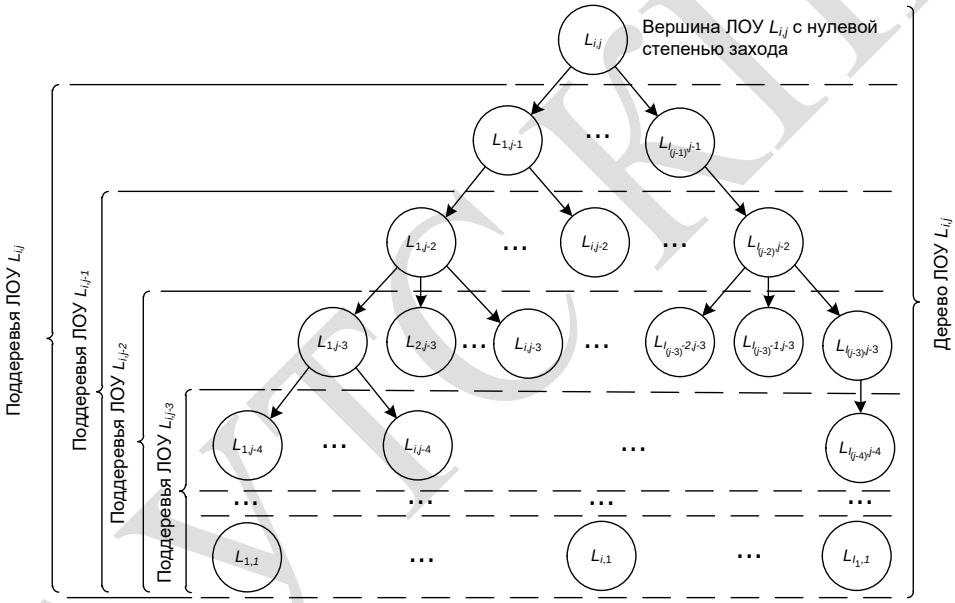


Рис. 4.1. Пример дерева зависимостей между ЛОУ с корнем $L_{i,j}$

Только одна вершина ориентированного графа i-го, $i = \overline{1, I_1}$, ЛОУ $L_{i,j}$, j-го иерархического уровня, ассоциированного с РОУ $R_{i,j}$, имеет нулевую степень захода. Все дочерние ЛОУ дерева ЛОУ $L_{i,j}$ являются самостоятельными объектами, которые могут быть родительскими для дочерних ЛОУ.

Представление элементов ИТ-инфраструктуре в виде ЛОУ основывается на следующих принципах:

- любой составной элемент ИТ-инфраструктуры можно представить в виде ЛОУ;
- ЛОУ инкапсулирует основные свойства, характеристики и параметры этого элемента с точки зрения процессов МАУОП;
- для ЛОУ определяются набор функций мониторинга и управления, методы оценки состояния;

• ЛОУ – единый объект, состоящий из множества связанных друг с другом элементов, представляющих собой определенное целостное образование;

• ЛОУ нижнего уровня является составной частью ЛОУ смежного расположенного выше уровня;

• ЛОУ имеет набор функций и параметров, обеспечивающих его связь с РОУ, свойства которого он инкапсулирует;

• все ЛОУ имеют единый интерфейс мониторинга и управления;

• добавление нового элемента в ИУС соответствует добавлению нового ЛОУ;

• функционирование подсистем ИТ-инфраструктуры представляется функционированием и взаимодействием составляющих их ЛОУ.

Такой подход позволяет использовать общие универсальные методы работы СУИ с ЛОУ, а также единый метод или несколько универсальных методов для оценки состояния ЛОУ.

Кроме того, использование ЛОУ позволит расширить сферу применения парадигмы объектно-ориентированного управления, которое в настоящее время часто применяется для управления сетями, на управление ИТ-инфраструктуры. При этом под объектно-ориентированным управлением ИТ-инфраструктуры понимается управление, основанное на представлении ИТ-инфраструктуры в виде множества взаимосвязанных и взаимодействующих адресуемых и управляемых ОМУ, которые характеризуются: частотой опроса, пороговыми значениями параметров, совокупностями событий, вызывающих сигнальные сообщения.

4.1.2. Механизм взаимосвязи РОУ и ЛОУ

РОУ могут иметь любую структуру, отличную от внутренних структур данных ИТ УИТИ. ЛОУ – это объект данных СУИ, который может быть сконструирован и сконфигурирован администратором посредством клиентского инструментария ИТ УИТИ и сохранен в БД. ЛОУ фиксируют определенные части ИТ-инфраструктуры и позволяют прикладывать к ним управляющие воздействия. РОУ могут иметь любую структуру, отличную от внутренних структур данных СУИ. ЛОУ не является моделью в чистом виде, используемой только для моделирования процесса управления, скорее наоборот, ЛОУ сами по себе не позволяют проводить симуляцию реальных событий. ЛОУ является представлением РОУ внутри ИТ УИТИ, наподобие унифицированного интерфейса, с которым могут работать механизмы оценки состояний и управления, а также администраторы ИТ-инфраструктуры.

Для связи ЛОУ и РОУ целесообразно использовать предлагаемый механизм DALL-функций, основанный на рефлексии типов. Посредством DALL-функций проводится мониторинг параметров и состояния РОУ и осуществляется воздействие на них. DALL-функции являются передаточными звеньями при взаимодействии ЛОУ и РОУ. Путем вызова соответствующих DALL-функций мониторинга проводится синхронизация состояния ЛОУ с состоянием РОУ, а в случае оказания на ЛОУ управляющих воздействий они передаются РОУ путем вызова соответствующих управляющих DALL-функций.

Механизм DALL-функций предоставляет инструментарий для расширения функционала и возможностей ИТ УИТИ, а также интеграции СУИ с внешними СУ, что делает разрабатываемую СУИ универсальной и позволяет применять ее для решения широкого спектра задач управления ИТ-инфраструктурой и ИУС.

Кроме того, абстрактность DALL-функций по отношению к СУИ является их важным преимуществом, так как позволяет наращивать количество типов РОУ, с которыми может взаимодействовать СУИ, только добавлением новых DALL-функций и соответствующего конфигурирования системы, без внесения изменений в программную реализацию самой СУИ. Таким образом, посредством механизма DALL-функций можно осуществлять взаимодействие СУИ с практически любыми внешними ИТ-системами и объектами.

Формально DALL-функцию можно представить функцией, которая принимает кортеж от ЛОУ из n параметров и возвращает кортеж из m значений от РОУ:

$$f:(a_1, a_2, ..., a_n) \to (b_1, b_2, ..., b_m). \quad (4.1)$$

Следовательно, параметры, с которыми оперирует каждый ЛОУ, можно разделить на две группы: параметры $a_h, h=\overline{1,n}$, которые передаются в DALL-функции, и параметры $b_z, z=\overline{1,m}$, значения которых являются результатом выполнения DALL-функций.

DALL-функции не являются чистыми функциями, т. е. функциями, не обладающими побочными эффектами и всегда возвращающими одинаковый результат для одинаковых параметров, поскольку результат их работы зависит от времени выполнения и конкретного РОУ, с которым взаимодействует данная функция. Например, запустив функцию мониторинга, в один момент времени на разных серверах можно получить разные результаты. Поэтому выражение (4.1) принимает вид

$$f:(t, X_{i,j}, a_1, a_2, ..., a_n) \to (b_1, b_2, ..., b_m), \quad (4.2)$$

где t – текущий момент времени, $X_{i,j}$ – состояние i-го РОУ $R_{i,j}$ j-го уровня иерархии, для которого проводится вызов функции.

Данные параметры можно определить во время выполнения функции, но невозможно указать явно. На них можно повлиять только косвенно, указав РОУ, с которым функции следует взаимодействовать, а также указав время или частоту вызова функции.

DALL-функции используются не только для мониторинга состояния, но и оказания управляющих воздействий на РОУ $R_{i,j}$, поэтому в механизм DALL-функций заложена возможность создания побочных эффектов [132]. DALL-функция (4.2), выполняет функции управления:

$$f:(t, X_{i,j}, a_1, a_2, ..., a_n) \to (\Delta t, X'_{i,j}, b_1, b_2, ..., b_m), \quad (4.3)$$

где Δt – время выполнения функции, $X'_{i,j}$ – состояние РОУ $R_{i,j}$ после выполнения функции.

Каждая DALL-функция f_k из множества функций $\{f_k\}$ осуществляет взаимодействие с определенным множеством РОУ $\{R_l\}$, содержащим РОУ с единым шаблоном ОМУ, например «коммутаторы». DALL-функция выбирает конкретный РОУ из всего множества $\{R_l\}$, доступных агенту A $(R_{i,j}, R_{i,j} \in \{R_l\}, R_{i,j} \in A)$ на основании идентификаторов РОУ $R_{i,j}$. При выборе используются определенные критерии, задаваемые разработчиками конкретной DALL-функции. Кроме

того, DALL-функция может одновременно взаимодействовать с несколькими разными РОУ. Каждый ЛОУ посредством одной или нескольких DALL-функций может взаимодействовать с одним или несколькими РОУ, либо с другими ЛОУ.

Входные параметры $a_h,\ h=\overline{1,n}$, DALL-функции делятся на конфигурационные $c_l,\ l=\overline{1,k}$, и управляющие $d_p,\ p=\overline{1,v}$. С помощью конфигурационных параметров $c_l,\ l=\overline{1,k}$, задается привязка ЛОУ к РОУ. Управляющие параметры связаны с управляющими воздействиями. Таким образом, зафиксировав изложенное выше,

$$\begin{cases} \{c_1,...,c_k\} \subseteq \{a_1,...,a_n\}, \\ \{d_1,...,d_v\} \subseteq \{a_1,...,a_n\}, \\ \{c_1,...,c_k\} \bigcup \{d_1,...,d_v\} = \{a_1,...,a_n\}, \\ \{c_1,...,c_k\} \bigcap \{d_1,...,d_v\} = \varnothing, \\ k+v=n, \end{cases} \tag{4.4}$$

получим следующее описание DALL-функции:

$$f:(t,X,c_1,...,c_k,d_1,...,d_v)=(\Delta t,X',b_i,...,b_m). \tag{4.5}$$

Если РОУ является сетевой картой сервера, а задача СУИ заключается в ограничении генерируемого трафика при возникновении перегрузок в сети, то при взаимодействии с РОУ соответствующий ЛОУ будет использовать две DALL-функции: мониторинга – f_1 и управления – f_2. Поскольку DALL-функции f_1 и f_2 могут использоваться для взаимодействия с множеством серверов, маршрутизаторов и ПК, для идентификации конкретного РОУ в качестве входного конфигурационного параметра c_1 DALL-функций f_1 и f_2 будем использовать IP-адрес сервера. Функция f_1 возвращает значение текущей установленной скорости передачи данных через сетевую карту (b_1), а функция f_2 принимает максимальную скорость обмена данными (d_1), и, соответственно, изменяет состояние сетевой карты сервера на (X'). Таким образом, параметры ЛОУ:

$$\begin{cases} c_1 - \text{IP-адрес}, \\ b_1 - \text{текущие ограничения по скорости передачи данных}, \\ d_1 - \text{новые ограничения по скорости передачи данных}. \end{cases} \tag{4.6}$$

DALL-функции ЛОУ с учетом параметров (4.6) имеют вид

$$\begin{aligned} &f_1:(t,X,c_1)\to(\Delta t,b_1), \\ &f_2:(t,X,c_1,d_1)\to(\Delta t,X'). \end{aligned} \tag{4.7}$$

Основное преимущество DALL-функций – они предоставляют открытую модульную архитектуру, позволяющую писать функционал управления новыми РОУ без внесения изменений в ИТ УИТИ.

Архив КПИ

4.1.3. Классификация объектов мониторинга и управления

Особенности классификации ОМУ. Современные ИТ-инфраструктуры отличаются большой сложностью и содержат десятки, а то и сотни тысяч разнообразных аппаратных и программных элементов. Учитывая отсутствие наработанных универсальных методов классификации и правил выделения классов и объектов [15], задачу идентификации классов и объектов ИТ-инфраструктуры можно полагать одной из самых сложных задач разработки СУИ. Отсутствует универсальный критерий классификации, а одна и та же структура классов может подходить для одной задачи управления ИТ-системой и быть совершенно не пригодной для другой задачи управления.

Поскольку СУИ призвана эффективно решить множество разнообразных задач управления ИТ-инфраструктурой, наряду с комплексом моделей управления необходимо разработать универсальную классификацию ОМУ, учитывающую весь спектр признаков, связанных с управлением. Так, для решения задач управления доступом к ограниченным ресурсам ИТ-инфраструктуры идентификацию классов и объектов целесообразно проводить на основе одинакового поведения, выделяя однотипные ресурсы. При решении задач управления сетевым трафиком идентификацию классов и объектов можно выполнять на основании того, являются ли объекты генераторами трафика или они осуществляют его перераспределение. При этом не важно какой объект порождает трафик – сервер или рабочая станция, функциональная или технологическая подсистема, поскольку при решении задачи управления трафиком учитывается его важность, определяемая приоритетом или значимостью распределенного приложения, которое его генерирует, для решения бизнес-задач.

Выделение ОМУ, а также схем и структур взаимодействия объектов, обеспечивающих выполнение решаемых задач, является важным этапом проектирования таких больших и сложных систем, как СУИ. Кроме структуры, описывающей отношения между физическими элементами ИТ-инфраструктуры, необходимо создать иерархию процессов, описывающую отношения между динамическими элементами. Идентификация классов и объектов ИТ-инфраструктуры носит произвольный характер, осуществляется администраторами ИТ-инфраструктуры [346] или проводится разработчиками СУИ при декомпозиции ИТ-инфраструктуры [345] с учетом следующих характерных признаков ИТ-инфраструктуры и ИУС [15, 219]:

- ИТ-инфраструктура является иерархической и состоит из взаимосвязанных систем, которые, в свою очередь, могут быть разделены на подсистемы и т. д.;
- обычно внутрикомпонентная связь сильнее, чем связь между компонентами, а различие внутрикомпонентных и межкомпонентных взаимодействий обуславливает возможность разделения функций между частями ИТ-инфраструктуры и позволяет относительно изолированно изучать каждую из частей;
- иерархические системы могут быть представлены комбинацией небольшого числа по-разному скомбинированных и организованных типов подсистем.

Идентификация классов и классификация ОМУ должны проводиться на основании признаков, определяемых сформулированной целью управления. При определении элементов системы, которые считаются элементарными с точки зрения СУИ, следует использовать критерий разумной достаточности, поскольку при управлении функциональными и технологическим подсистемами нет необходимости осуществлять мониторинг работоспособности, например, отдельных

микросхем компьютеров или подпрограмм ПО. Выбор класса или объекта в качестве элементарного зависит только от рассматриваемой задачи управления ИТ-инфраструктурой.

Все ОМУ могут быть сгруппированы по принадлежности к отдельному структурному компоненту ИТ-инфраструктуры, объединены по функциональному признаку и разделены на уровни по иерархическому принципу. При этом может выделяться различное количество уровней и использоваться множество принципов объединения в зависимости от характера решаемых задачи управления, специфики ИТ-инфраструктуры или ИУС. Иерархия ОМУ может быть привязана к уровням ИТ-инфраструктуры или накладываемой на ИТ-инфраструктуру структуры, например ИУС, которую поддерживает ИТ-инфраструктура, с выделением различных, необходимых для целей управления уровней, например, может быть введен уровень ПТК.

Принципы классификации ОМУ. При классификации ОМУ сначала создаются концептуальные описания классов, после чего проводится классификация ОМУ в соответствии с описаниями. Выделение классов и объектов является важным этапом составления словаря предметной области управления ИТ-инфраструктурой. Идентификация классов и ОМУ проводится по свойствам, важным для решения задач управления в СУИ, с учетом иерархических уровней ИТ-инфраструктуры или ИУС. При этом один тот же объект может принадлежать к нескольким категориям одновременно.

ОМУ ИТ-инфраструктуры можно классифицировать по следующим признакам:

- в соответствие с иерархией элементов ИТ-инфраструктуры или ИУС;
- по характеру выполняемых операций для определения состояния ОМУ;
- по выполняемым функциям и пр.

ОМУ в соответствие с иерархией элементов ИТ-инфраструктуры можно разделить на:

- бизнес-процессы или процессы деятельности;
- операции бизнес-процессов;
- функциональные и технологические подсистемы;
- ПТК, КСА и пр.;
- АРМ, хранилища данных и пр.,
- аппаратное и программное обеспечение ИТ-инфраструктуры, сервисы, службы и пр.;
- устройства ИТ-инфраструктуры (серверы, рабочие станции, активное коммуникационное оборудование и пр.);
- элементы серверов, рабочих станций и коммуникационного оборудования.

По характеру выполняемых операций для определения состояния можно разделить ОМУ на объекты, для определения состояния которых требуется выполнение операций (см. рис. 4.2):

- сбора и обработки данных,
- сбора, обработки и анализа информации;
- только анализа информации.

Архив

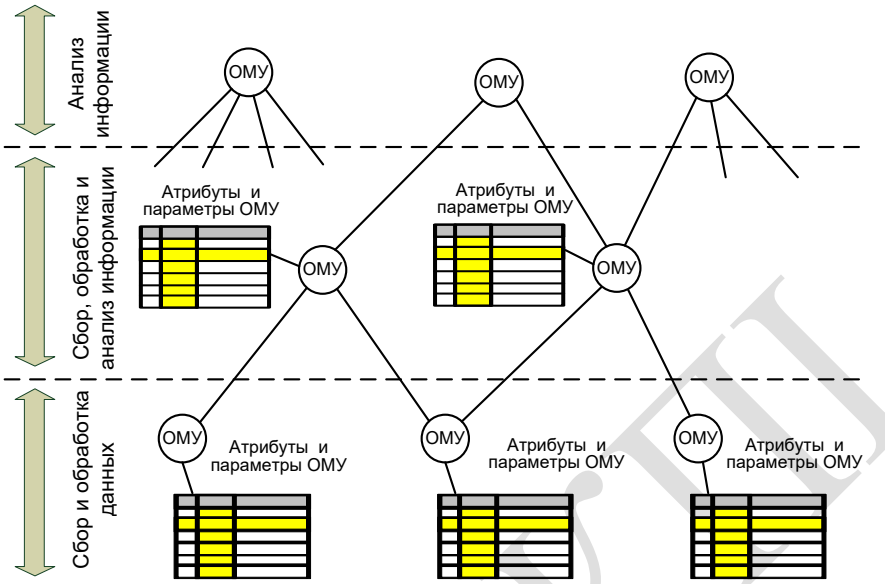


Рис. 4.2. Классификация ОМУ по характеру выполняемых в СУИ операций

На самом нижнем уровне иерархии находятся ОМУ, которые не включают в себя другие ОМУ, и состояние которых определяется только значениями их параметров (рис. 4.3). Это же может быть справедливо и для некоторых объектов расположенных выше уровней.

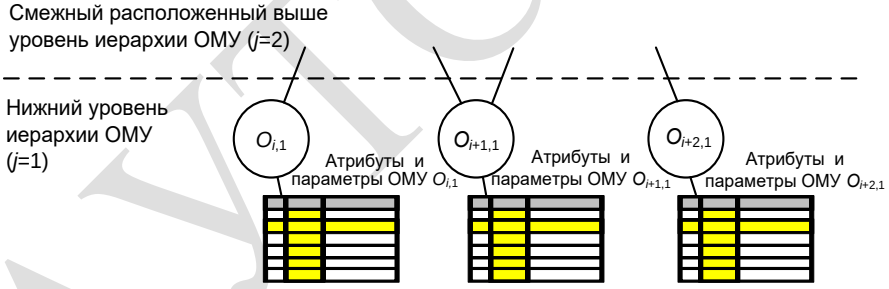


Рис. 4.3. Состояние ОМУ нижнего уровня определяется только значениями его собственных параметров

Для определения состояний ОМУ $O_{i,1}$, $i=\overline{1,I_1}$, самого нижнего уровня ИТ-инфраструктуры выполняются операции сбора и обработки данных. В этом случае состояние ОМУ определяется обработкой значений параметров, непосредственно полученных в результате мониторинга. Для этого СУИ должна постоянно получать непосредственно от ОМУ $O_{i,1}$, $i=\overline{1,I_1}$, значения параметров и после обработки значений параметров определять состояние ОМУ. К ОМУ этого уровня относятся элементы устройств ИТ-инфраструктуры, например ОЗУ, процессоры, порты и прочие, а также компоненты ПО.

На расположенных выше уровнях для определения состояния ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, и $j=\overline{1,J}$, или оценки качества функционирования подсистем, наряду с

операциями сбора и обработки данных о значениях параметров, характеризующих состояние и функционирование этих ОМУ, используются операции анализа, когда для определения их состояния СУИ осуществляет анализ обработанных значений параметров функционирования ОМУ, не зависящих от работы включаемых ОМУ, в совокупности с анализом состояний входящих в него ОМУ. К таким ОМУ относятся серверы, рабочие станции, активное сетевое оборудование и пр.

На верхних уровнях иерархии, где в состав ОМУ входят только другие ОМУ, для определения состояния ОМУ в СУИ осуществляется только анализ информации о состоянии ОМУ, входящих в состав рассматриваемого ОМУ. К ОМУ такого типа относятся функциональные и технологические подсистемы, ТПК и аналогичные компоненты.

Также в качестве ОМУ могут выступать абстрактные объекты, не имеющие РОУ. Такие объекты могут обобщать в себе информацию мониторинга нескольких ОМУ.

Представление структуры ИТ-инфраструктуры на основе ОМУ. Модель ИТ-инфраструктуры в виде совокупности ОМУ можно представить графом, подобным приведенному на рис. 4.1. Вершины этого графа соответствуют ОМУ, обладающим информацией о текущем состоянии и конфигурации соответствующих элементов ИТ-инфраструктуры, а ребра графа – функциональным связям между элементами ИТ-инфраструктуры.

Такой граф является неориентированным, а его вершины, разбитые в соответствии со структурой ИТ-инфраструктуры, создают иерархические уровни наподобие ярусно-параллельных форм (рис. 4.4).

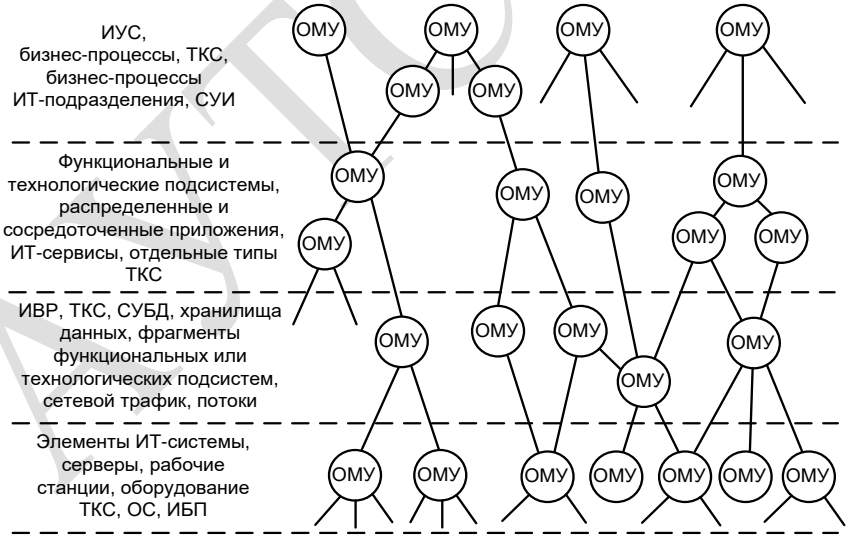


Рис. 4.4. Иерархический граф ОМУ

Иерархия ОМУ, приведенная на рис. 4.4, отличается от иерархии ИТ-инфраструктуры, приведенной на рис. 1.2 и используемой для разработки ИТ УИТИ. Иерархия ОМУ может быть использована, например, менеджерами ИУС при оценке влияния неполадок в ИТ-инфраструктуре на эффективность выполнения бизнес-процессов. При этом каждый из уровней при необходимости может быть

разбит на несколько подуровней. Каждый ОМУ является частью ИТ-инфраструктуры, которая помимо вербальной характеристики может быть описана набором параметров. На нижнем уровне ОМУ находятся аппаратные и программные элементы ИТ-инфраструктуры, на верхнем – функциональные и технологические подсистемы, выполняющие возложенные на ИУС задачи.

Каждый ОМУ содержит сведения о конфигурации и текущем состоянии отдельного РОУ. Так, для компьютеров это процессорная емкость, использование ОЗУ, своп-файла, жестких носителей, текущая скорость считывания/записи контроллера жесткого диска и пр., для веб-серверов – пропускная способность, количество удачных соединений между клиентами и веб-сервером в секунду, среднее время отклика веб-сервера, относительное быстродействие интерфейса программирования приложений, количество ошибок в секунду и др., для единиц ПО – количество строк исходного кода, циклическая сложность, количество методов в классе, глубина наследственности и др. Выделение ОМУ в процессе декомпозиции ИТ-инфраструктуры состоит во введении и формальном описании логических сущностей в соответствии с весомостью функциональных единиц и формировании наборов свойств, определяющих их состояние.

ОМУ могут влиять на несколько различных ОМУ высшего уровня иерархии. Например, ОМУ «Сервер СЕДО» может входить в ОМУ «СЕРВЕРЫ» и в ОМУ «ПТК СЕДО». При необходимости для анализа состояния ОМУ в нем могут быть выделены новые ОМУ или введены для рассмотрения новые параметры. Например, в ОМУ «Аппаратные средства сервера СЕДО» может быть добавлен ОМУ «ОЗУ сервера СЕДО» или введен параметр «Средняя загрузка ОЗУ сервера СЕДО», значения которого будут учитываться при анализе состояния ОМУ «Аппаратные средства сервера СЕДО» и ОМУ следующего уровня иерархии – «Сервер СЕДО».

ОМУ могут представлять как реальные объекты, такие как персональные компьютеры и серверы, так и абстрактные сущности, такие как приложения и сервисы, запущенные на каком-то компьютере или сразу на нескольких серверах. Сюда же можно отнести бизнес-процессы, которые зависят от работоспособности сразу нескольких распределенных сервисов.

Для решения задач МАУОП необходимо получать и обрабатывать значения параметров, характеризующих состояние и показатели функционирования *i*-го ОМУ $O_{i,j}$, *j*-го иерархического уровня или подуровня. При этом одни и те же параметры могут использоваться для решения разнообразных задач управления, как это показано на рис. 4.5.

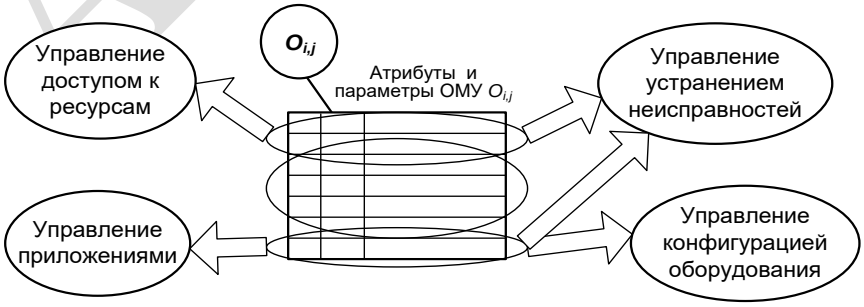


Рис. 4.5. Использование значений одних и тех же параметров ОМУ для решения задач различных категорий управления

Каждый ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, и $j=\overline{1,J}$, описывается совокупностью параметров или атрибутов, а его состояние определяется значениями отдельных параметров. При этом ОМУ могут, как показано на рис. 4.6, включать или не включать в себя другие ОМУ. ОМУ самого нижнего уровня иерархии описывается только атрибутами и параметрами, значения которых анализируются СУИ для определения состояния ОМУ или оценки качества его функционирования.

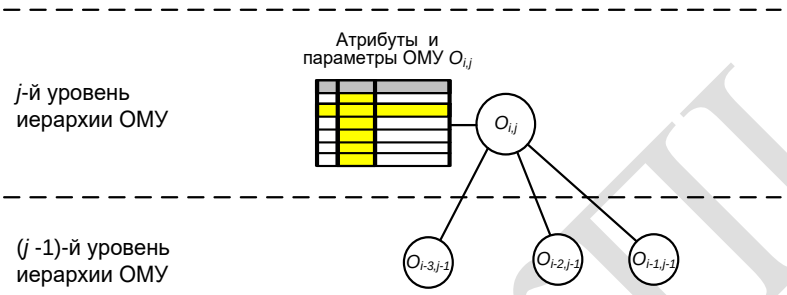


Рис. 4.6. ОМУ *j*-го уровня иерархии включает в себя ОМУ нижних уровней и имеет собственные атрибуты и параметры

Для определения состояния ОМУ *j*-го иерархического уровня деления ОМУ СУИ необходимо получить информацию о значениях параметров рассматриваемого ОМУ, а также обобщенную информацию об ОМУ (*j*-1)-го иерархического уровня, входящих в состав рассматриваемого ОМУ *j*-го уровня (см. рис. 4.6).

Как отмечалось, целесообразно рассматривать ОМУ двух типов – РОУ и ЛОУ. Первые соответствуют реальным элементам ИТ-инфраструктуры, а вторые универсализуют абстрактные понятия, в которых воплощены свойства и состояния всех составляющих ИТ-инфраструктуры, попадающих в сферу интересов СУИ. Каждый ЛОУ характеризуется состоянием, поведением, идентичностью и является моделью РОУ или его составной части, отражающей такие свойства, которые имеют отношение к управлению ИТ-инфраструктурой. ЛОУ являются элементами логической модели ИТ-инфраструктуры и им могут соответствовать не только такие реальные элементы ИТ-инфраструктуры, как сетевые узлы, серверы или рабочие станции, но и любая часть сервера, а также функциональные или технологические системы или любые их составляющие. Таким образом, с точки зрения СУИ ЛОУ может быть все, с чем СУИ может оперировать, по отношению к чему могут быть применены процессы МАУОП или часть этих процессов.

Параметры ОМУ можно поделить на конфигурационные, информационные и контролируемые.

Конфигурационные параметры – динамически неизменяемые параметры, характеризующие конфигурацию элементов и их технические характеристики. Эти параметры задаются администратором. Также могут проводиться автоматическое конфигурирование и мониторинг данных параметров.

Информационные параметры – параметры, мониторинг которых проводится и должен быть предоставлен в выходных данных. Этот тип параметров используется при анализе.

Контролируемые параметры – параметры, которые косвенным образом влияют на состояние ОМУ и на основе которых проводится управление для приведения этих параметров в норму.

При разработке ИТ УИТИ можно воспользоваться другой классификацией, которая предполагает деление параметров ОМУ на *измеряемые*, *вычисляемые* и *конфигурируемые*.

Параметры могут быть *численными* и *логическими*. Пример численного параметра – объем свободного пространства на жестком диске, логического – параметр, фиксирующий наличие определенного приложения в памяти сервера. В целях унификации логические параметры приводятся к численным по следующему принципу – логическому значению «истина» присваивается числовое значение 1, а значению «ложь» – 0.

Значения *измеряемых* и *конфигурируемых параметров* можно получить с помощью встроенных диагностических средств ОС, например, счетчиков производительности и WMI, посредством протоколов управления – SNMP, CMIP и других или тестового ПО, возвращающего численные или логические значения параметров, характеризующих отдельный аспект поведения ОМУ. Например, тестовое ПО, генерирующее серию тестовых запросов к СУБД и возвращающее суммарное или среднее время их обработки.

Конфигурируемые параметры обычно остаются неизменными во время работы ОМУ. Новые значения они приобретают в случае переконфигурации элемента, например, после установки дополнительной планки ОЗУ увеличивается объем оперативной памяти. Конфигурируемые параметры используются при определении значений вычисляемых параметров, а самостоятельного влияния на качество функционирования элемента не оказывают. Так, размер оперативной памяти сервера используется для определения доли свободной оперативной памяти сервера в процессе его функционирования.

Значения *вычисляемых параметров* определяются аналитически на основании значений одного или нескольких других параметров. Вычисляемые параметры являются простыми, если их значение вычисляется на основании значений одного параметра, и сложными, если они вычисляются на основе значений нескольких параметров. Значения вычисляемых параметров могут быть получены в результате выполнения простых (нормирование, вычисление среднего) и достаточно сложных математических операций (например, корреляция значений нескольких параметров).

К простым вычисляемым параметрам относятся:

- среднее значение параметра за период T_i. Период для каждого параметра устанавливается индивидуально в зависимости от его свойств во временной области и может изменяться администратором СУИ;
- нормированное значение параметра – приведение значений параметра к диапазону [0, 1];
- среднее нормированное значение параметра за период T_i.

Нормированные значения параметров удобны при применении библиотеки унифицированных функциональных зависимостей для определения оценочного коэффициента и коэффициента критичности параметра.

Для большинства параметров использование мгновенного значения при анализе состояния не имеет смысла. Так, мгновенное значение параметра «загруженность процессора» не несет достоверной информации о реальной загруженности процессора. Этот параметр в момент загрузки приложения в память принимает максимальное значение, а пики в загруженности процессора – нормальное явле-

ние, не являющееся сигналом о проблемах с производительностью вычислительной системы. В таких случаях для вычисления показателей качества функционирования ОМУ целесообразно использовать среднее за период значение.

Значения для вычислений параметров измеряются через фиксированный интервал времени, который задается индивидуально для каждого параметра и может изменяться администратором СУИ. Для упрощения работы планировщика СУИ интервалы опроса параметров могут группироваться и выбираться из фиксированного ряда. Кроме того, планировщик выполняет сбор значений параметров так, чтобы равномерно распределить нагрузку на каналы связи.

4.1.4. Реальные объекты мониторинга и управления

Понятие РОУ. Для подавляющего большинства существующих СУИ объектами мониторинга и управления являются РОУ, которыми обычно являются аппаратно-программные средства ИТ-инфраструктуры [44, 236, 267, 334, 357, 356, 358, 360, 376, 389]. Как правило, взаимодействие с РОУ осуществляется посредством стандартных или специализированных аппаратно-программных агентов, устанавливаемых на РОУ или встроенных в ОС [44, 267, 296, 360]. Стандартные агенты, являясь агентами протоколов SNMP, CMIP, RMON и других, собирают информацию о состоянии РОУ, событиях, происходящих в РОУ, или характеристиках сетевого трафика и передают эту информацию в СУИ.

РОУ отражают суть реальных объектов относительно управления ИТ-инфраструктурой и являются моделью реальных физических и программных объектов ИТ-инфраструктуры, которыми могут быть аппаратные части элементов ИТ-инфраструктуры, программные сервисы, персонал ИТ-подразделения и т. п. Работа СУИ с РОУ должна обеспечивать достаточную точность для проведения управления и анализа состояния объектов реального мира с бесконечным, непрерывным, многомерным пространством состояний.

ИТ-инфраструктура представляет собой нечеткое множество РОУ, поскольку между реальными объектами не всегда явное разграничение, например, с одной стороны, сервер приложений является частью физического сервера и размещается в его памяти, а с другой, является частью веб-сервиса.

СУИ оперирует с деревом ЛОУ, за которым стоит размытая структура РОУ.

В настоящее время не существует универсальной классификации РОУ [217, 235, 345], а лучшим признанным способом классификации является интеллектуальный, последовательный итеративный процесс [15].

Часто для РОУ не целесообразно вводить какую-то иерархию, их индексируют или осуществляют кодификацию, определяют вложенность иным способом, например с учетом базового элемента.

Далее к РОУ будем относить физические объекты, такие как серверное и сетевое оборудование, рабочие станции и их составляющие, и логические – сервисы, процессы, серверы БД.

Классификация РОУ. Классификация РОУ должна выполняться на основании решаемой задачи управления. Так, для мониторинга и анализа функционирования ИТ-инфраструктуры с целью подержания параметров совместного функционировании технологических и функциональных подсистем ИУС в соответствии с заданным регламентом необходимо выделить РОУ, оказывающие влияние на работу этих подсистем. Затем следует определить взаимосвязи и степень влияния

РОУ на функционирование подсистем ИУС. Далее необходимо отнести РОУ к одному из уровней иерархии ИУС. Причем на нижних двух уровнях ИУС (см. рис. 1.2) целесообразно осуществлять классификацию РОУ, приняв за основу цель функционирования, место в ИТ-инфраструктуре и совокупность услуг, предоставляемых в соответствии с обязательствами и ответственностью РОУ.

Сгруппировав объекты I-го и II-го уровней ИТ-инфраструктуры, имеющие сходные ответственности, получим следующие три класса объектов:

- «Серверы» – включает в себя всех поставщиков информационных и телекоммуникационных услуг, служб и сервисов. Объекты этого класса реализуют главную целевую функцию ИТ-инфраструктуры – обработку информации и определяют основные информационно-вычислительные и телекоммуникационные ресурсы, предоставляемые пользователям. Этот класс включает в себя все аппаратные и программные серверы, кластеры, а также рабочие станции, которые наряду с доступом пользователей к информационным ресурсам, выполняют функции поставщика различных услуг;
- «Терминалы» – включает в себя все устройства потребителей ИТ-услуг. В этот класс входят все рабочие станции, предоставляющие пользователям доступ к информационным и телекоммуникационным ресурсам и осуществляющие получение результатов доступа к ресурсам в удобном для пользователя виде;
- «Узлы» – обеспечивает взаимодействие серверов с терминалами, серверов с серверами, а также терминалов между собой на основе процедур обмена данными. В этот класс входят все телекоммуникационное оборудование: коммутаторы, маршрутизаторы, каналы связи и другие элементы телекоммуникационной сети.

Для группы этих классов суперклассом будет «Устройства ИТ-инфраструктуры» (рис. 4.7).

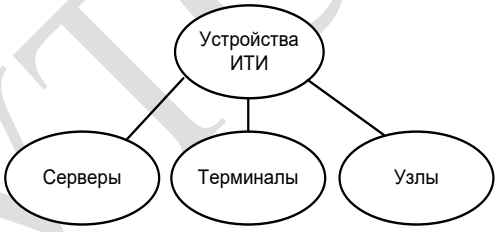


Рис. 4.7. Суперкласс «Устройства ИТ-инфраструктуры»

Наибольшую важность для работоспособности ИУС представляют серверы и узлы. Хотя терминалы также могут быть предметом пристального внимания со стороны СУИ. Особенно это касается специализированных ИТ-систем и ИУС, когда некоторые рабочие станции активно участвуют в работе функциональных подсистем ИУС, а также финансовых организаций с их банкоматами и рядом рабочих станций. Поэтому от СУИ может потребоваться обеспечение бесперебойной и корректной работы терминалов. Часто целесообразно, чтобы СУИ осуществляла контроль работоспособности не самих рабочих станций, а только ряда приложений пользователей.

Необходимо отметить, что работа СУИ с РОУ имеет множество недостатков, порождает большое количество проблем концептуального характера. Для устранения недостатков в данной работе в качестве ОМУ для СУИ использованы ЛОУ, что принципиально отличает предлагаемое решение СУИ от аналогов и прототипов.

4.1.5. Базовый уровень иерархии реальных объектов мониторинга и управления

Создать идеальную иерархию РОУ такой сложной многокомпонентной системе, какой является ИТ-инфраструктура, не представляется возможным, поскольку для этого требуется разработка моделей многочисленных сильно различающихся объектов с большим количеством свойств, причем поведение многих РОУ чрезвычайно сложно охарактеризовать.

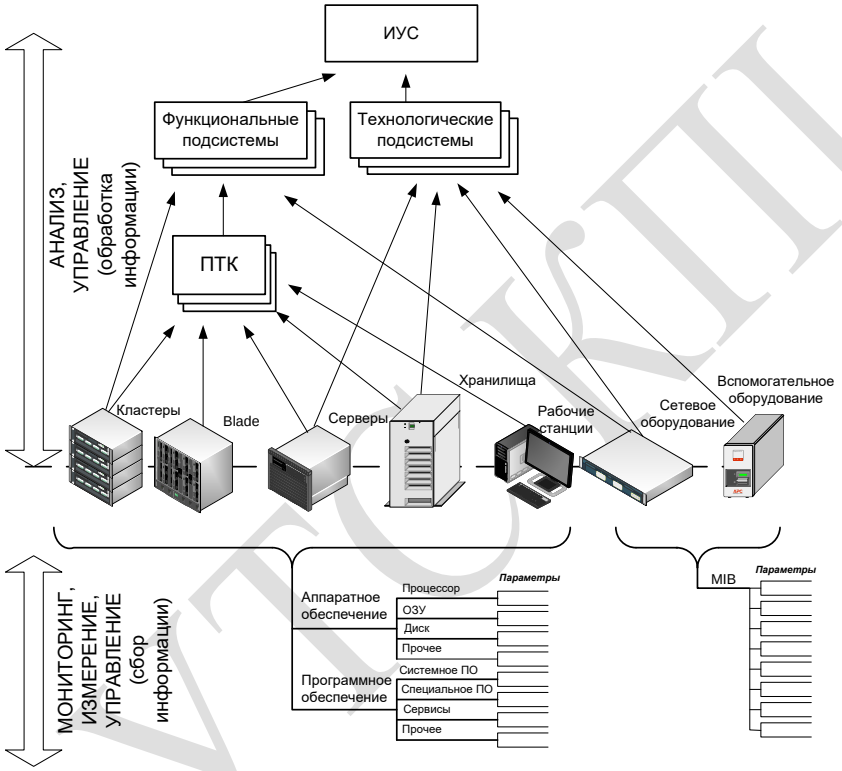


Рис. 4.8. Иерархия объектов ИТ-инфраструктуры с базовым уровнем устройств с IP-адресами

В настоящее время подавляющее большинство ИТ-инфраструктур в качестве транспортной среды для передачи данных используют IP-сеть, а по прогнозам ITU далее основным протоколом, применяемым в сетях, будет IP. Кроме того, взаимодействие элементов ИТ-инфраструктур, осуществляющих распределенную обработку информации, также в подавляющем большинстве случаев осуществляется с использованием протокола IP. Этот протокол работает на сетевом уровне модели OSI ISO [261] и использует для адресации устройств ИТ-инфраструктуры, подключенные к сети, IP-адреса. IP-адреса отождествляются с уникальными аппаратными MAC-адресами сетевых карт серверов, рабочих станций и других устройств, взаимодействующих через сеть, а также один или несколько MAC-адресов имеют активные сетевые устройства. Кроме того, происходящая в настоящее время конвергенция, объединяющая разнотипные устройства, различ-

ные сетевые технологии и сетевые сервисы стала возможной только из-за повсеместного применения IP-протокола. Благодаря этому происходит объединение трафиков систем охранной, пожарной сигнализации и множества других разнотипных систем, упрощается подключение к IP-сети различных датчиков, средств видеонаблюдения и пр.

Поэтому в качестве базового уровня иерархии элементов ИТ-инфраструктуры предлагается выбрать уровень элементов, имеющих собственный IP-адрес, т. е. уровень устройств (рис. 4.8). Кроме того, такой уровень целесообразно выбрать в качестве базового и по той причине, что при мониторинге функционирования ИУС в первую очередь проверяется доступность элементов ИТ-инфраструктуры в сети. Если устройство недоступно, то это означает, что оно не функционирует или не способно взаимодействовать через сеть.

В качестве базового уровня выбран уровень элементов ИТ-инфраструктуры, являющихся устройствами, имеющими IP-адрес, MAC-адрес и сетевое имя, еще и потому, что на этом уровне происходит объединение аппаратных и программных средств в едином устройстве (см. рис. 4.8). РОУ, из которых состоят IP-устройства, на самом нижнем уровне иерархии содержат РОУ, параметры которых можно измерить (сбор и обработка информации). На расположенных выше уровнях состояние РОУ получается в результате анализа состояния расположенных ниже РОУ и измерения собственных оригинальных параметров. На уровнях, лежащих выше уровня IP-устройств, определение состояния РОУ проводится анализом состояний РОУ, влияющих на работу рассматриваемого РОУ.

На иерархических уровнях РОУ осуществляются процедуры обработки данных, указанные на рис. 4.9.

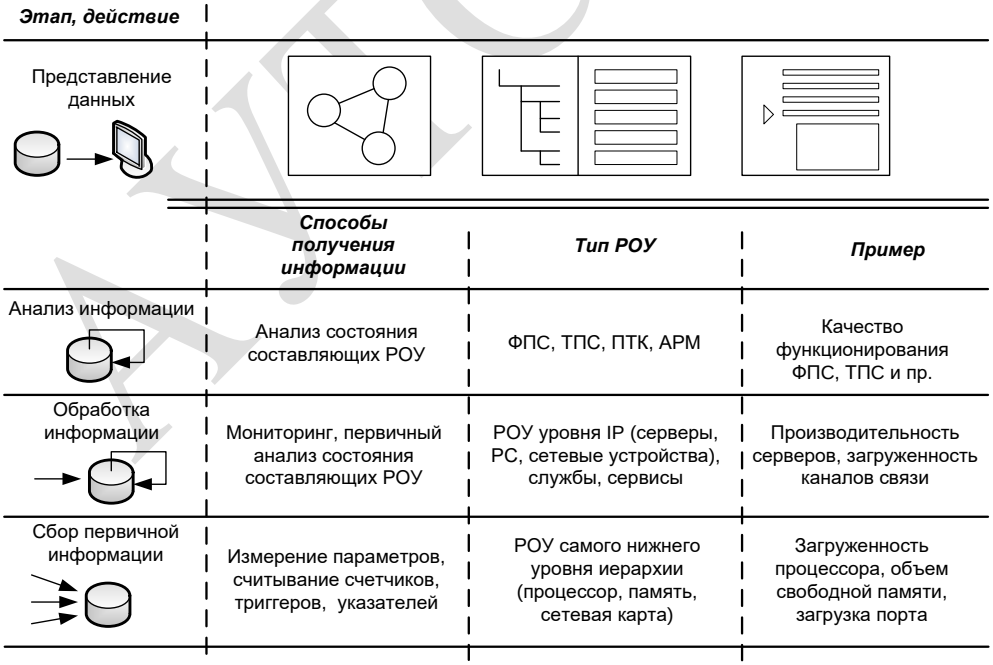


Рис. 4.9. Этапы обработки информации в СУИ

Для анализа работы или состояния, например, ФПС или ТПС, необходимо чтобы работали IP-устройства, функционировало общесистемное и специальное ПО и было обеспечено сетевое взаимодействие (рис. 4.10). При этом состояние каждой составляющей влияет на состояние ФПС или ТПС. Производительность IP-устройств и сетевых соединений оказывает влияние на производительность ФПС, а составляющая – общесистемное и специальное ПО – необходимое условие функционирования ФПС в принципе.

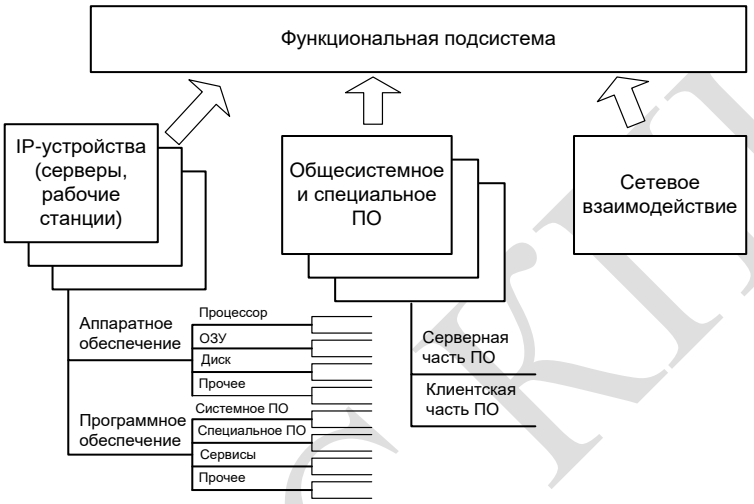


Рис. 4.10. Учет значений параметров ОМУ при анализе функционирования ФПС

4.2. МЕТОДЫ МОНИТОРИНГА ИТ-ИНФРАСТРУКТУРЫ

4.2.1. Анализ проблем и методов мониторинга ИТ-инфраструктуры

Современные ИУС направлены на повышение эффективности бизнес-процессов и представляют собой сложные гетерогенные структуры, в среде которых взаимодействуют разнообразные программные и аппаратные средства. Непрерывное предоставление доступа к распределенным ресурсам ИТ-инфраструктуры, эффективное распределение и закрепление ресурсов за бизнес-процессами, повышение доступности сетевых ресурсов и сервисов пользователям и процессам, поддержание высокой работоспособности и готовности ИТ-инфраструктуры обеспечивают СУИ [106, 107, 165]. Для решения этих и других задач, а также обеспечения надежного и качественного функционирования ИТ-инфраструктуры СУИ должны выполнять мониторинг, анализ функционирования, управление компонентами ИТ-инфраструктуры и другие функции.

Отказы оборудования, нарушения в работе ПО, а также неквалифицированные действия персонала могут привести к простоям сети, вычислительной техники и обусловленными этим потерями в деятельности предприятия. Поэтому на СУИ возлагается задача непрерывного мониторинга состояния ИУС, а в случае обнаружения неполадок, вызывающих невозможность нормальной работы всех бизнес-процессов, – обеспечение функционирования важнейших из них.

В настоящее время отсутствует определение или однозначное толкование термина «мониторинг ИТ-инфраструктуры», а в перспективных системах мониторинга ИТ-инфраструктуры осуществляется переход от поддержки работы оборудования к поддержке функционирования ИТ-услуг, предоставляемых ИТ-подразделениями бизнес-подразделениям.

Для контроля работоспособности коммуникационных сетей и ИУС в СУИ используются различные методы мониторинга, среди которых наибольшей популярностью пользуются статистический [223], реактивный [223] и проактивный мониторинги [315].

При *статистическом* мониторинге на основании анализа поступающих многочисленных исходных данных СУИ находит некоторые статистические закономерности, позволяющие предсказать тенденции в поведении ИТ-инфраструктуры. В этом случае все необработанные данные должны поступить в центральный сервер СУИ и возможности сокращения объема трафика мониторинга отсутствуют. Системы статистического мониторинга позволяют анализировать характеристики сетевого трафика и ОМУ, общую функциональность сети и ИУС. Они хорошо подходят для анализа текущего состояния, перспектив и тенденций поведения телекоммуникационных сетей и ИУС, но менее пригодны для обнаружения и локализации неисправностей [135].

При *реактивном* мониторинге СУИ получает информацию о состоянии ОМУ в реальном или псевдореальном времени. Это позволяет реагировать на некоторые аварийные ситуации, которые могут прогрессировать в ИТ-инфраструктуре. Аварийные ситуации обычно свидетельствуют о неисправности в ОМУ или означают аномальное поведение ОМУ, которое может привести к неисправности. Такой вид мониторинга дает множество возможностей для поиска механизмов и алгоритмов минимизации объема контрольной информации, передаваемой по сети [183, 186, 189, 198, 205, 218, 223, 250, 271, 343, 344, 381, 382]. Системы реактивного мониторинга позволяют определять только часть проблем в сложных ИТ-инфраструктурах, испытывая затруднения при анализе функционирования сложных распределенных приложений. При этом диагностика и локализация ошибок в ИТ-инфраструктуре проводятся после обнаружения неполадок, так же как определяются только проблемы, которые уже существуют в аппаратном или программном обеспечении.

Более совершенными являются системы *проактивного* (упреждающего) мониторинга [268, 296, 315]. Они не только обеспечивают дистанционный мониторинг в режиме реального времени и регулярные проверки исправности ОМУ, но и позволяют прогнозировать критические состояния ИТ-инфраструктуры. Таким образом появляется возможность на ранней стадии генерировать предупреждения об ошибках, для того чтобы предотвратить возникновение отказов в работе ИУС. Такой мониторинг позволяет анализировать не только загруженность процессоров, объем свободной памяти, доступность ОМУ, состояние сервисов, но и проконтролировать сложные взаимосвязи между сервисами, работоспособность распределенных многоуровневых приложений, а также получить другие преимущества. При этом, естественно, приложение должно уметь выявлять неполадки в своей работе и сигнализировать о них. Преимущества проактивного мониторинга по сравнению с реактивным базируются на знаниях об ИТ-инфраструктуре, понимании логики работы распределенных приложений, способности предсказывать на основе анализа накопленных данных возможные сценарии развития текущей ситуации. За счет этого проактивный мониторинг позволяет выявлять и

предсказать гораздо больше проблем, что дает возможность устранить неполадки еще на этапе их зарождения и развития.

С помощью проактивного мониторинга можно не только выявить конкретный некорректно работающий в данный момент ОМУ, но и предсказать возможность отказа этого ОМУ в будущем. За счет этого обеспечивается стабильная работа ИУС и минимизируются издержки из-за ее простоя. На основании анализа результатов прогнозирования СУ с проактивным мониторингом могут вырабатывать рекомендации, позволяющие оптимизировать выполнение регламентных работ, загрузку оборудования, высвобождение дискового пространства, резервного копирования и пр.

Недостатки СУ с проактивным мониторингом – относительная сложность в установке и настройке, высокая стоимость. Кроме того, такие СУ, как правило, являются проприетарными решениями, например, компании Microsoft, которые хорошо работают только в среде Windows и продуктами Microsoft [296].

При отсутствии или ограниченности возможностей применения проактивного мониторинга для мониторинга ОМУ используются методы реактивного мониторинга, позволяющие существенно сократить объем трафика контрольной информации. Применять только реактивный мониторинг не целесообразно, поскольку ИТ-инфраструктура характеризуется большим количеством разнообразных технологий, ресурсов и сервисов, для эффективного управления которыми не достаточно только информации об аварийных ситуациях. Поскольку в случае применения проактивного мониторинга проблемы решаются только после их возникновения, то это приводит к финансовым потерям из-за простоя автоматизированных систем поддержки бизнес-процессов.

Необходимо иметь информацию, полученную в результате статистического мониторинга, используемую для анализа функционирования ИУС, решения разнообразных задач оптимизации в ИТ-инфраструктуре, задач планирования и пр. При этом для эффективного управления ИТ-инфраструктурой следует применять в тесной взаимосвязи статистический и реактивный мониторинги, когда для решения одной части задач используется статистический метод, для других – реактивный, для третьих – комбинация статистического и реактивного мониторинга или применяется проактивный мониторинг.

В настоящее время для получения данных в процессе мониторинга используются различные методы, среди которых можно выделить два наиболее простых и очевидных альтернативных метода – активный и пассивный мониторинги [123, 354]. В первом случае со стороны СУИ проводится опрос ОМУ с целью получения значений параметров функционирования, во втором, – ОМУ путем отправки специальных сообщений самостоятельно информирует СУИ об изменении своего состояния в случае пересечения текущими значениями контролируемых параметров установленных пороговых значений.

Перспективным для СУИ выглядит метод получения данных мониторинга, основанный на самостоятельном сборе, обработке и передаче контрольной информации программными элементами ОМУ. В этом случае осуществляется непосредственный контроль элементов приложения самим приложением, а о работоспособности приложения сервер СУИ судит на основании периодически получаемых сообщений от механизмов встроенного самоконтроля. При отсутствии таких сообщений в течение заданного интервала времени сервер СУИ делает вывод о неработоспособности приложения.

Кроме этих методов получения данных мониторинга целесообразно использовать агентскую технологию [117, 120], а также методы получения информации о функционировании ОМУ по косвенным признакам. Мониторинг на основании анализа косвенных признаков позволяет, например, по отклику на утилиту ping протокола ICMP к ОМУ, подключенному к сети, судить о работоспособности всего тракта передачи данных, соединяющего сервер СУИ с опрашиваемым устройством, либо по характеру запущенных на сервере процессов или по наличию запросов к БД судить о функционировании распределенного приложения. Недостаток косвенного метода мониторинга заключается в невозможности точного определения показателей качества работы приложения.

Метод активного мониторинга путем опроса предполагает отправку сервером СУИ запроса к ОМУ на получение информации о параметрах, характеризующих его состояние, и соответствующего отклика от ОМУ. Как правило, опрос проводится с фиксированным, заранее определенным периодом.

При пассивном мониторинге сообщения о событиях самостоятельно и асинхронно генерируются ОМУ и передаются в сервер СУИ. При этом часто используются стандартные протоколы, например, SNMP. Сообщения о событиях вырабатываются сетевыми элементами или элементами распределенных функциональных подсистем только в случае превышения заранее установленных пороговых значений. Этот метод очень часто используется вместо опроса при мониторинге телекоммуникационного оборудования и серверов, поскольку позволяет существенно уменьшить накладной сетевой трафик. Однако для контроля глобальных параметров ОМУ, являющихся функцией локальных свойств зависимых ОМУ, метод сообщений требует непрерывной генерации событий с высокой частотой, что приведет к такой же существенной загрузке каналов связи, как и метод периодического опроса.

В [268] предложен подход, позволяющий минимизировать трафик данных результатов опроса при реактивном мониторинге. Подход основан на том, что возможные значения переменных, описывающих состояние ОМУ, принадлежат некоторому диапазону. Знание и учет предельных значений позволяют СУИ на основании предыдущих значений предсказать возможное будущее состояние ОМУ и выполнять апериодические опросы только в случае приближения к аварийному состоянию. Предложенный в [268] метод дает возможность объединить преимущества метода опроса и метода отсылки сообщений, но существенно усложняет алгоритмы работы СУИ.

Каждый из методов активного и пассивного мониторинга имеет преимущества и недостатки, и вполне логично, что перспективным является комбинированный – интегральный подход, сочетающий преимущества обоих методов. При этом возникают задачи распределения функций между методами, интеграции результатов работы каждого из них, разработки структуры центральной и периферийных частей СУИ. Это же касается и реализации в СУИ управляющих функций, которые могут в виде команд управления поступать из центра управления или генерироваться в ОМУ при отслеживании и реализации общей политики управления. В обоих методах позволяет избежать недостатков применение агентской технологии при мониторинге и управлении в СУИ, которая является одной из основных особенностей разрабатываемой ИТ УИТИ.

Сложные, но очень интересные задачи возникают при автоматизации мониторинга и распределения ресурсов в случае использования облачных вычис-

лений [238]. Распределение ресурсов между сервисами в настоящее время проводится в ручную при запуске процедуры инициализации. При этом администратор должен проанализировать данные о производительность ресурсов. Для принятия решения об увеличении или уменьшении объемов ресурсов, выделяемых сервисам, необходимо постоянно контролировать качество услуг, а для эффективного управления производительностью приложений и гармоничного распределения ресурсов необходимо сочетать мониторинг производительности с аналитикой [238].

4.2.2. Специфика подсистемы мониторинга ОМУ

Для контроля работоспособности ИТ-инфраструктуры и оценки качества предоставляемых услуг в ИТ УИТИ должны поддерживаться статистический, реактивный и проактивный [315] методы мониторинга. Статистический мониторинг используется при анализе текущего состояния и тенденций поведения ИТ-инфраструктуры, но менее пригоден для обнаружения и локализации неисправностей [135, 283]. Реактивный мониторинг, использующийся для выявления аномального состояния и поведения элементов ИТ-инфраструктуры, позволяет минимизировать объем контрольной информации, передаваемой по сети. Проактивный мониторинг [268, 296, 315] прогнозирует критические состояния, что позволяет предотвратить возникновение отказов в ИТ-инфраструктуре. Интеграция методов активного и пассивного мониторинга позволяет воспользоваться преимуществами каждого из них [176].

В ИТ УИТИ для каждого ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, определено множество параметров $p_l^{(i,j)}$, $l=\overline{1,n}$:

$$P^{(i,j)}=\{p_1^{(i,j)},p_2^{(i,j)},...,p_n^{(i,j)}\}, \tag{4.8}$$

которые оказывают существенное влияние на состояние и качество функционирования ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$. Формирование множества $p_l^{(i,j)}$ проводится на основе анализа функционального назначения ОМУ $O_{i,j}$, условий его функционирования, статистических данных о функционировании ОМУ и других факторов. Эту задачу может решать администратор с использованием функционала подсистемы анализа.

Сбор значений параметров $p_l^{(i,j)}$, $l=\overline{1,n}$, для каждого $O_{i,j}$ осуществляет подсистема мониторинга. Эта подсистема собирает данные, измеряет значения параметров и формирует массив

$$P=\bigcup_{i,j}^{I,J}P^{(i,j)} \tag{4.9}$$

данных мониторинга.

Текущие массивы P накапливаются в БД и формируют базу данных с результатами мониторинга $P(t)$.

Для каждого параметра $p_l^{(i,j)}\in P^{(i,j)}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, $l=\overline{1,n}$, должны быть определены нормативные (целевые) значения $b_l^{(i,j)}\in B^{(i,j)}$, $l=\overline{1,n}$, $i=\overline{1,I_j}$,

$j=\overline{1,J}$, где $B^{(i,j)}$ представляет множество нормативных значений параметров ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$:

$$B^{(i,j)}=\{b_1^{(i,j)},b_2^{(i,j)},...,b_n^{(i,j)}\}. \quad (4.10)$$

Относительно значений из множества $B^{(i,j)}$ производится определение степени снижения значений параметров $P^{(i,j)}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$ определяющих показатели эффективности функционирования ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$ и качества предоставляемых услуг.

Можно предложить несколько вариантов определения нормативных или эталонных значений параметров множества $B^{(i,j)}$ в реальной среде эксплуатации ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$.

В первом случае значения $b_l^{(i,j)}\in B^{(i,j)}$, $l=\overline{1,n}$, определяются по результатам выполнения ряда тестов после установки ОС, инсталляции ПО и оптимизационной настройки вычислительной системы. После чего вычисляется среднее значение, которое принимается нормативным и используется для формирования (4.10).

Во втором случае для определения значений для формирования (4.10) в течение достаточно длительного времени, которое может измеряться неделями, подсистема мониторинга собирает данные о функционировании ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$. Усредненные значения после коррекции администратором принимаются за нормативные. В дальнейшем именно с ними проводится сравнение значений параметров ОМУ, определенных в (4.8).

Третий вариант характеризуется определяющей ролью самих ОМУ. Каждый ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, самостоятельно периодически запускает тестовые задания, накапливает статистику, по шаблонам определяет нормативные значения и в случае существенных отклонений значений (4.8) от (4.10) информирует СУИ. Накопленные статистические данные могут сбрасываться оператором.

Кроме того, нормативные значения могут не измеряться, а задаваться, например, администратором СУИ на основании документов, регламентирующих требования к функционированию определенного класса ОМУ.

В модуле первичной обработки проводится преобразование и нормирование значений параметров массива *P*. Нормирование осуществляется относительно значений $b_l^{(i,j)}\in B^{(i,j)}$, $l=\overline{1,n}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, и в БД модуля мониторинга хранятся нормированные значения измеряемых параметров $p_l^{(i,j)}$, $l=\overline{1,n}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$.

Для каждого параметра $p_l^{(i,j)}$, $l=\overline{1,n}$, каждого ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, определены пороговые значения $h_l^{(i,j)}$, $l=\overline{1,n}$, нормального функционирования, нормированные относительно нормативных значений $b_l^{(i,j)}\in B^{(i,j)}$ параметров

$p_l^{(i,j)}$, $l=\overline{1,n}$. Пересечение пороговых значений свидетельствует о нештатной работе ОМУ. При этом инициируется соответствующее событие и информируется администратор СУИ.

Множество $H^{(i,j)}$ пороговых значений параметров $P^{(i,j)}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$,

$$H^{(i,j)}=\{h_1^{(i,j)},h_2^{(i,j)},...,h_n^{(i,j)}\}, \tag{4.11}$$

хранится в БД мониторинга.

В ИТ УИТИ при определении критичности значений параметров и при принятии решений о состоянии ОМУ использованы не обычные пороги (4.11), а пороги со свойством гистерезиса [123], когда каждому значению $h_l^{(i,j)}\in H^{(i,j)}$, $l=\overline{1,n}$, сопоставляются два значения ${}^{+}h_l^{(i,j)}$ и ${}^{-}h_l^{(i,j)}$ таких, что выполняются условия

$$\begin{aligned} {}^{+}h_l^{(i,j)} &= h_l^{(i,j)}+\Delta^{+}h_l^{(i,j)};\\ {}^{-}h_l^{(i,j)} &= h_l^{(i,j)}+\Delta^{-}h_l^{(i,j)}, \end{aligned} \tag{4.12}$$

где ${}^{+}h_l^{(i,j)}$ и ${}^{-}h_l^{(i,j)}$ – соответственно положительное и отрицательное значения гистерезисного порога, а $\Delta^{+}h_l^{(i,j)}$ и $\Delta^{-}h_l^{(i,j)}$ – положительное и отрицательное приращения порогового значения $h_l^{(i,j)}$.

При использовании традиционного порога изменение состояния $S_{i,j}$ ОМУ $O_{i,j}$ фиксируется при каждом пересечении порогового значения (рис. 4.11,*а*). В случае девиации состояния в зоне пороговых значений происходит большое количество срабатываний на выходе модуля сравнения данных (рис. 4.11,*в*). Каждое изменение состояния вызывает необходимость включения механизмов реакции СУИ на аномальное состояние ОМУ $O_{i,j}$ и информирования администратора СУИ о появления критичных ситуаций.

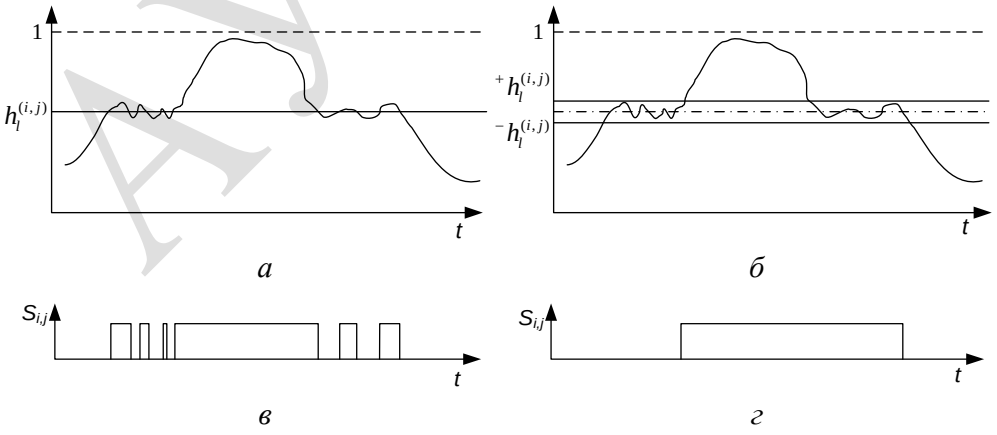


Рис. 4.11. Сравнение количества срабатываний решающей схемы без использования гистерезиса и при использовании гистерезиса

АУЭС КП

Применение в СУИ порогов со свойством гистерезиса (4.12) (рис. 4.11,*б*) позволяет существенно уменьшить количество срабатываний блока сравнения данных (рис. 4.11,*г*), поскольку сообщение об изменении состояния выдается только после того, когда изменение состояния надежно зафиксировано.

4.2.3. Особенности агентских технологий мониторинга и управления ИТ-инфраструктурой

Эффективное управление ИТ-инфраструктурой требует достоверной информации об управляемой системе. Для получения этой информации осуществляется сбор и обработка данных о функционировании ИТ-инфраструктуры и ее элементов. Эти процессы составляют сущность этапов мониторинга и анализа. Жесткие требования к управляемости ИТ-инфраструктуры предопределяют непрерывный или с малым периодом контроль параметров ОМУ единственным способом получения информации. Именно на основании данных такого контроля вырабатываются управляющие решения.

Сложность управляемой ИТ-инфраструктуры, большое количество предоставляемых ею услуг требуют контроля большого числа различных параметров. Причем для передачи значений параметров используется та же сеть, что и для передачи информации пользователей, которая также является объектом контроля и управления. В этом случае передача большого количества значений параметров, необходимых для принятия управляющих решений, создает большую нагрузку на управляемую систему. Поэтому уменьшение объема передаваемых данных о параметрах функционирования всех ОМУ является важной задачей. Один из основных и кардинальных способов сокращения сетевого трафика – применение агентской технологии, использующей агентов и метаагентов, способных принимать самостоятельные решения [117].

Предлагаемая концепция реализации ИТ УИТИ использует агентскую технологию [14, 164, 234] для мониторинга и дистанционного управления аппаратно-программными ОМУ. В этом случае в ОМУ встраиваются оригинальные программные агенты, созданные при разработке СУИ [108, 120], или типовые агенты, например, аппаратно-программные агенты стандартных протоколов управления, таких как SNMP, RMON, CMIP и др.

Агентский подход можно использовать не только для управления программным обеспечением, техническими средствами, функциональными и технологическими подсистемами ИТ-инфраструктуры, но и для организации самой СУИ при построении иерархической системы управления с региональными серверами [154]. В этом случае на основе агентов строится многоагентская СУИ, обладающая высокой степенью автоматизации при решении заданий управления и большей гибкостью, поскольку может решать специфические задачи управления в отдельной зоне ответственности. Агенты могут быть стационарными и мобильными [117]. В отличие от стационарных агентов, не способных к перемещению в ИТ-инфраструктуре, мобильные агенты могут перемещаться по сети с учетом ее загруженности, отключения отдельных узлов или фрагментов сети и пр. Мобильные агенты могут самостоятельно перемещаться по сети, управляя своей миграцией, прекращать работу на одном узле и возобновлять на другом [120].

Автономные программные агенты обладают свойствами интеллектуальности [274], мобильности [216] и способности взаимодействовать [224].

Автономные программные агенты самостоятельно решают задачи мониторинга и управления, а также координируют собственные действия для решения задач, поставленных перед ними системными администраторами посредством СУИ. В отличие от активных объектов [14] и других типов программных компонентов автономные агенты, обладая интеллектуальными свойствами и иногда будучи мобильными, способны самостоятельно генерироваться и действовать. При этом они способны координировать свои действия для достижения целей функционирования и решения поставленных перед ними задач. Одной из таких задач является контроль функционирования аппаратно-программного обеспечения телекоммуникационных узлов и серверов приложений с возможностью, например, самостоятельного принятия решений о перезагрузке, перераспределении ресурсов или реализации других мероприятий, направленных на восстановление эффективной работы серверов. Автоматизация таких процессов позволяет существенно сократить расходы на эксплуатацию ИТ-инфраструктуры.

Различная степень интеллектуальности позволяет агентам генерировать цели функционирования в зависимости от поставленных перед ними задач. При этом агенты могут осуществлять проактивное, т. е. упреждающее поведение, обучаться, самостоятельно принимать решения на основании получаемой от окружения информации.

Способность взаимодействовать позволяет агентам осуществлять координированные действия совместно с другими агентами для достижения общих целей.

В отличие от технологии клиент-сервер, при которой клиент через сеть обменивается с сервером информацией, в технологии мобильных агентов позволяется миграция части распределенного приложения, в результате чего удаленное взаимодействие заменяется локальным.

Технология мобильных агентов имеет и другие преимущества по сравнению с технологией клиент-сервер. В частности, использование мобильных агентов позволяет сократить накладной трафик за счет решения части задач удаленно агентами. Это наиболее ощутимо при интенсивном обмене между клиентом и сервером. Тогда мобильный агент может быть направлен в удаленный узел для обработки данных, а по окончании обработки вернуться с результатами. Последний пример характерен для Grid-сетей, в которых мобильные агенты можно использовать и для решения проблемы недостаточности ресурсов. В этом случае агент, мигрируя в сети, может найти сервер с количеством свободных вычислительных ресурсов, достаточных для решения задачи, выполнить задачу распределенного приложения и вернуть результаты.

Эффективность решений на основе агентских технологий для автоматизации деловых процессов поставщиков услуг и операторов связи продемонстрирована в работе [234]. В [164] показана целесообразность использования мобильных агентов для автоматизированного управления программным обеспечением в Grid-системах, когда программные компоненты и вычислительные ресурсы распределены среди множества сетевых узлов и объединены сетями связи. Агентские технологии с мобильными агентами годятся для решения множества задач управления в специализированных ИТ-инфраструктурах, объединяющих множество различных информационных и сетевых технологий и предоставляющих ИТ-услуги. Технология мобильных агентов прекрасно подходит для решения задач в ИУС с мобильными пользователями и группами пользователей, а также позволяет осуществлять адаптацию к изменяющимся характеристикам технических средств и изменяющимся потребностям пользователей при их перемещении.

Для того чтобы агенты могли перемещаться по сети и выполнять какие-либо действия в ОМУ нужна платформа, способная принять агента и создать ему условия, необходимые для решения задач управления, обеспечить его коммуникацию с другими агентами, а также идентификацию, управление агентом, целостность информации и пр.

Преимуществами агентской технологии являются:

- уменьшение загруженности сети за счет сокращения служебного трафика;
- возможность автономного выполнения задач;
- возможность работы без постоянной связи с сервером;
- меньшее использование вычислительных ресурсов агентом (при отсутствии агента на сервере ресурсы, зарезервированные для агента, не задействуются);
- возможность конкурентного выполнения заданий в многоагентской среде;
- параллельное выполнение заданий группой агентов сокращает время на решение общей задачи;
- миграция агентов вместе с решаемыми им задачами позволяет выровнять использование вычислительных ресурсов и снизить вероятность перегрузки отдельных узлов;
- использование мобильных агентов позволяет освободить администраторов от операций инсталляции, запуска и завершения использования программных компонентов.

К недостаткам агентской технологии следует отнести высокую сложность реализации, возможную неопределенность в работе и проблематичность использования в сетях с системой комплексной защиты информации. Эти и другие недостатки усложняют решение вопроса использования этой технологии в ИУС специального назначения. Поэтому агентские технологии хорошо подходят для автоматизированного управления программным обеспечением в Grid-окружении, для организации мониторинга и управления в корпоративных ИТ-инфраструктурах, но имеют ограниченное применение в ИУС специального назначения.

4.2.4. Преимущества использования ЛОУ в ИТ УИТИ

Предложенная и реализованная в ИТ УИТИ концепция управления ИТ-инфраструктурой ориентирована на применение агентского подхода [164, 216, 224, 234, 274] с использованием автономных программных агентов, обладающих свойствами интеллектуальности, мобильности, способности взаимодействовать и самостоятельно решать задачи мониторинга, анализа и координации собственных действий для управлении ИТ-инфраструктурой.

Агенты устанавливаются на РОУ, но связываются с другими РОУ и/или ЛОУ. Агент собирает данные от одного или нескольких РОУ или ЛОУ, обрабатывает их и передает обобщенную информацию в сервер СУИ. Получая обобщенные команды от сервера СУИ, агент осуществляет управление ЛОУ и/или РОУ. Каждый агент работает в отдельном регионе распределенной ИТ-инфраструктуры, принимает и исполняет команды от сервера СУИ. Независимо от расположения каждый РОУ может обслуживаться только одним ЛОУ или агентом, а для удаленной работы с РОУ могут создаваться его копии на других агентах. В таких случаях копии выступают заместителями РОУ и переадресовывают ему все обращения.

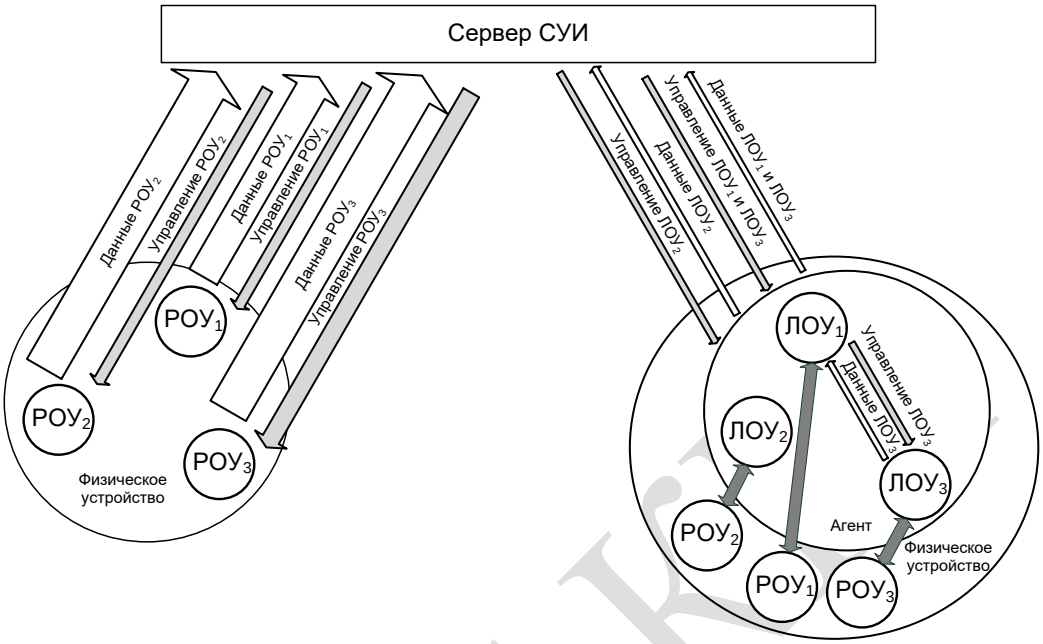


Рис. 4.12. Сравнение объемов потоков данных и управления при работе СУИ с РОУ и ЛОУ

При использовании агентской технологии, ориентированной на работу с ЛОУ, существенно снижается сетевой трафик СУИ по сравнению с системами, использующими технологии непосредственного взаимодействия с РОУ (рис. 4.12). Кроме уменьшения накладных сетевых расходов работа ИТ УИТИ с ЛОУ имеет и другие преимущества, приведенные в табл. 4.1.

Таблица 4.1. Сравнительный анализ работы ИТ УИТИ с ЛОУ и РОУ

Характеристика	ЛОУ	РОУ
Объем данных мониторинга	Незначительный	Большой
Подключение новых типов объектов	Просто	Затруднительно
Детализация команд управления	На уровне макрокоманд	Подробная
Автономное выполнение задач	Возможно	Невозможно
Распределение задач управления	Имеется	Отсутствует
Возможность передачи данных мониторинга зависимых объектов	Имеется	Отсутствует
Возможность передачи команд управления зависимым объектам	Имеется	Отсутствует
Возможность анализа состояния объекта	Имеется	Затруднена
Оперативное наращивание функций	Просто	Затруднено
Зависимость от аппаратно-программной платформы	Отсутствует	Существенная
Универсальность	Существенная	Ограниченная

АУПС К

Для работы с РОУ разных типов СУИ должна поддерживать большой набор протоколов и методов мониторинга и управления, а обработка данных осуществляется сервером СУИ. При работе СУИ с ЛОУ может быть использован универсальный инструментарий для широкого класса РОУ, который можно будет задействовать для работы с новыми типами РОУ.

Работа разрабатываемой ИТ УИТИ с ЛОУ, а не РОУ принципиально отличает предлагаемую СУИ от известных СУИ с подобными функциями.

ВЫВОДЫ К ГЛАВЕ 4

1. Введено и определено понятие логического объекта мониторинга и управления для абстрактного представления информационно-вычислительных ресурсов и элементов ИТ-инфраструктуры, которые проявляют свои свойства в целях и с точки зрения управления ИТ-инфраструктурой. Использование ЛОУ принципиально отличает предлагаемую ИТ. Такой подход позволяет использовать общие универсальные методы работы с элементами ИТ-инфраструктуры, а также единый метод или несколько универсальных методов для оценки состояния ЛОУ.

2. Предложен механизм взаимосвязи реальных и логических объектов в виде DALL-функций, основанный на рефлексии типов. Он предоставляет открытую модульную архитектуру, позволяющую реализовать функции управления новыми элементами ИТ-инфраструктуры без внесения изменений в саму технологию.

3. Обосновано применение агентской технологии при управлении ИТ-инфраструктурой. Применение агентов и метаагентов для сбора данных и управления ИТ-инфраструктурой позволяет не только сократить накладной сетевой трафик, но и повысить оперативность управления ИТ-инфраструктурой.

АУТСОРС

ГЛАВА 5

МЕТОДЫ АНАЛИЗА СОСТОЯНИЯ И СВЕДЕНИЯ МЕТРИК КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ

Из всех задач, которые необходимо решить при создании СУИ, наиболее слабо проработаны методы анализа ОМУ и сведения метрик качества функционирования ОМУ. Поэтому данная глава посвящена разработке методов оценки состояния, качества функционирования элементов ИТ-инфраструктуры и сведения метрик. Предложенные методы, в частности метод кодирования состояния и принцип классификации оценки качества функционирования элементов ИТ-инфраструктуры, позволят реализовать декомпозиционно-компенсационный способ организации управления уровнем услуг.

5.1. МЕТОДЫ АНАЛИЗА СОСТОЯНИЯ ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ

Решение задач управления, нацеленных на обеспечение функционирования ОМУ и ИТ-инфраструктуры в соответствии с заданным регламентом или связанных с поддержанием производительности ИТ-инфраструктуры на заданном уровне, требует знания состояния, производительности и других характеристик ее структурных компонентов. Данная глава посвящена разработке моделей и методов определения состояния ОМУ и вопросам сведения метрик, характеризующих качество функционирования ОМУ.

5.1.1. Проблемы анализа состояния объектов ИТ-инфраструктуры и сведения метрик

В процессе мониторинга ИТ-инфраструктуры СУИ осуществляет оценку состояния ОМУ, непрерывно контролируя их исправность и рассчитывая показатели их эффективности функционирования. Для этого осуществляется сбор информации, необходимой для оценки состояния ОМУ, в виде значений параметров функционирования, их обработка и последовательное сведение метрик во все более обобщенные метрики, комплексно характеризующие функционирование ОМУ разных уровней. Наряду с числовыми оценками важное место занимают классы качества функционирования, отнесение к которым позволяет определить перечень и приоритеты действий, применяемых с целью повышения эффективности функционирования ИТ-инфраструктуры.

При мониторинге параметров функционирования ОМУ один из основных и наиболее часто применяемых методов контроля заключается в сравнении измеренных величин с пороговыми значениями. В этом случае при пересечении пороговых значений вырабатывается аварийный сигнал.

Частота и продолжительность превышения пороговых значений – важные показатели при анализе функционирования ОМУ. Так, процент потерянных пакетов хорошо характеризует отдельный канал связи, а средняя очередь пакетов –

загруженность порта маршрутизатора. Такие параметры полезны, когда оценивается только отдельный элемент ТКС или простейшие системы.

При сравнении с пороговыми значениями множества параметров $S_i = \{s_{i,r}\}$, $r = 1,...,R$, характеризующих функционирование отдельного *i*-го ОМУ, необходимо проводить R сравнений с R пороговыми значениями из множества $L_i = \{l_{i,r}\}$, $r = 1,...,R$. В этом случае необходимо решить достаточно сложную проблему – определение общего состояния ОМУ, когда часть параметров превышает пороговые значения, а часть – нет. Можно ограничиться только фиксацией параметров, которые превышают пороговые значения. В противном случае для решения этой задачи должны быть разработаны соответствующие методы и алгоритмы.

Поэтому интерес представляют методы, позволяющие свести множество параметров $S_i = \{s_{i,r}\}$, $r = 1,...,R$, характеризующих функционирование отдельного *i*-го ОМУ, к одному обобщенному показателю (параметру) S_i^*. После этого можно будет провести сравнение значения этого параметра с единственным пороговым значением L_i^*.

Эта задача может быть достаточно просто решена, если, например, для анализа функционирования подсистемы проводится анализ одного параметра функционирования однотипных компонентов подсистемы. Например, для анализа функционирования ТКС можно взять среднюю загруженность всех каналов связи ядра сети. Тогда ее можно просто вычислить и легко сравнить среднюю загруженность группы каналов с одним пороговым значением.

5.1.2. Методы оценки качества функционирования элементов и подсистем ИТ-инфраструктуры

Управление процессами в ИТ-инфраструктуре требует оценки составляющих этих процессов, непосредственно или скрыто влияющих на подконтрольные метрики качества, и их сведение к обобщенным метрикам. В последнее время множество решений управления ОМУ все чаще заменяется централизованными СУИ [154, 155, 120], автоматизирующими решение всего комплекса задач управления ИТ-инфраструктурой. Их подсистемы мониторинга и анализа используют новейшие методы, охватывая широкий аспект деятельности функциональных единиц ИТ-инфраструктуры и проводя многогранную оценку качества их функционирования.

Для эффективного управления ИТ-инфраструктурой необходима информация о состоянии элементов и подсистем. Эту информацию предоставляют системы мониторинга и анализа функционирования ОМУ. Мониторинг параметров функционирования обеспечивает сбор первичной информации о состоянии различных ОМУ. Эта информация используется для определения и глубокого анализа значений показателей качества функционирования составных элементов и подсистем, оценки их надежности, прогнозирования поведения в последующие периоды. Мониторинг состояния может проводиться с помощью встроенных в ОМУ средств диагностики, например, для операционной системы Windows это могут быть инструментарий управления WMI или счетчики про-

изводительности (Performance Counters). Мониторинг также может осуществляться с помощью протоколов SNMP, CMIP и других средств, например программных агентов [117].

Для оценки качества функционирования IP-сетей разработаны стандартные метрики [308, 283], стандартизованы и методы проведения испытаний для определения значений этих метрик. Работу IP-сети можно оценить, используя небольшое число показателей, например, задержку передачи пакетов, частоту ошибок в пакетах и некоторые другие. Число стандартных метрик сильно ограничено. В [197] предложена система метрик для оценки работы провайдера сетевых услуг, организованная в форме дерева метрик. Там же рассмотрены вопросы корреляции метрик. Однако рассмотрение ограничивается только метриками для различных сетевых технологий.

В [194] представлена система мониторинга среды крупномасштабных параллельных и распределенных вычислений, применяющая методику сведения метрик, которая использует древовидное представление выполняемых процессов, что позволяет существенно уменьшить объемы данных, которые собираются и хранятся на жестких носителях. Система решает большинство рассмотренных выше вопросов, однако требует классификации данных мониторинга.

В [300] предлагается модель непрерывного взвешенного сведения метрик, которая специально разработана для оценки качества функционирования ПО. В отличие от [350] данные мониторинга проецируются на непрерывную шкалу единого масштаба, но используются строго определенные уровни сводных метрик, что сдерживает распространение модели на новые подходы к оценке кода, и система оценивания верхнего уровня применяет функции взвешивания. Такой подход требует длительного налаживания механизма оценки и не позволяет в кратчайшие сроки адаптироваться к текущим приоритетам и стратегии корпорации.

В работе [89] предлагаются система, метод сбора и сведения метрик функционирования ЭВМ с применением факторного анализа, позволяющие определить множество наиболее важных для данной единицы техники показателей, однако даже преобразование полученных характеристик часто делает невозможным их содержательное толкование.

Анализ известных реализаций СУИ, несмотря на их распространенность и всеохватность, обнаруживает определенные недостатки. Так, пакет программ HP Operations Manager [360, 376] требует дорогостоящей лицензии, привлечения значительного количества высококвалифицированных специалистов и длительного, затратного и многоэтапного процесса внедрения. Ряд признаков также исключает применение программных модулей System Center от Microsoft в качестве основы для построения СУИ для малого и среднего бизнеса. Некоторые из указанных проблем решаются в [107, 108, 37], однако не хватает универсального обобщающего математического аппарата оценки состояния ОМУ.

5.1.3. Постановка задачи анализа состояния элементов ИТ-инфраструктуры и сведения метрик

Для анализа состояния ОМУ и сведения метрик необходимо выработать принципы структурирования представления ИТ-инфраструктуры в СУИ, выделить группы однородных элементов. После этого можно применить процедуры

оценки качества функционирования ОМУ и функциональных единиц ИТ-инфраструктуры для выявления отклонений от состояний, характерных для режима штатной работы.

При этом возникает необходимость определения показателей качества функционирования распределенных элементов ИТ-инфраструктуры, привлеченных к выполнению возложенных на ИТ-инфраструктуру задач. Такими показателями являются обобщенные метрики, характеризующие определенные аспекты деятельности функциональных единиц ИТ-инфраструктуры, а именно: отдельные задачи, комплексы задач или же целые функциональные подсистемы. Для расчета обобщенных метрик необходимо иметь иерархию сведения оценок, которая содержит ориентированную на реально выполняемые задачи структуру зависимостей между ОМУ. Зависимости указывают на то, какие параметры, каких элементов ИТ-инфраструктуры и каким образом влияют на определенную характеристику отдельного ее элемента. Тогда задача состоит в разработке методов получения показателей функционирования элементов и их сведения на основе иерархического представления ИТ-инфраструктуры к обобщенным метрикам, комплексно описывающим качество выполнения возложенных на СУИ задач.

5.2. ПРИНЦИПЫ ПОСТРОЕНИЯ И ИСПОЛЬЗОВАНИЯ МЕТРИК ПРИ УПРАВЛЕНИИ ИТ-ИНФРАСТРУКТУРОЙ

5.2.1. Общие принципы анализа состояния и сведения метрик

Компоненты и взаимосвязи в ИТ-инфраструктуре целесообразно представлять в виде иерархической структуры, на верхнем уровне которой находятся функциональные и технологические подсистемы ИУС, на нижнем – аппаратные и программные элементы. Подсистемы включают в себя аппаратные и программные элементы, поэтому задачи определения качества функционирования ИТ-инфраструктуры могут решаться в такой последовательности: определение качества функционирования элементов ИТ-инфраструктуры, подсистем, ИТ-инфраструктуры в целом. Элементы могут входить в состав различных подсистем. Например, на сервере могут выполняться различные серверные приложения, поддерживающие функционирование разных подсистем ИУС. В этом случае состояние аппаратного сервера будет учитываться при определении состояний всех функциональных или технологических подсистем ИУС, элементы распределенных приложений которых работают на данном сервере.

Процесс мониторинга и оценки полученных показателей удобно проводить, когда ИТ-инфраструктура представлена в виде графа (см. рис. 4.4). Отсутствие явно выраженной иерархической структуры зависимостей между элементами в современных многоэлементных и многофункциональных ИУС ведет к необходимости выделения функциональных подсистем, т. е. групп элементов, объединенных по признаку участия в выполнении определенной функции (см. рис. 4.1). ИУС может содержать множество таких групп или составлять единое целое. В зависимости от задач некоторые группы могут оставаться неизменными длительное время или подвергаться частым изменениям, например, те, которые отвечают за дискретно выполняемые функции [78].

Интегрированность функциональных подсистем в рамках современных предприятий, ориентированных на минимизацию задействованных ресурсов, приводит к смешиванию аппаратно-программных частей подсистем, использованию одной и той же вычислительной и телекоммуникационной техники несколькими функциональными подсистемами. Например, на одном сервере могут находиться приложения для управления документооборотом и приложения бухгалтерского учета. Множественность функциональных связей представляется графом зависимостей (см. рис. 4.4), вершины которого отвечают ОМУ, а ребра связывают ОМУ. При этом граф определяет, какие параметры и какие ОМУ нижних уровней иерархии влияют на каждый отдельный ОМУ верхнего уровня. Этот граф в идеальном случае должен быть деревом. Отсутствие циклов (см. рис. 4.1) и приписывание каждой вершине соответствующего набора параметров ОМУ делает граф зависимостей удобным инструментом определения множества ОМУ и их параметров, достоверно влияющих на данный ОМУ. А поскольку граф охватывает максимум параметров ОМУ разных уровней, отвечает внутреннему строению ИУС, то на его основе можно рассчитывать обобщенные целевые показатели. При этом необходимо осуществлять отображение полученных показателей на качественную шкалу. Граф позволяет также выделять функциональные группы (ФГ) – множества однородных объектов, предназначенных для решения одной или нескольких совместных задач и описываемые одинаковым набором параметров.

При определении состояния ОМУ следует учитывать значения не всех параметров, характеризующих его состояние и функционирование, а только тех, которые оказывают наибольшее влияние на качество его работы. При этом, естественно, возникает погрешность. Чем меньше параметров учитывается при определении состояния ОМУ, тем больше погрешность. Однако, тем меньше затраты полосы пропускания на получение информации для определения значения параметров и меньше времени на обработку меньшего количества значений параметров.

Состояние всех ОМУ целесообразно характеризовать набором свойств и значениями показателей, влияющих на производительность, надежность, эффективность функционирования, прогноз поведения, эксплуатационные затраты и пр. При этом одни и те же параметры *i*-го ОМУ *j*-го иерархического уровня могут использоваться для определения различных свойств (рис. 5.1).

Рис. 5.1. Атрибуты и параметры ОМУ

При решении задач анализа ОМУ и определении пороговых значений показателей функционирования ОМУ часто полагаются на опыт и квалификацию администраторов. Такой подход вносит элемент субъективизма, поэтому необхо-

димо выбрать или разработать математические модели оценки состояния и определения качества функционирования элементов и подсистем ИТ-инфраструктуры.

Для определения состояния и качества функционирования ОМУ можно ограничиться анализом результатов тестовых самопроверок компонентов ОМУ (рис. 5.2).

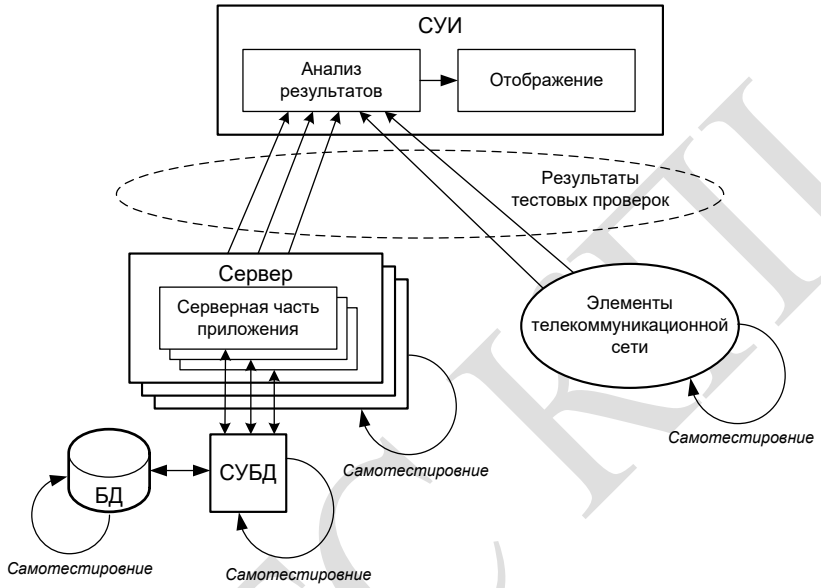


Рис. 5.2. Передача в СУИ результатов самотестирования ОМУ

В этом случае все ОМУ с заданной периодичностью выполняют самотестирование, а результаты сообщают в СУИ, где проводится сравнение результатов выполнения тестовых заданий с эталонными значениями. Это позволяет не только определить функционируют ли ОМУ, но и оценить качество их работы. При этом разработчики ПО могут встраивать тесты в приложения, не раскрывая деталей их работы. Такой метод позволяет выполнять универсальным образом анализ функционирования самых разнообразных, сильно различающихся приложений, реализованных на разных платформах, написанных на разных языках программирования. Однако при этом затрудняется локализация неисправностей и определение причин их появления, усложняется прогнозирование поведения ОМУ, определение управляющих воздействий. Поэтому, несмотря на явные преимущества, такой метод не может быть единственным методом, используемым для определения состояния ОМУ в СУИ.

Для быстрого и эффективного определения и анализа состояния различных ОМУ необходимо проводить мониторинг и анализ множества параметров функционирования ОМУ различными методами. Это позволит легко локализовать неисправности, осуществлять проактивное управление проблемами и эффективное управление ИТ-инфраструктурой.

Таким образом, необходимо выбрать или разработать различные методы и средства определения и анализа состояния ОМУ и сведения метрик для оценки качества функционирования ИТ-инфраструктуры. Анализ состояний необходим

для устранения неисправностей в ИТ-инфраструктуре. Для ОТС гораздо важнее сведение метрик, позволяющее оценить качество предоставляемых сервисов. Поэтому в работе рассматриваются методы как анализа состояния, так и сведения метрик.

Анализ состояний всех ОМУ $O_{i,j},\ i=\overline{1,I_j}$, $j=\overline{1,J}$, осуществляет подсистема анализа. На основании значений параметров $p_l^{(i,j)}$, $l=\overline{1,n}$, массива (4.10) подсистема анализа посредством функционала $\Phi_0^{(i,j)}$ определяет состояние $S_{i,j},\ i=\overline{1,I_j}$, $j=\overline{1,J}$ ОМУ $O_{i,j},\ i=\overline{1,I_j}$ и $j=\overline{1,J}$:

$$S_{i,j}=\Phi_0^{(i,j)}\{p_1^{(i,j)},p_2^{(i,j)},...,p_n^{(i,j)}\}. \tag{5.1}$$

Вид функционала $\Phi_0^{(i,j)},\ i=\overline{1,I_j}$, $j=\overline{1,J}$, определяется в результате статистического анализа массива $P(t)$ данных мониторинга, выбирается из соответствующего шаблона и закрепляется администратором за ОМУ $O_{i,j},\ i=\overline{1,I_j}$, $j=\overline{1,J}$:

$$\Phi_0^{(i,j)}\in\mathfrak{F}_0\{\Phi_{0,k}\}, \tag{5.2}$$

где $\mathfrak{F}_0$ – множество шаблонов функционала $\Phi_0^{(i,j)},\ i=\overline{1,I_j}$, $j=\overline{1,J}$, а $\Phi_{0,k}$, $k=\overline{1,K}$ – отдельный шаблон функционала $\Phi_0^{(i,j)},\ i=\overline{1,I_j}$, $j=\overline{1,J}$.

В случае, если состояние ОМУ определяется с выделением отдельных свойств, вместо состояния $S_{i,j}$ ОМУ рассматривается состояние $\hat{S}_{i,j}$ ОМУ, которое посредством функционала Φ_1 определяется следующим образом:

$$\hat{S}_{i,j}=\Phi_1\{s_1^{(i,j)},s_2^{(i,j)},...,s_M^{(i,j)}\}, \tag{5.3}$$

где $s_m^{(i,j)},\ m=\overline{1,M}$ – показатель, характеризующий отдельное *m*-е, в общем случае комплексное свойство ОМУ $O_{i,j}$, например производительность, надежность и др.

Для связи показателей отдельных свойств ОМУ $O_{i,j}$ с наиболее существенными параметрами, влияющими на *m*-е, введен функционал $\Phi_{2,m}^{(i,j)},\ m=\overline{1,M}$:

$$s_m^{(i,j)}=\Phi_{2,m}\{E_m^{(i,j-1)},P_m^{(i,j)}\}, \tag{5.4}$$

где $E_m^{(i,j-1)}=\{s_m^{(i,j-1)}\}$ – множество показателей *m*-го свойства ОМУ $O_{i,j-1}$, оказывающих влияние на свойство $s_m^{(i,j)}$, $P_m^{(i,j)}\in P^{(i,j)}$ – множество оригинальных параметров ОМУ, оказывающих влияние на *m*-е свойств $s_m^{(i,j)}$ ОМУ. Выбор и закрепление функционала $\Phi_{2,m}$ за отдельным свойством осуществляется подобно (5.2).

АУТСКИ

5.2.2. Иерархия и сведение метрик

Механизмы контроля и управления ИТ-услугами используют различные метрики [10]. ITIL, IS020000, COBIT, Six Sigma и eTom и другие стандарты предполагают использование метрик для оценки функционирования процессов и являются основой программ улучшения процессов управления при процессном подходе или повышения эффективности управления сервисами при сервисном подходе. В тоже время ITIL рассматривает только процессный подход, используемый при управлении ИТ-подразделением. COBIT, являясь стабильным методом проектирования, оказывается чрезвычайно удобным при проведении ИТ-аудита при процессном подходе к управлению ИТ-подразделением.

Six Sigma можно рассматривать как модель бизнес-процесса, главная цель которого состоит в уменьшении количества отдельных дефектов в пределах заданных диапазонов, связанных с сигмой, – среднеквадратичным отклонением. Этот подход, используемый в производстве, можно реплицировать на управление ИТ-инфраструктурой, для чего нужно ввести совместимые с Six Sigma метрики, позволяющие выявлять дефекты. После этого для метрик необходимо провести бенчмаркинг и накопить историю их использования с проведением соответствующего анализа. При этом дефект может рассматриваться как достижение некоторого заданного значения, а управление сводится к тому, чтобы не достигнуть этого значения. При таком подходе задачи Six Sigma можно рассматривать в контексте управления ИТ-инфраструктурой. В тоже время достижение высоких уровней качества ИТ-услуг в соответствии с Six Sigma может потребовать чрезмерных усилий и оказаться непомерно дорогим. Поэтому при использовании подхода Six Sigma следует руководствоваться принципом разумной достаточности, а не стремлением к идеалу, для того чтобы найти баланс между затратами и качеством предоставления ИТ-услуг.

Усилия TMF в OSS сосредоточены на аспектах обслуживания сети и решении бизнес-проблем, таких как забота о клиентах. Однако, поскольку фокус находится на определении методологий поддержки системы структурированных бизнес-процессов обслуживания сети, моделирование метрик и их корреляция не исследованы.

Для сетей передачи данных метрики четко определены и хорошо расписаны или даже стандартизованы. В тоже время методики и механизмы сведения метрик практически отсутствуют. В [197, 386, 385] рассмотрены вопросы корреляция, а также сведения метрик, когда на основании большого количества метрик нижних уровней иерархии ИТ-инфраструктуры определяется значение меньшего количества метрик расположенных выше уровней.

Для сведения в [197] использована концепция моделей дерева метрик (Metrics Tree Models – MTM). Элементы дерева метрик основаны на большой библиотеке MTM и моделируют специфические особенности топологии сети поставщика услуг, используемых технологий, обслуживаемых типов и профилей клиентов. MTM является механизмом описания корреляции между метриками. При этом метрики многоуровневые, конкретные, измеримые и определенные показателями работоспособности, их спецификация и интерпретация осуществляются в определенном контексте. В [197] метрики распределены по пяти уровням – технологическому, сетевому, сервисов, клиентов и бизнеса – и относятся к определенным аспектам деятельности OTC. Уровни бизнеса, клиентов и сервиса

отображают ориентированные на потребителей сетевые операции. Для пользователей важно, какие цели бизнеса могут быть реализованы, например увеличение количества обслуживаемых покупателей, непосредственно влияющего на прибыль. На технологическом уровне используются показатели работы отдельных устройств, например загруженность портов маршрутизатора. На сетевом уровне – такие параметры, как задержка передачи или пропускная способность. На клиентском – удовлетворенность абонентов и эффективность заботы о клиенте. Для каждого из уровней определены специфические метрики, причем количество метрик уменьшается при переходе от нижнего – технологического уровня, на котором специфицировано 185 метрик, к верхнему – уровню бизнеса с 10 метриками. На нижних уровнях используется много метрик из-за многообразия технологий и множества атрибутов в каждой технологии. Метрики верхних уровней часто являются объединением и абстрагированием метрик нижних уровней. Метрики коррелированы с метриками того же уровня и/или смежных выше и ниже расположенных уровней.

СУИ обрабатывает большое количество информации о функционировании ИУС, подсистем, элементов подсистем, сервисов и служб, полученную в результате мониторинга. Эта информация поступает в различных формах и форматах, часто несогласованных. Отдельные показатели имеют высокую либо незначительную степень корреляции. В то же время фокус управления в ИТ УИТИ смещен от управления элементами к управлению ресурсами и подсистемами. Представление информации очень важно для эффективного управления. Поэтому СУИ реализует функции сбора и анализа информации, оценивая прежде всего степень корреляции и осуществляя фильтрацию метрик на всех уровнях – бизнес-процессов (процессов деятельности), работы приложений, сервисов, подсистем, сети и элементов. Используя степени корреляции, можно определить какую информацию и в каком виде СУИ должна предоставлять различным категориям пользователей, сконструировать метрики и спроектировать их систему. Определение степени корреляции между метриками на всех уровнях ИТ-инфраструктуры дает важную информацию для проектирования системы метрик. Необходимо помнить, что плохо спроектированная система метрик может существенно препятствовать правильному функционированию ИТ-организации [10].

5.2.3. Требования к метрикам, используемым при управлении ИТ-инфраструктурой

Итак, управление ИТ-инфраструктурой предопределяет необходимость постоянного получения информации о состоянии ОМУ, ИУС и деятельности ИТ-подразделения и ее анализа.

Оценка состояния ОМУ обычно проводится сравнением значений параметров, измеренных с помощью различных метрик, с пороговыми значениями. Для эффективного применения ИТ УИТИ при создании СУИ для ИТ-инфраструктуры провайдера необходимо разработать методику выбора параметров, уточнить критерии выбора параметров и их комплексов, период и точность измерения для каждого из них и другие свойства. Выбор параметров и метрик должен основываться на приоритетах бизнеса и целях ИТ-подразделения. В иерархических ИТ-инфраструктурах количество метрик, QPI и KPI уменьшается при переходе

от нижних иерархических уровней к верхним [197]. Деятельность ИТ-подразделения также должна оцениваться объективными измеряемыми параметрами или метриками.

Метрики и целостная их система должны удовлетворять следующим требованиям [10, 197]:

- быть простыми, непротиворечивыми, измеримыми и не абстрактными, а конкретными без использования всеобъемлющих статусов и категорий;
- количество метрик должно быть не чрезмерным, а достаточным;
- метрики должны образовывать упорядоченную иерархию.

Предоставление ИТ-услуг осуществляется на высшем уровне управления ИТ-инфраструктурой. Здесь ИТ-подразделение выполняет организационное управление, руководствуясь принципами ITIL. При организационном управлении используются метрики, которые больше характеризуют работу ИТ-персонала, чем аппаратно-программных средств ИТ-инфраструктуры. Определение значений этих метрик чаще основывается на подсчете событий и проводится раз в неделю, месяц или квартал. Метрики верхнего уровня выбираются на основе разработанной методики из набора метрик, определенных и закрепленных в многочисленных стандартах [10, 180, 206, 215, 260, 318, 336, 338, 339]. При определенных условиях ИТ-подразделение дополнительно создает собственные системы оценок.

На нижних уровнях осуществляется операционное управление ИТ-инфраструктурой, при этом часть функций реализуется в автоматическом режиме. Для оценки качества функционирования ОМУ нужно KPI измерять гораздо чаще, чем это необходимо для оценки качества работы ИТ-подразделения. Для конструирования, интеграции и определения значений таких метрик нужно разработать соответствующую методику, учитывая указанные особенности. Метрики этих уровней в основном связаны с определением состояния элементов ИТ-инфраструктуры. Они практически не проработаны в международных стандартах. Такие методики отсутствуют, а литературных источников, посвященных решению этой проблемы, очень мало.

Для того чтобы охарактеризовать состояние ОМУ верхнего уровня иерархии ИТ-инфраструктуры в определенный момент времени или отдельный процесс за период времени одним показателем, необходимо осуществить комбинирование метрик, которые могут принимать значения в широких диапазонах.

Существует множество способов интеграции метрик, например, путем взвешенного суммирования [130] или использования более сложных способов, основанных, например, на методах статистической геометрии [124]. Метод взвешенного суммирования с учетом целей и приоритетов [10] использует сбалансированную карту показателей с приведением к системе оценок, имеющих три значения: (-1), соответствующее измеренному значению, далекому от целевого и ниже опасного; (0), соответствующее измеренному значению в интервале между опасным и целевым значениями; (+1), соответствующее измеренному значению, равному или превосходящему целевое. После умножения этих оценок на их вес проводится суммирование результатов. Преимущество метода состоит в возможности учета изменения требований бизнеса за счет изменения приоритетов метрик. Он удобен для оценки эффективности работы ИТ-подразделения, но при управлении ИТ-инфраструктурой используется редко.

5.2.4. Особенности работы с метриками в системе управления ИТ-инфраструктурой

При управлении ИТ-инфраструктурой необходимо использовать иерархически упорядоченную систему метрик – множество мер, каждая из которых позволяет получить численное значение некоторого отдельного свойства ОМУ.

Для проектирования, внедрения и применение метрик необходимо:

1. На начальном этапе определить стратегические цели и показатели для контроля достижения этих целей. На основе ITIL, Six Sigma, COBIT и других стандартов спроектировать процессы и определить необходимые сервисы. Выделить функциональные и технологические подсистемы – ОМУ самого верхнего уровня иерархии.

2. На этапе проектирования процессов или сервисов выбрать метрики верхнего уровня для измерения эффективности функционирования ИТ-подразделения, процессов и/или сервисов, состояния ОМУ высшего уровня ИТ-инфраструктуры. Эти метрики могут быть стандартными [10] и/или спроектированными ИТ-подразделением с учетом целей бизнеса и приоритетов ИТ-подразделения.

3. Создать систему метрик, основанную на объективно измеримых показателях. Определить, как эти метрики могут быть сконструированы и проверить возможность реализации метрик.

4. Декомпозировать метрики сверху вниз для подчиненных ИТ-отделов и ОМУ нижних уровней с распределением по всем нижним иерархическим уровням. Специфицировать KPI и KQI.

Таким образом, в целях оперативного управления ИТ-инфраструктурой следует проводить конструирование метрик сверху вниз, а определение их значений – снизу вверх. Такой подход удобен как для оценки работы ИТ-подразделения, так и для оценки качества сервисов, определения влияния функционирования ИТ-инфраструктуры на бизнес-цели. В частности, для оценки качества функционирования ИУС при операционном управлении ИТ-инфраструктурой следует сначала определить, какие метрики ОМУ верхних уровней необходимо оперативно отслеживать и какие параметры контролировать. После этого необходимо определить способы получения этих параметров на основе значений KPI ОМУ нижних уровней. Тем самым будут предопределены датчики, которые должны быть в ИУС, и параметры, значения которых необходимо измерять в ОМУ самого нижнего уровня.

На ниже расположенных уровнях ОМУ может быть измерено гораздо больше параметров, характеризующих состояние ОМУ, чем требуется для определения состояний ОМУ верхних уровней. При этом большая часть параметров оказывается не востребованной. Однако в СУИ должны присутствовать механизмы сбора значений этих параметров, поскольку неиспользуемые параметры ОМУ могут понадобиться в следующий момент времени, когда изменятся приоритеты бизнеса и будут пересмотрены метрики или сконструированы другие метрики.

5.2.5. Особенности работы с метриками в системах управления операторов телекоммуникационных сервисов

Для коммуникационных инфраструктур ОТС при оценке параметров функционирования сетей метрики необходимы для всех иерархических уровней, что определяется спецификой бизнеса, а интеграция метрик проводится снизу вверх [197]. Такой подход можно использовать и при оперативном управлении ИТ-инфраструктурой ОТС.

В этом случае необходимо:

1. Выбрать совокупность метрик для измерения параметров, характеризующих состояние ОМУ и элементов ИТ-инфраструктуры ОТС.
2. Проверить возможность реализации метрик.
3. Распределить метрики по иерархическим уровням.
4. Определить степень корреляции или другие показатели взаимозависимости между метриками.
5. Разработать методы и алгоритмы сведения метрик.
6. Задать принципы оценки параметров ОМУ с помощью соответствующих метрик или разработать шкалу градаций для измерения или оценки свойств или состояний ОМУ.
7. Определить перечень метрик, актуальный с точки зрения бизнеса на текущем интервале времени.
8. Периодически или при изменении приоритетов бизнеса делать ревизию этого перечня.

При таком подходе возникает достаточно сложная проблема сведения, интеграции или агрегирование метрик. Необходимо различать интегрированные метрики и агрегированные – объединенные в одну систему метрики. Причем одна и та же метрика может быть как интегрированной, так и агрегированной. Так, интегрированная метрика QoE удовлетворенности пользователя сервисом IPTV определяет итоговую удовлетворенность пользователя по балльной или процентной шкале относительно максимума. В агрегированном же виде эта метрика представляет собой совокупность метрик, среди которых может быть субъективная составляющая, метрика удобства пользовательского интерфейса, ассортимента программ и другие метрики.

Выбор рациональной системы метрик для сложной многоуровневой иерархической системы, включающей множество ОМУ с самыми разнообразными свойствами, является сложной задачей. Это обусловлено тем, что на нижних уровнях ИТ-инфраструктуры используется большое количество метрик. Обычно метрики нижнего уровня характеризуют функционирование коммуникационного и серверного оборудования и используются для обнаружения проблем в сетях, контроля качества и управления коммуникационным и серверным оборудованием. На выше расположенных уровнях используется меньшее количество метрик, причем иерархия метрик может не совпадать с иерархией ИТ-инфраструктуры. Для конкретных технологий передачи данных метрики четко определены. Так, метрики для IP сетей стандартизованы в [308, 283, 176, 177, 178], Также в [198, 288, 201] определены методы проведения испытаний при определении значений в метриках IP сетей. Но в целом для технологий выше расположенных уровней пока не существует стандартного набора метрик, характеризующих состояние ИТ-инфраструктуры.

Достаточно просто подлежат измерению бизнес-процессы ИТ-подразделения. В [10] изложены общие принципы проектирования и реализации метрик, используемых в операционной деятельности ИТ-подразделения.

Обработка, в том числе определение степени корреляции метрик нижних уровней, обычно не составляет труда, поскольку здесь чаще всего используются простейшие операции, например, суммирование для определения суммарной задержки на основании задержки передачи пакетов на отдельных участках, усреднение событий за период времени и др.

Сложность сведения или агрегирования метрик возрастает на промежуточных уровнях ИТ-инфраструктуры и при переходе к верхнему уровню ИУС. Поэтому в работе большое внимание уделено вопросам сведения метрик ИТ-инфраструктуры.

5.3. КОДИРОВАНИЕ СОСТОЯНИЯ ОБЪЕКТОВ МОНИТОРИНГА И УПРАВЛЕНИЯ

Для эффективного управления ИТ-инфраструктурой необходимо располагать информацией о состоянии $S_{i,j}$, производительности, надежности и других характеристиках ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$. Для этого подсистема мониторинга формирует массив P (4.10) значений параметров (4.8), а подсистема анализа определяет качество функционирования ОМУ.

В масштабных ИТ-инфраструктурах на работу ОМУ $O_{i,j}$ оказывает влияние большое количество параметров $p_l^{(i,j)}$ (4.8), которые просто измерить, но сложно обобщить в универсальной математической модели, пригодной для решения многочисленных задач управления. Поэтому целесообразно двигаться по пути создания множества простых моделей, отражающих отдельные аспекты функционирования ОМУ и пригодных для решения отдельных задач управления.

С целью упрощения последующей обработки представляется удобной универсальная схема кодирования статуса ОМУ, которая позволяет представить значения параметров, оказывающих влияние на эффективность и качество функционирования ОМУ, в виде символьного кода. Эта схема рассмотрена в данном параграфе.

Национальный центр компьютерной безопасности Министерства обороны США определил критерии оценки уровня защиты вычислительных систем – TCSES, на основании которых классифицировал оценку безопасности ИТ-инфраструктуры. Это сделало информационную безопасность величиной, допускающей измерение [222]. Возникает вопрос: можно ли перенести общий подход к классификации информационной безопасности на классификацию состояния ОМУ, так, чтобы состояние ОМУ также стало величиной, которую можно измерить? Далее дан положительный ответ на этот вопрос и предложен метод кодирования состояния ОМУ, делающий оценку состояния и качества функционирования ОМУ величинами, допускающими измерение.

Постановка задачи. Необходимо разработать универсальную схему кодирования состояния ОМУ, которая позволит свести значения параметров, оказывающих влияние на эффективность и качество функционирования ОМУ, к символам кода.

5.3.1. Метод кодирования состояния или статуса объектов мониторинга и управления

В [151] предложен метод кодирования состояния $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, $i=\overline{1,I_j}$ и $j=\overline{1,J}$, позволяющий свести значения параметров, оказывающих влияние на эффективность и качество функционирования ОМУ, к символам кода с целью упрощения последующей обработки. Суть предложенного метода кодирования заключается в следующем.

Состояние $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$ определяется модулем анализа ИТ УИТИ, исходя из значений параметров, оказывающих влияние на состояние ОМУ. Формальное описание определения состояния ОМУ приведено в п. 5.1.3, где определены задачи анализа качества функционирования ОМУ. Состояние $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, определяется в соответствии с выражением (5.3) и зависит от показателей $s_m^{(i,j)}$, $m=\overline{1,M}$, значение каждого из которых влияет на отдельное m-е, $m=\overline{1,M}$, свойство ОМУ $O_{i,j}$, например производительность, надежность и др.

Выражение (5.4) устанавливает зависимость показателей свойства $s_m^{(i,j)}$ от элементов множества $E_m^{(i,j-1)}=\{s_m^{(i,j-1)}\}$ показателей m-го свойства ниже расположенных по иерархии ОМУ $O_{i,j\text{-}1}$, $i=\overline{1,I_{j-1}}$, $j=\overline{1,J}$, влияющих на показатели свойства $s_m^{(i,j)}$ ОМУ $O_{i,j}$, и множества $P_m^{(i,j)}$ оригинальных параметров ОМУ $O_{i,j}$, оказывающих влияние на m-е свойство $s_m^{(i,j)}$ ОМУ $O_{i,j}$.

Для получения возможности кодирования состояния $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, функционал Φ_1 в (5.3) определяется следующим образом.

Каждый показатель $s_m^{(i,j)}$, $m=\overline{1,M}$, приводится к численному виду, преобразовывается так, чтобы максимальному значению $max\, s_m^{(i,j)}$ соответствовал максимальный положительный вклад в определение состояния $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, и нормируется относительно $\max s_m^{(i,j)}$. Тогда значения показателя $s_m^{(i,j)}$, $m=\overline{1,M}$, находятся в отрезке [0, 1]:

$$0 \le s_m^{(i,j)} \le 1, \quad m=\overline{1,M},\ i=\overline{1,I_j},\ j=\overline{1,J}. \tag{5.5}$$

Отрезок изменения $s_m^{(i,j)}$, $m=\overline{1,M}$, разбивается на $N_{m,i,j}$ непересекающихся диапазонов $d_{m,i,j}^{(n_m)}$, $n_m=\overline{2,N_{m,i,j}}$, с использованием $L_{m,i,j}=(N_{m,i,j}-1)$ пороговых значений $h_{m,i,j}^{(l_m)}$, $l_m=\overline{1,L_{m,i,j}}$. Нумерация диапазонов начинается от максимального значения $\max s_m^{(i,j)}$, равного 1.

Принадлежность значения $s_m^{(i,j)}$ к диапазону $d_{m,i,j}^{(n_m)}$ определяется следующим образом:

АУТСКИЙ

$$s_m^{(i,j)} \Rightarrow d_{m,i,j}^{(n_m)}, \tag{5.6}$$

если выполняется условие

$$\frac{1}{(L_{m,i,j}+1)}(L_{m,i,j}+1-d_{m,i,j}^{(n_m)}) < s_m^{(i,j)} \le \frac{1}{(L_{m,i,j}+1)}(L_{m,i,j}+2-d_{m,i,j}^{(n_m)}). \tag{5.7}$$

Величина $s_m^{(i,j)}$ кодируется символом $A_{b_{n_m}}^{(N_{m,i,j})}$ так, что значение b_{n_m}, $n_m = \overline{1, N_{m,i,j}}$ соответствует номеру диапазона $d_{m,i,j}^{(n_m)}$, в котором находится значение $s_m^{(i,j)}$. Символ $A_{b_{n_m}}^{(N_{m,i,j})}$ принадлежит алфавиту $\mathrm{A} = \{A_{b_{n_m}}^{(N_{m,i,j})}, n_m = \overline{2, N_{m,i,j}}, m = \overline{1, M}\}$.

Кодовая комбинация для обозначения состояния $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, $i = \overline{1, I_j}$, $j = \overline{1, J}$, имеет вид

$$A_{b_{n_1}}^{(N_{1,i,j})} A_{b_{n_2}}^{(N_{2,i,j})} \ldots A_{b_{n_M}}^{(N_{M,i,j})}. \tag{5.8}$$

Поскольку двумя наиболее важными свойствами, характеризующими работу любого ОМУ, являются показатели производительности и надежности, то в качестве примера рассмотрим кодирование состояния $\hat{S}_{i,j}$ ОМУ при $M = 2$.

На основе выражения (5.3) состояние $\hat{S}_{i,j}$ задается функционалом Φ_1:

$$\hat{S}_{i,j} = \Phi_1\{s_1^{(i,j)}, s_2^{(i,j)}\}, \tag{5.9}$$

где $s_1^{(i,j)}$, $s_2^{(i,j)}$ – соответственно значения показателей производительности и надежности ОМУ $O_{i,j}$, $i = \overline{1, I_j}$, $j = \overline{1, J}$. Показатель производительности $s_1^{(i,j)}$ i-го ОМУ $O_{i,j}$, $i = \overline{1, I_j}$, $j = \overline{1, J}$, в соответствии с (5.4) определяется через функционал

$$s_1^{(i,j)} = \Phi_{2,1}\{E_1^{(i,j-1)}, P_1^{(i,j)}\}, \tag{5.10}$$

где $E_1^{(i,j-1)}$ – множество показателей производительности ОМУ (j-1)-го иерархического уровня, оказывающих влияние на производительность ОМУ $O_{i,j}$, а $P_1^{(i,j)}$ – множество оригинальных параметров ОМУ $O_{i,j}$, влияющих на его производительность.

Показатель потенциальной надежности $s_2^{(i,j)}$ ОМУ $O_{i,j}$ подобно (5.10) определяется с помощью функционала

$$s_2^{(i,j)} = \Phi_{2,2}\{E_2^{(i,j-1)}, P_2^{(i,j)}\}, \tag{5.11}$$

где $E_2^{(i,j-1)}$ – множество показателей потенциальной надежности элементов (j-1)-го иерархического уровня ОМУ, оказывающих влияние на надежность ОМУ $O_{i,j}$, а $P_2^{(i,j)}$ – множество оригинальных параметров ОМУ $O_{i,j}$, влияющих на его потенциальную надежность.

АУТСКИЙ

Пример кодирования состояния $\hat{S}_{i,j}$ для трех- ($L_{1,i,j}=3$) и одно- ($L_{2,i,j}=1$) порогового ранжирования состояний, а также для одно- ($m=1$) и двух- ($m=2$) параметрического определения состояния $\hat{S}_{i,j}$ приведен в табл. 5.1.

Таблица 5.1. Пример кодирования состояния или статуса ОМУ

Код состояния $\hat{S}_{i,j}$ при $m=1$	Код состояния $\hat{S}_{i,j}$ при $m=2$	Показатель производительности $s_1^{(i,j)}$	Показатель надежности $s_2^{(i,j)}$
$A_1^{(4)}$	$A_1^{(4)}\ A_1^{(2)}$	$h_{1,i,j}^{(3)} < s_1^{(i,j)} \le 1$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le 1$
	$A_1^{(4)}\ A_2^{(2)}$	$h_{1,i,j}^{(3)} \le s_1^{(i,j)} \le 1$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$
$A_2^{(4)}$	$A_2^{(4)}\ A_1^{(2)}$	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le 1$
	$A_2^{(4)}\ A_2^{(2)}$	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$
$A_3^{(4)}$	$A_3^{(4)}\ A_1^{(2)}$	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le 1$
	$A_3^{(4)}\ A_2^{(2)}$	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$
$A_4^{(4)}$	$A_4^{(4)}\ A_1^{(2)}$	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le 1$
	$A_4^{(4)}\ A_2^{(2)}$	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$

Здесь $h_{1,i,j}^{(l_1)}$ и $h_{2,i,j}^{(l_2)}$, $l_1=1,2,3$, $l_2=1$ – пороговые значения для показателей производительности и надежности i-го ОМУ, соответственно, причем выполняется условие

$$h_{m,i,j}^{(l_m)} \ge h_{m,i,j}^{(l_m+1)}. \tag{5.12}$$

Если при определении состояния ОМУ показатели надежности не учитываются, то состояние $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, будет кодироваться только одним символом алфавита $\mathrm{A}=\{A_{b_{n_1}}^{(N_{1,i,j})}, n_1=\overline{2,N_{1,i,j}}\}$.

5.3.2. Принципы определения значений пороговых величин

Значения порогов $h_{m,i,j}^{(l_m)}$, $l_m=1,...,L_{m,i,j}$, устанавливаются администраторами и могут изменяться в процессе работы СУИ. По умолчанию они устанавливаются следующим образом:

$$h_{m,i,j}^{(l_m)} = 1 - \frac{1}{(L_{m,i,j}+1)} l_m \tag{5.13}$$

для всех $l_m=\overline{1,L_{m,i,j}}$.

Значения (5.13) нормированы относительно нормативных значений (4.10) величин $s_m^{(i,j)}$, $m=\overline{1,M}$. Соображения по поводу определения нормативных значений

$b_l^{(i,j)} \in B^{(i,j)}$, $l = \overline{1,n}$, изложены в п. 4.2.2. В то же время модернизация ОМУ $O_{i,j}$, например увеличением объема ОЗУ, приводит к повышению значений показателей качества функционирования ОМУ $O_{i,j}$, поэтому в таких случаях необходимо переопределить значения параметров множества $B^{(i,j)}$. При увеличении количества пользователей ИУС, количества документов в БД, объемов обрабатываемой информации и других показателей нагрузки производительность ИТ-инфраструктуры снижается. В этом случае нормативные показатели не переопределяются, а проводится понижение класса состояния ОМУ, например, статус $A_2^{(4)}$ изменяется на $A_3^{(4)}$.

В (5.7), (5.12) и табл. 5.1 сравнение $s_m^{(i,j)}$, $m = \overline{1,M}$, с $h_{m,i,j}^{(l_m)}$, $l_m = \overline{1,L_{m,i,j}}$, выполняется по условию – не больше. Это позволяет устанавливать равные значения $h_{m,i,j}^{(l_m)}$, $l_m = \overline{1,L_{m,i,j}}$, сводя предложенную схему кодирования состояния или статуса, например, к двух- или однопороговой (рис. 5.3).

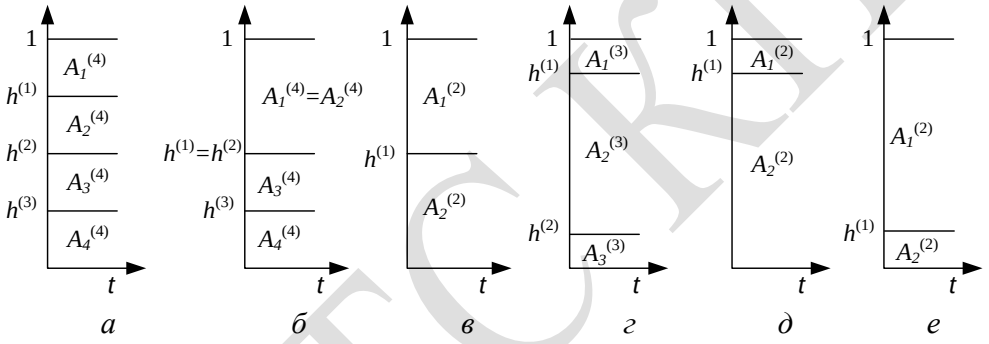


Рис. 5.3. Примеры реализации пороговых схем: *а* – трехпороговая с равными значениями диапазонов; *б* – перевод трехпороговой схемы в двухпороговую; *в* – однопороговая схема; *г* – вариант трехпороговой схемы; *д* и *е* – варианты однопороговой схемы

На рис. 5.3 приведены различные варианты пороговых схем, используемые при реализации решающих правил для оценки состояния $\hat{S}_{i,j}$ в соответствии с выражением (5.1) или кодирования состояния $\hat{S}_{i,j}$ ОМУ $O_{i,j}$, $i = \overline{1,I_j}$, $j = \overline{1,J}$. Желательно, чтобы все пороги обладали свойством гистерезиса (4.12).

Трехпороговая схема, приведенная на рис. 5.3,*а*, используется для соотнесения диапазонов отрезка [0, 1] с такими категорийными именованиями качества функционирования как «отлично», «хорошо», «удовлетворительно», «плохо» ($A_1^{(4)}$ соответствует «отлично», …, $A_4^{(4)}$ – «плохо»).

На рис. 5.3,*а* приведен вариант реализации трехпороговой схемы. В этом случае решение о состоянии $S_{i,j}$ ОМУ $O_{i,j}$ может выноситься на основании соотнесения диапазонов отрезка [0, 1] с такими категорийными именованиями качества функционирования как «отлично», «хорошо», «удовлетворительно», «плохо» ($A_1^{(4)}$ соответствует «отлично», …, $A_4^{(4)}$ – «плохо»).

Многопороговые схемы принятия решений о состоянии $S_{i,j}$ или состоянии $\hat{S}_{i,j}$ ОМУ $O_{i,j}$ хороши для СУИ, построенных с использованием агентских технологий.

АУТСКИ

При этом в сервер СУИ агенты передают сообщения только об изменении состояния ОМУ $O_{i,j}$ при переходах $A_{b_{n_m}}^{(N_{m,i,j})} \Leftrightarrow A_{b_{(n+1)_m}}^{(N_{m,i,j})}$. Большое количество порогов позволяет плавно отслеживать динамику изменения состояния ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, и повысить чувствительность при принятии решений по управлению.

На рис. 5.3,*б* трехпороговая схема сведена к двухпороговой приписыванием равных значений порогам $h^{(1)}$ и $h^{(2)}$. Первый диапазон соответствует функционированию ОМУ в соответствии с заданным регламентом, второй диапазон – «предупреждение», третий – «критическая ситуация». Такой подход может быть использован для приведения любой l_m-пороговой схемы, $l_m=\overline{1,L_{m,i,j}}$, к $(l_m-l_m^k)$-пороговой схеме, $l_m^k=1,...,(L_{m,i,j}-1)$, при этом должно выполняться условие $(l_m-l_m^k)\geq 1$.

Схема на рис. 5.3,*е* часто используется для оценки состояния сетевого оборудования. В этом случае производительность порта будет находиться в обратно пропорциональной зависимости от загруженности линии связи, подсоединенной к порту. Величина порога традиционно устанавливается на уровне 0,05–0,1, а падение производительности до 5–10% соответствует загруженности линии связи на 90-95%.

Кодирование состояния $\hat{S}_{i,j}$ с использованием выражений (5.3) и (5.9) позволяет унифицировать оценку состояния ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$. Для большинства практических случаев целесообразно использовать трехпороговые или однопороговые схемы при вынесении решения о состоянии или статусе ОМУ.

5.3.3. Классификация качества функционирования

Применение трехпороговых и однопороговых схем целесообразно не только при кодификации состояния ОМУ, но и при классификации оценки качества функционирования ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$. Качество функционирования ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, в основном оценивается по двум показателям – производительности и надежности [104]. Поэтому для практического применения удобно использовать следующий принцип классификации оценки качества функционирования ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$.

Оценку качества функционирования каждого ОМУ, в зависимости от значений показателей производительности и потенциальной надежности, при трехпороговом и однопороговом вынесении решения можно разделить на классы и подклассы (табл. 5.2.). Если показатели надежности не определяются или не учитываются, то функционирование ОМУ оценивается только производительностью и кодируется одним символом A, B, C или D, а оценка качества функционирования ОМУ получается в результате сравнения фактической производительности с нормативной (4.10). В этом случае штатному функционированию, обозначаемому A или A0, соответствует, например, такой диапазон значений параметра текущей производительности: $0{,}9<s_1^{(i,j)}\leq 1$. В качестве параметра производительности могут быть использованы среднее время поиска по запросу, средняя загруженность каналов связи, время выполнения контрольных примеров и другие.

Таблица 5.2. Классы оценки качества функционирования ОМУ

Класс качества функционирования	Подкласс качества функционирования	Показатель производительности $s_1^{(i,j)}$	Показатель надежности $s_2^{(i,j)}$
A	A0	$h_{1,i,j}^{(3)} < s_1^{(i,j)} \le 1$	Не определяется
	A1	$h_{1,i,j}^{(3)} < s_1^{(i,j)} \le 1$	$h_{2,i,j}^{(3)} < s_2^{(i,j)} \le 1$
	A2	$h_{1,i,j}^{(3)} < s_1^{(i,j)} \le 1$	$h_{2,i,j}^{(2)} < s_2^{(i,j)} \le h_{2,i,j}^{(3)}$
	A3	$h_{1,i,j}^{(3)} < s_1^{(i,j)} \le 1$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le h_{2,i,j}^{(2)}$
	A4	$h_{1,i,j}^{(3)} < s_1^{(i,j)} \le 1$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$
B	B0	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	Не определяется
	B1	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	$h_{2,i,j}^{(3)} < s_2^{(i,j)} \le 1$
	B2	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	$h_{2,i,j}^{(2)} < s_2^{(i,j)} \le h_{2,i,j}^{(3)}$
	B3	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le h_{2,i,j}^{(2)}$
	B4	$h_{1,i,j}^{(2)} < s_1^{(i,j)} \le h_{1,i,j}^{(3)}$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$
C	C0	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	Не определяется
	C1	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	$h_{2,i,j}^{(3)} < s_2^{(i,j)} \le 1$
	C2	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	$h_{2,i,j}^{(2)} < s_2^{(i,j)} \le P_{2,i,j}^{(3)}$
	C3	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le P_{2,i,j}^{(2)}$
	C4	$h_{1,i,j}^{(1)} < s_1^{(i,j)} \le h_{1,i,j}^{(2)}$	$0 \le s_2^{(i,j)} \le P_{2,i,j}^{(1)}$
D	D0	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	Не определяется
	D1	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	$h_{2,i,j}^{(3)} < s_2^{(i,j)} \le 1$
	D2	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	$h_{2,i,j}^{(2)} < s_2^{(i,j)} \le h_{2,i,j}^{(3)}$
	D3	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	$h_{2,i,j}^{(1)} < s_2^{(i,j)} \le h_{2,i,j}^{(2)}$
	D4	$0 \le s_1^{(i,j)} \le h_{1,i,j}^{(1)}$	$0 \le s_2^{(i,j)} \le h_{2,i,j}^{(1)}$

Установка пороговых значений может осуществляться на основе практического опыта, когда значения порогов устанавливаются и изменяются администраторами вручную.

АУТСОРС

Таким образом, предложенные универсальная схема кодирования состояния ОМУ и принцип классификации оценки качества функционирования ОМУ делают качество функционирования величиной, допускающей измерение.

5.4. АНАЛИТИЧЕСКИЙ МЕТОД ОЦЕНКИ СОСТОЯНИЯ И КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ

Рассмотрим аналитический подход к определению качества функционирования ОМУ, который предполагает вычисление обобщенных параметров, характеризующих качество функционирования [104].

Результатом мониторинга ОМУ являются численные значения параметров, которые не обладают нужной для принятия решения администратором наглядностью. Поэтому целесообразно использовать понятие состояния ОМУ – качественной оценки, морфологической интерпретации численных значений параметров ОМУ. Администратору намного проще оценить функционирование конкретного ОМУ, когда он характеризуется не только численными значениями параметров, но и состоянием. Для ИТ УИТИ необходимо разработать методы оценки, которые на основе численных значений параметров ОМУ дают качественную оценку их текущего состояния.

Постановка задачи исследования. Пусть задано множество ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$. Для каждого ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, определено множество параметров $P^{(i,j)}$ (4.8). Известно влияние ниже расположенных ОМУ на выше расположенные и связи между ОМУ. Структура связей и степень влияния не изменяются во времени. Необходимо разработать метод определения интегрированных показателей оценки качества функционирования ОМУ.

Суть предлагаемого метода. При известной и фиксированной структуре связей между ОМУ целесообразно использовать аналитический метод оценки качества функционирования ОМУ, предполагающий введение единого показателя, вычисляемого на основе значений параметров $P^{(i,j)}$, что позволяет проецировать значения этого показателя на качественную ось.

Предлагаемый метод основан на вычислении обобщенных параметров, характеризующих качество функционирования ОМУ. Оценка качества функционирования ОМУ выполняется на основе анализа происходящих в нем процессов. С этой целью определяются параметры, характеризующие состояние и работу ОМУ, и их значения сравниваются с оценкой качества функционирования. На этой основе формируется характеристика, сводящая параметры, характеризующие аспекты функционирования ОМУ, в единый показатель. В качестве такой характеристики предлагается использовать интегральный показатель качества функционирования (ИПКФ) $E_{i,j}(t)\in[0,1]$, $i=\overline{1,I_j}$, $j=\overline{1,J}$ ОМУ.

Задача определения $E_{i,j}$ – численного значения ИПКФ ОМУ $O_{i,j}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$ сводится к определению функциональной зависимости вида

$$E_{i,j}=f(P^{(i,j)}). \qquad (5.14)$$

Текущее значение ИПКФ предлагается определять с помощью выражения

$$E_{i,j}(t) = \frac{[\sum_{l=1}^{n} k_l^{(i,j)} \cdot m_l^{(i,j)}(p_l^{(i,j)}(t),t)] \cdot \prod_{l=1}^{n} s_l^{(i,j)}(p_l^{(i,j)}(t),t)}{\sum_{l=1}^{n} k_l^{(i,j)}}, \qquad (5.15)$$

где $p_l^{(i,j)}(t)$ – значение параметра $p_l^{(i,j)}$, $l = \overline{1,n}$, в момент времени t; $k_l^{(i,j)}$ – весовой коэффициент l-го параметра; $m_l^{(i,j)}(p_l^{(i,j)}(t),t)$ – оценочная функция параметра $p_l^{(i,j)}$, $l = \overline{1,n}$; $s_l^{(i,j)}(p_l^{(i,j)}(t),t)$ – функция критичности параметра $p_l^{(i,j)}$, $l = \overline{1,n}$ ОМУ $O_{i,j}$, $i = \overline{1,I_j}$.

Значения интегрального показателя $E_{i,j}(t)$, принадлежат отрезку [0, 1], причем $E_{i,j}(t) = 1$ соответствует нормативному функционированию ОМУ $O_{i,j}$, а $E_{i,j}(t) = 0$ означает, что ОМУ $O_{i,j}$ неработоспособен.

Ключевыми понятиями подхода, использованными в выражении (5.15), являются оценочная функция $m_l^{(i,j)}(p_l^{(i,j)}(t),t)$ и функция критичности $s_l^{(i,j)}(p_l^{(i,j)}(t),t)$. Они вводятся для выделения особой роли некоторых параметров при оценке состояния ОМУ. Суть оценочной функции и функции критичности помогают выяснить следующие соображения.

Из множества $P_l^{(i,j)}$ выделяется подмножество критичных параметров, значения которых, принадлежащие некоторому диапазону, свидетельствуют о наличии проблем в ОМУ и снижении качества функционирования ОМУ. Для таких параметров функция критичности $s_l^{(i,j)}(p_l^{(i,j)}(t),t) \in [0,1)$. Для некритичных параметров $s_l^{(i,j)}(p_l^{(i,j)}(t),t) = 1$.

В функцию критичности $s_l^{(i,j)}(p_l^{(i,j)}(t),t)$ параметра $p_l^{(i,j)}$, $l = \overline{1,n}$, введена зависимость от времени для компенсации реакции ИТ УИТИ на запланированные работы по обслуживанию путем блокировки аварийных сигналов. Например, на серверах по расписанию могут запускаться антивирусные пакеты, проводиться резервное копирование и другие регламентные работы, изменяющие режим функционирования ОМУ. Кроме того, некоторые ОМУ могут не работать на протяжении заведомо известных интервалов времени, например рабочие станции ночью и в выходные дни. Учет этих событий в функции критичности позволяет блокировать в СУИ выдачу аварийных сигналов.

Параметры множества $p_l^{(i,j)}$ являются некритичными, если при их изменении во всем диапазоне возможных значений не происходит значительного изменения качества функционирования ОМУ. В формуле определения ИПКФ некритичные параметры учитываются в виде взвешенной суммы значений их оценочных функций. Каждый из таких параметров определяет качество функционирования ОМУ в определенном аспекте. Оценочная функция $m_l^{(i,j)}(p_l^{(i,j)}(t),t) \in [0,1]$ некритичного параметра зависит от текущего значения параметра и времени. Зависимость от времени введена из тех же соображений, что и для критичных параметров.

Значимость параметра $p_l^{(i,j)}$, $l=\overline{1,n}$, для определения состояния ОМУ определяется весовым коэффициентом $k_l^{(i,j)}$. Для ряда критичных параметров выполняется условие $k_l^{(i,j)} \equiv 0$, а расчет $m_l^{(i,j)}(p_l^{(i,j)}(t),t)$ не проводится.

Оценочную функцию $m_l^{(i,j)}(p_l^{(i,j)}(t),t)$ и функцию критичности $s_l^{(i,j)}(p_l^{(i,j)}(t),t)$ параметра $p_l^{(i,j)}$, $l=\overline{1,n}$, можно задавать аналитически или в виде таблиц. ИПКФ ОМУ $O_{i,j}$ используется для определения ИПКФ ОМУ $O_{i,j+1}$ выше расположенного уровня иерархии, поэтому он также нормируется делением на $\sum_{l=1}^{n} k_l^{(i,j)}$.

На каждом уровне входными данными для определения ИПКФ ОМУ $O_{i,j}$ являются ИПКФ ОМУ $O_{i,j-1}$ смежного нижнего уровня иерархии ОМУ и значения собственных параметров. При этом параметры могут иметь различный смысл и тип, а также быть получены разными способами: в результате тестирования или моделирования, прогнозирования, статистического анализа, вычислений на основе нечетких моделей, моделей искусственного интеллекта и т. д. Это позволяет использовать предлагаемый подход для анализа качества функционирования сложных, иерархических, гетерогенных ИТ-инфраструктур.

Обобщим выражение (5.15) для определения ИПКФ составного ОМУ $O_{l,j}$, которым может быть, например, функциональная подсистема, для определения ИПКФ которого необходимо учитывать не только ИПКФ составных частей, но и значения собственных параметров ОМУ.

Пусть на функционирование ОМУ $O_{l,j}$ оказывает влияние состояние k ОМУ $O_{l,j-1}$ и значения m собственных параметров. Тогда выражение для определения ИПКФ $E_{l,j}(t)$ составного ОМУ $O_{l,j}$ имеет вид

$$E_{l,j}(t)=\left[\frac{\sum_{i=1}^{k} K_i^{(l,j)} \cdot M_i^{(l,j)}(E_i^{(l,j)}(t),t)+\sum_{i=1}^{m} k_i^{(l,j)} \cdot m_i^{(l,j)}(p_i^{(l,j)}(t),t)}{\sum_{i=1}^{k} K_i^{(l,j)}+\sum_{i=1}^{m} k_i^{(l,j)}}\right] \times \tag{5.16}$$

$$\times \prod_{i=1}^{k} S_i^{(l,j)}(E_i^{(l,j)}(t),t) \cdot \prod_{i=1}^{m} s_i^{(l,j)}(p_i^{(l,j)}(t),t),$$

где $p_i^{(l,j)}(t)$ – значение i-го параметра ОМУ $O_{l,j}$ в момент времени t, $k_i^{(l,j)}$ – весовой коэффициент для i-го параметра, $m_i^{(l,j)}(p_i(t),t)$ – оценочная функция i-го параметра, $s_i^{(l,j)}(p_i(t),t)$ – функция критичности i-го параметра, $E_i^{(l,j)}(t)$ – значение ИПКФ i-го подэлемента в момент времени t, $K_i^{(l,j)}$ – весовой коэффициент i-го подэлемента, $M_i^{(l,j)}(p_i(t),t)$ – оценочная функция i-го подэлемента, $S_i^{(l,j)}(p_i(t),t)$ – функция критичности i-го подэлемента.

Применение предлагаемого метода для оценки качества функционирования простого объекта. В качестве примера рассмотрим набор параметров,

Аутсорсинг

функций и коэффициентов для определения качества функционирования рабочей станции или ПК, которая рассматривается как простой ОМУ, а качество ее функционирования определяется только на основе значений параметров. Рабочая станция является совокупностью взаимодействующих аппаратных и программных ОМУ, а функционирование программных средств влияет на значения многих аппаратных параметров. Однако для простоты изложения ограничимся рассмотрением только аппаратных параметров, анализ значений которых позволяет определить, насколько аппаратное обеспечение справляется с нагрузкой, создаваемой совокупностью выполняемых системных и прикладных программ.

В первую очередь, необходимо выбрать множество параметров и определить для каждого из них интервал времени Δt_i, через который проводятся измерения мгновенных значений, определить k_i, вид функциональной зависимости оценочной функции и функции критичности. Интервалы и форма зависимостей определяются, исходя из сущности параметра и его влияния на функционирование системы.

Для определения ИПКФ необходимо отобрать наиболее важные параметры ОМУ, причем количество параметров должно быть не большим, чтобы не затруднять расчет ИПКФ, и не малым, чтобы учесть специфику работы ОМУ.

В качестве набора параметров выберем: p_1 – средняя загруженность процессора за период; p_2 – использование ОЗУ; p_3 – использование дискового пространства; p_4 – загруженность диска – относительное количество времени, в течение которого жесткий диск занят обработкой запросов ввода/вывода.

Что касается выбора значений Δt_i и T_i, то в рассматриваемом случае эти значения одинаковы для всех параметров, а в целях упрощения изложения весовые коэффициенты k_i примем равными 1 для всех значений i.

Средняя загруженность процессора за период T_1. Этот параметр является простым вычислимым параметром. Для его расчета используется параметр «мгновенное значение загруженности процессора» p_1', замеряемый на протяжении периода T_1 с интервалом Δt_i. Значение параметра рассчитывается по формуле

$$p_1(t) = \frac{1}{N_1} \sum_{i=0}^{N_1-1} p_1'(t - i \cdot \Delta t_1), \tag{5.17}$$

где $N_1 = \dfrac{T_1}{\Delta t_1}$, причем $p_1(t) \in [0.1]$.

Данный параметр считается некритичным, поскольку рабочая станция может выполнять возложенные на нее функции даже при полной загруженности процессора в течение длительных временных интервалов, поэтому $s_1 = 1$, а форма функции критичности приведена на рис. 5.4,*б*.

При выборе оценочной функции (рис. 5.4,*а*) учитывается, что чем больше значение параметра, тем хуже функционирует рабочая станция.

Для того чтобы не усложнять пример, не будем учитывать временные зависимости оценочной функции и функции критичности и использовать простейшие кусочно-линейные функции.

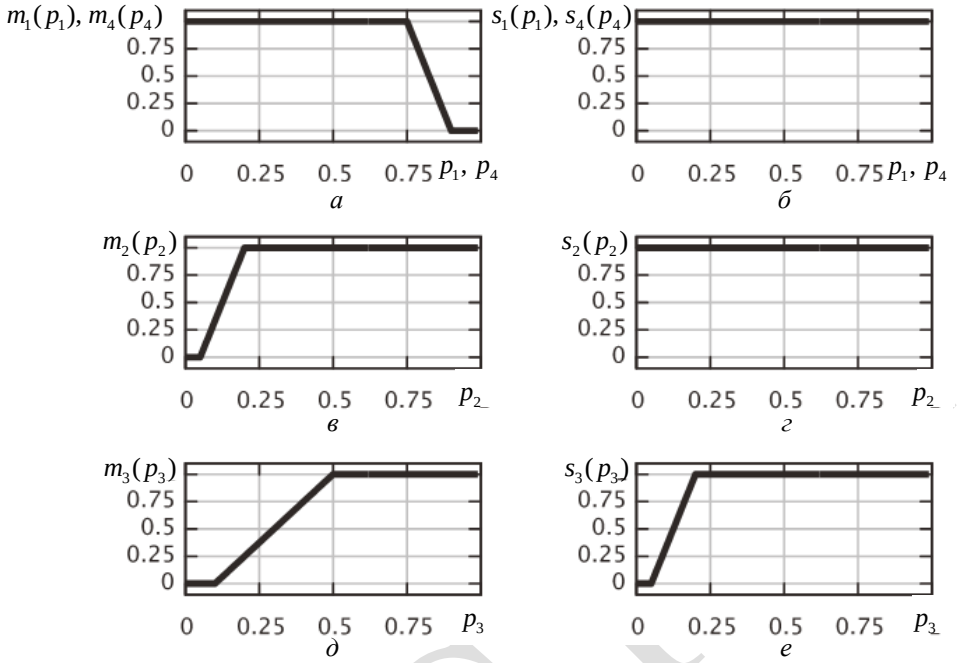


Рис. 5.4. Оценочные функции и функции критичности для параметров p_1–p_4

Использование памяти. Данный параметр является вычислимым и сложным, а его значение определяется по формуле

$$p_2(t) = \frac{M(t)}{M}, \tag{5.18}$$

где $M(t)$ – свободный объем физической памяти в текущий момент времени в байтах, M – объем физической памяти рабочей станции в байтах.

Значения параметра $p_2(t) \in [0,1]$. Данный параметр является некритичным: $s_2 = 1$, а вид зависимости приведен на рис. 5.4,*г*.

На рис. 5.4,*в* представлена форма оценочной функции.

Использование дискового пространства. Параметр является сложным и вычисляется по формуле

$$p_3(t) = \frac{D(t)}{D}, \tag{5.19}$$

где $D(t)$ – свободный объем дисковой памяти в текущий момент времени в байтах, D – объем физического диска в байтах.

Параметр p_3 – критичный. Форма функции критичности приведена на рис. 5.4,*д*, а на рис. 5.4,*е* представлена выбранная форма оценочной функции.

Загруженность диска за период T_4. Этот параметр является простым вычислимым параметром. Для его расчета используется параметр «мгновенное значение загруженности диска» $p'_4 \in [0,1]$, замеряемый на протяжении периода T_4 с интервалом Δt_4. Значение параметра рассчитывается по формуле

$$p_4(t) = \frac{1}{N_4} \sum_{i=0}^{N_4-1} p'_4(t - i \cdot \Delta t_4), \tag{5.20}$$

где $N_4 = \frac{T_4}{\Delta t_4}$. Причем $p_4(t) \in [0,1]$.

Данный параметр является некритичным: $s_4 = 1$. Форма функции критичности для этого параметра приведена на рис. 5.4,*б*.

При выборе оценочной функции учитывается, что чем больше значение параметра, тем хуже функционирует рабочая станция; форма оценочной функции параметра приведена на рис. 5.4,*а*.

Предложенная модель использована для исследования функционирования реального ПК. При моделировании значения весовых коэффициентов k_1, k_2, k_3, k_4 были взяты равными 1; интервал Δt_i для всех параметров равен 1 мин; продолжительность моделирования – 120 мин; значения параметров p_1 и p_4 усреднялись по последним 10 значениям (T_i = 10 мин); в качестве оценочных функций и функций критичности использованы зависимости, приведенные на рис. 5.4,*а–е*; вычисление ИПКФ проводилось каждую минуту.

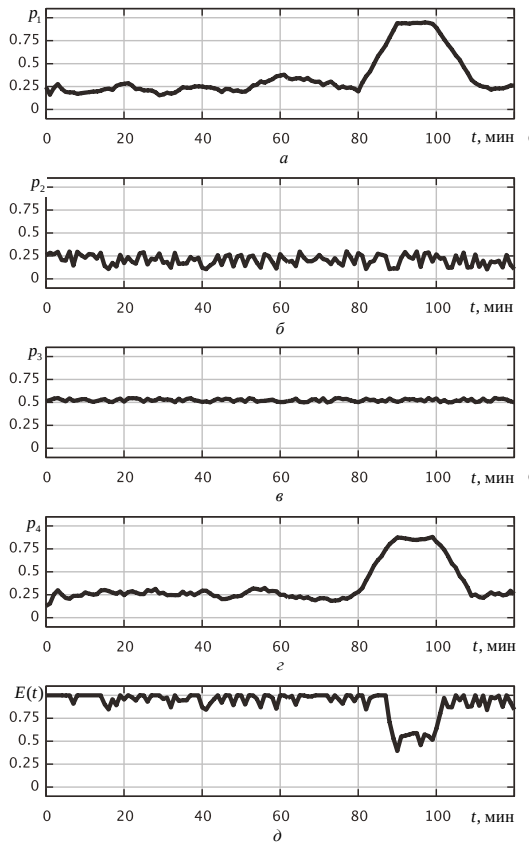


Рис. 5.5. Результаты моделирования рабочей станции

Результаты моделирования приведены на рис. 5.5. На рис. 5.5,*а–г* показано изменение значений параметров p_1, p_2, p_3, p_4 во времени, а на рис. 5.5,*д* – изменение значения ИПКФ. Снижение ИПКФ в диапазоне 80–100 мин обусловлено увеличением загруженности процессора и дисковой подсистемы, вызванное запуском приложения, потребляющего значительные ресурсы.

Таким образом, предложен аналитический метод оценки состояния ОМУ по интегральному показателю качества функционирования. Оценка состояния ОМУ определяется с помощью аналитических методов сведения метрик, однако расчеты требуют длительного калибровки схемы оценивания.

5.5. СВЕДЕНИЕ МЕТРИК ОЦЕНИВАНИЯ УРОВНЯ ОБСЛУЖИВАНИЯ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ ЭКСПЕРТНЫХ ОЦЕНОК

5.5.1. Оценка уровня обслуживания пользователей

Внутренний контур управления охватывает среди прочих процессы определения рационального уровня услуг и обеспечения предоставления сервисов на уровне, зафиксированном в SLA. Для поддержки этих процессов нужны модели и методы анализа качества обслуживания пользователей, результаты которых будут использоваться при оптимизации использования ресурсов. ИТ УИТИ должна обеспечить бизнес-процессы планирования и проектирования сервисов информацией для их эффективного осуществления по стоимостным критериями с учетом временных ограничений и регламента. После нотификации сервис поддерживается процессами оперативного контура управления.

Для определения управляющих решений по поддержанию согласованного уровня услуг необходимо собирать значения метрик, осуществлять их обработку, выполнять анализ корреляции и фильтрацию на всех уровнях иерархии ИТ-инфраструктуры, а также сводить метрики уровня элементов ИТ-инфраструктуры до метрик уровня услуг. Для этого нужно иметь соответствующие механизмы и эффективно их использовать при функционировании СУИ.

Оценка состояния таких сложных многопараметрических систем, как сети крупных ОТС, требует учета многих факторов. В то же время в СУ ОТС оценка состояния сети и ее составляющих, которая предоставляется техническим службам и менеджерам соответствующих уровней для реагирования и принятия решений, должна быть краткой и информативной. Возникает необходимость разработать подход к оценке состояния сетей ОТС на основе сложной системы параметров и простых количественных общих показателей, чему и посвящен данный параграф.

Решение подобных задач для автоматизированного принятия решений, управления и моделирования в сложных системах ОТС на верхних уровнях связано с необходимостью учета доминирующего характера качественных, неопределенных и нечетких факторов. Среди методов учета таких факторов широкое распространение получили подходы на основе теории нечетких множеств и нечеткой логики Л. Заде [390]. Однако для принятия качественных решений в упомянутых реальных задачах не найдено лучшего подхода, чем использование опыта, знаний и интуиции специалистов-экспертов. Поэтому для решения практических задач вполне оправданным является комбинированное применение методов экспертных оценок и подходов на основе теории нечетких множеств [55].

Задачи многокритериального оценивания и выбора решений из множества альтернативных вариантов в СУ ОТС становятся актуальными ввиду перемещения акцента с технических проблем на вопросы бизнеса [347, 348, 349]. Особое значение приобретают задачи оценки уровня качества сервисов, предоставляемых пользователям, и уровня обслуживания пользователей ОТС в целом. Создание эффективных СУИ ОТС невозможно без решения указанных задач [148, 154].

В [145] предложен подход к оценке уровня обслуживания пользователей ОТС на основе комбинированного применения методов экспертных оценок и подходов на основе теории нечетких множеств. Суть предлагаемого подхода состоит в следующем.

С точки зрения теории управления сети ОТС представляют собой сложные системы с четко выраженным функциональным назначением. Критерии управления такими системами должны концентрироваться на эффективности выполнения системами своих функций, которых обычно несколько. К тому же с целью упрощения управления этими системами они рассматриваются на нескольких взаимосвязанных уровнях представления, соответствующих уровням ИТ-инфраструктуры (см. рис. 1.2). Поэтому речь должна идти о системе векторов эффективностей выполнения отдельных функций, объединенных межуровневыми связями.

Особенностью ведения бизнеса ОТС является наличие мощной ИТ-инфраструктуры, обеспечивающей высокое качество информационно-коммуникационных сервисов, как решающего фактора успеха бизнеса [347, 348, 349]. В этих условиях TMF предложила новую модель организации деятельности ОТС в области телекоммуникаций – eTOM. Модель является процессно-ориентированной и нацеливает деятельность ОТС на потребности пользователя. В нее включены процессы, необходимые для управления уровнем обслуживания пользователей и качеством сервисов. Осуществление этих процессов позволяет определить характеристики доступности, надежности и качества выполнения сервисов, предоставляемых пользователю. Договор SLA фиксирует эти параметры, чтобы обеспечить условия для доставки услуг пользователям с согласованным уровнем защиты.

Сложность связей и многоуровневость сетей приводит к существованию $n \cdot 1000$ различных параметров и метрик для измерения и оценки состояния сетей, их составляющих и уровня обслуживания пользователей. Метрики уровня пользователей часто носят качественный характер, характеризуются неопределенностью, нечеткостью. К тому же они носят субъективный характер. Поэтому для формирования сводных метрик необходимы соответствующие методы. Кроме того, многоуровневость представления сетей определяет наличие иерархических связей между метриками разных уровней. При оценке удовлетворенности пользователя уровнем обслуживания ОТС эту сложную задачу можно разбить на каждом иерархическом уровне по схеме, приведенной на рис. 5.6.

Параметры обслуживания различных сервисов ОТС являются схожими и могут быть представлены, например, с помощью следующих метрик: удобство пользования, техническая поддержка, стабильность и непрерывность обслуживания, время определения причины неисправности, о которой сообщил клиент, время устранения неисправности, о которой сообщил клиент.

Остановимся на сервисах, в которых пользователи нуждаются прежде всего: доступ к интернету и VoIP. Система оценки этих сервисов приведена на рис. 5.7 и рис. 5.8 соответственно.

Одновременный учет технических параметров, предоставляемых зачастую в цифровом виде с возможностью вычисления их значений на основе аналитических зависимостей, в сочетании с лингвистическими переменными, которые часто представляют параметры обслуживания и ценовые параметры, дает возможность с помощью метода нечеткого логического вывода получить интегральную характеристику удовлетворенности пользователя уровнем обслуживания. При этом собственно сведение параметров целесообразно основывать на экспертных оценках, поскольку оценки пользователей имеют субъективный характер.

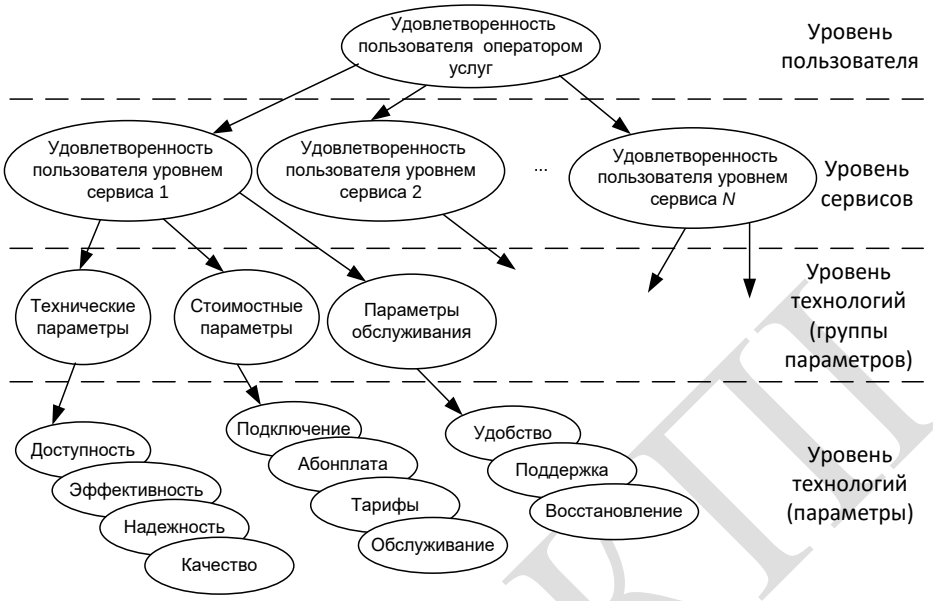


Рис. 5.6. Система метрик оценивания удовлетворенности пользователей

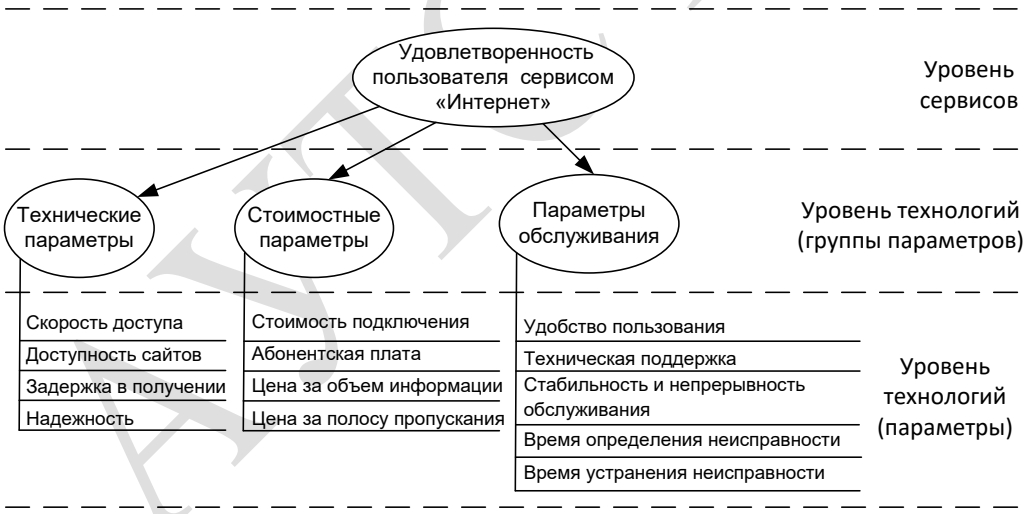


Рис. 5.7. Система оценок сервиса «Интернет»

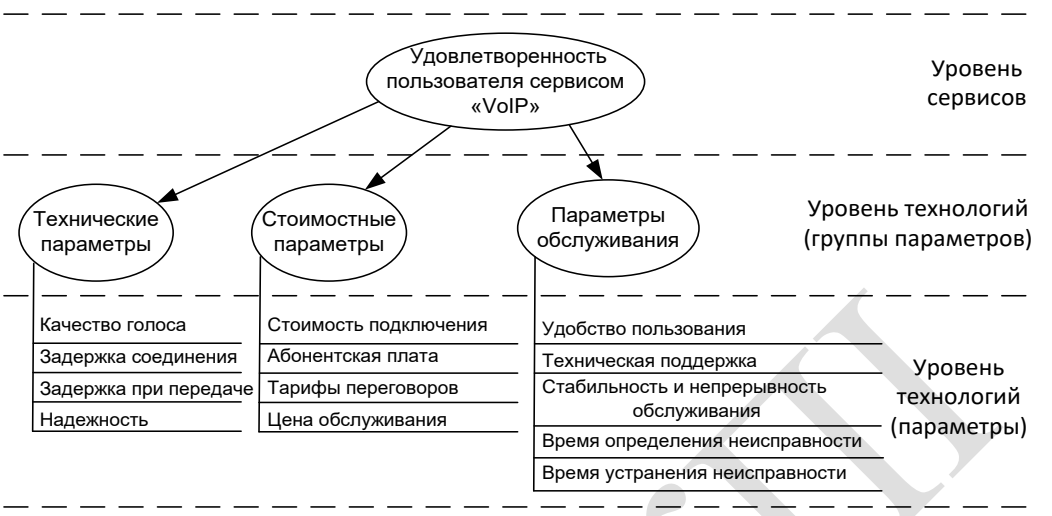


Рис. 5.8. Система оценок сервиса VoIP

Количество метрик QPI и KPI уменьшается при переходе от нижних иерархических уровней к верхним [197], при этом на нижних уровнях используются конкретные, а на верхних – обобщенные метрики. Метрики должны быть простыми, непротиворечивыми, измеримыми и конкретными [10, 197]. Конструирование метрик проводится сверху вниз, а интеграция метрик осуществляется снизу вверх [197].

В работе [10] изложены принципы проектирования и реализации метрик, используемых в операционной деятельности ИТ-подразделения, но принципы проектирования метрик оперативного управления не рассмотрены. Для уровня сетевого взаимодействия метрики четко определены [308, 283, 176, 177, 178], при этом стандартизованы методы проведения испытаний при определении значений этих метрик [198, 288, 201]. Однако для технологий выше расположенных уровней пока не существует стандартного набора метрик и методов их вычисления. В работах [197, 385, 385] рассмотрены вопросы корреляции и сведения метрик. Получение метрик нижних уровней не составляет труда, поскольку здесь используются простейшие операции, например, суммирования или взвешенного суммирования [130], усреднения событий за период времени и пр. Метод взвешенной суммы с учетом целей и приоритетов [10] позволяет учесть изменения требований бизнеса путем коррекции приоритетов метрик. Но сведению метрик на верхних уровнях ИТ-инфраструктуры свойственны определенные сложности, вызванные влиянием субъективизма, неопределенности и возрастания значимости для принятия решений. В работе [194] представлена методика сведения метрик на основе древовидного представления процессов и классификации данных мониторинга. В работе [300] предложена модель взвешенного сведения метрик. В работе [350] данные мониторинга проецируются на шкалу единого масштаба, но используются строго определенные уровни сводных метрик, что требует длительного налаживания механизма оценки. В работе [89] предлагаются методы сведения метрик с применением факторного анализа, однако затруднено их содержательное толкование.

Усилия TMForum (международная ассоциация ОТС) в OSS сосредоточены на аспектах обслуживания сети и решении бизнес-проблем. Поскольку в фокусе внимания находится методология структурирования и описания бизнес-процессов обслуживания сети, моделирование метрик и анализ их корреляции не исследованы. Стандарты ITIL, ISO 20000, COBIT, Six Sigma и eTom и другие рассматривают метрики в качестве аппарата контролирующего управления ИТ-услугами [10], а определение значений метрик преимущественно основывается на подсчете событий, происходящих в ИУС [10, 180, 206, 260, 318, 336, 338, 339]. Поскольку эти стандарты основаны на процессном подходе, в них отсутствуют метрики для оперативного контура и соответствующие методы сведения метрик. Поэтому возникает достаточно сложная задача разработки методов сведения значений показателей функционирования ОМУ на основе иерархического представления ИТ-инфраструктуры к обобщенным метрикам, комплексно описывающим качество выполнения возложенных на ИТ УИТИ задач.

В ИТ УИТИ оценка состояния ИТ-инфраструктуры и ее элементов должна быть краткой и информативной. Возникает необходимость в разработке метода оценки состояния ИТ-инфраструктуры на основе сложной системы параметров и простых количественных общих показателей. Предлагается метод оценки уровня обслуживания на основе комбинированного применения методов экспертных оценок и подходов на основе теории нечетких множеств. Суть предлагаемого метода заключается в одновременном учете значений технических параметров, вычисленных на основе аналитических зависимостей, и лингвистических переменных параметров обслуживания. Это дает возможность с помощью нечеткого логического вывода получить интегральную характеристику удовлетворенности пользователя уровнем услуг. При этом сведение параметров основывается на экспертных оценках.

5.5.2. Общая постановка задачи сведения метрик качества сервиса при нечетком оценивании

Осознав важность комплексной оценки сервисов сетей ОТС, разработчики средств автоматизации предложили несколько решений, (см., например, [197]). Сложность проблем оценки обусловлена размерами сетей и организационных структур поддержки деятельности ОТС, необходимостью увязать в единой СУ множества факторов, в том числе тех, которые ранее не учитывались, многокритериальностью, неопределенностью и рисками, присущими автоматизируемым процессам деятельности ОТС. Смещение акцентов от операционного управления сетью в сторону бизнеса требует построения новых зависимостей между состоянием сети и ее элементов и бизнес-метриками.

Выделение в больших сетях четырех основных уровней – ресурсов, сетей, сервисов и пользователей привело к построению на каждом из них своей системы оценки качества. Отдельные технологические показатели технологий и оборудования последовательно сводятся в релевантные показатели функционирования технологий и оборудования – KPI. На основе KPI определяются релевантные уровня сервисов показатели качества – KQI. Они обеспечивают измерение специфических аспектов функционирования продуктов, сервисов и их элементов на основе KPI, а также информации из других источников. На основе сервисных KQI определяются обобщенные KQI для оценки предлагаемых пользователю

услуг. Значения KQI часто включаются в SLA, а на уровне пользователей применяются также специфические для пользователей показатели QoS, которые позволяют оценить выполнение услуги. Специфика системы метрик состоит в том, что метрики высших уровней, которые исчисляются на основе метрик нижних уровней, уже дают обобщенные значения, которые скрывают влияние отдельных метрик нижнего уровня. Так, все метрики, начиная с уровня сервисов, являются технологически нейтральными.

В принципе существуют хорошо продуманные подходы к комплексной оценке метрик нижних уровней, например [308, 283], но проблема разработки моделей и методов сведения метрик высших уровней, особенно с учетом аспектов бизнеса ОТС, в целом еще не решена. Появились интересные идеи по управлению параметрами SLA с учетом бизнес-интересов [190, 203], но не учтен весь спектр факторов влияния, особенности для новых видов ОТС, сервисов и пользователей.

Стоит отметить, что чем выше уровень анализа сетей ОТС, тем более неопределенность влияет на оценку его показателей. Поэтому представляется оправданным применение теории нечетких множеств и нечеткой логики, что позволит отказаться от предположения о стационарности потоков и расширить выбор законов распределения времени поступления заявок и их обслуживание. Так, общую оценку удовлетворенности пользователя сервисом или общую оценку сервиса очень удобно моделировать в терминах нечетких множеств, а агрегирование показателей осуществлять с помощью правил нечеткой логики. В условиях нечеткого описания системы и/или нечетко сформулированных критериев эти методы можно эффективно применять для оптимизации параметров сетей и управления ими, поскольку они позволяют более адекватно представить процессы в сетях. Моделирование влияния субъективных предпочтений пользователей по оценке сервисов и продуктов наилучшим образом можно осуществить с помощью экспертов.

Сложность проблем оценки обусловлена размерами ИТ-инфраструктуры, необходимостью увязать в единой ИТ УИТИ множество факторов, многокритериальностью, неопределенностью и рисками, присущими процессам управления ИТ-инфраструктурой. Смещение акцентов от управления ИТ-инфраструктурой в сторону бизнеса требует определения зависимостей между состоянием ИТ-инфраструктуры и бизнес-метриками.

Выделение в ИТ-инфраструктуре иерархических уровней привело к построению на каждом из них своей системы оценки качества. Отдельные технологические показатели технологий и оборудования последовательно сводятся к релевантным показателям функционирования – KPI и показателям качества сервисов – KQI.

Задачу сведения метрик на основе группового экспертного оценивания можно сформулировать следующим образом. Пусть имеется множество предложенных метрик $M = \{M_i\}$, $i = \overline{1,n}$, и экспертный комитет – $E = \{E_t\}$, $t = \overline{1,k}$. Необходимо сформировать сведенную метрику $M_{\text{св}}$, которая наилучшим образом охарактеризует уровень услуги с точки зрения заданной цели управления сетью.

Для решения подобных задач экспертным комитетом E обычно определяется набор важных критериев $C = \{C_c\}$, $c = \overline{1,h}$, с которыми предложенные метрики множества M будут входить в сведенную метрику $M_{\text{св}}$ [56]. Критерии множества

C, по которым эксперты будут оценивать вклад метрик в формирование $M_{св}$, могут отличаться по своей важности и в результате оказывать разное влияние на решение задачи, причем каждый эксперт может иметь свое мнение по ранжированию и распределению веса каждого критерия. Поэтому целесообразно, чтобы каждый эксперт E_t давал свои индивидуальные весовые коэффициенты W_{ct} для каждого критерия C_c, $c=\overline{1,h}$, $t=\overline{1,k}$. К тому же представляется целесообразным вести учет значимости, опыта, уровня подготовки, должности и других характеристик экспертов. Для этого экспертным комитетом E, коллегиально или его председателем задаются весовые коэффициенты V_t оценок экспертов E_t, $t=\overline{1,k}$.

Затем метрикам M_i, $i=\overline{1,n}$, каждым экспертом E_t, $t=\overline{1,k}$, по каждому критерию C_c, $c=\overline{1,h}$, предоставляются отдельные субъективные экспертные оценки S_{ict}. Отметим, что преимущество по одному из критериев не зависит от того, какие значения принимают оценки определенной метрики по другим критериям [56].

В практических задачах приемлемым оказывается использование процедуры сведения задачи многокритериального выбора к задаче однокритериального выбора. Целесообразно оценку вклада метрик проводить по обобщенной агрегированной оценке каждой метрики M_i, $i=\overline{1,n}$, – степени важности A_i, $i=\overline{1,n}$, метрики для сводной метрики $M_{св}$ [56].

Для нахождения обобщенных агрегированных метрик в условиях неопределенности, характерной сведению метрик на высших уровнях, целесообразно применять системы нечеткого логического вывода (СНЛВ). СНЛВ для получения обобщенных оценок метрик имеет h входов и один выход – обобщенную оценку метрики и используется для каждой метрики M_i, $i=\overline{1,n}$, и для каждого эксперта E_t, $t=\overline{1,k}$, отдельно. СНЛВ отрабатывает $n{\cdot}k$ циклов, на каждом из которых на вход СНЛВ подают оценки S_{ict} по критериям C_c, $c=\overline{1,h}$, метрики M_i от эксперта E_t, а на выходе получают обобщенную четкую оценку метрики M_i экспертом E_t.

Затем на основе обобщенных по критериям оценок A_{it}^{E} и весовых коэффициентов экспертов V_t, $i=\overline{1,n}$, $t=\overline{1,k}$, осуществляется операция нахождения агрегированной оценки A_i каждой метрики M_i, $i=\overline{1,n}$, – степени важности ее вклада в сведенную метрику $M_{св}$. Для расчета таких агрегированных оценок обычно используют взвешенную сумму:

$$A_i=\frac{\sum_{t=1}^{k}V_t\cdot A_{it}^{E}}{\sum_{t=1}^{k}V_t}. \qquad (5.21)$$

Затем сведенную метрику $M_{св}$ получают с помощью выражения

$$\sum_{i=1}^{n}A_iZ_i, \qquad (5.22)$$

где Z_i – нормированное значение метрики M_i, $i=\overline{1,n}$.

Диапазоны изменения значений метрик $z_{i\min} - z_{i\max}$ могут отличаться один от другого как величиной, так и знаком. В этом случае нормированное значение чаще всего приводят к диапазону 0–100 по формуле

$$Z_i = \frac{100 z_i}{z_{i\max} - z_{i\min}}, \tag{5.23}$$

где z_i – конкретное значение метрики M_i.

СНЛВ – это удобный механизм решения подобных задач, который обеспечивает прозрачность процедур, легкость их корректировки, дает возможность учитывать количественные и качественные метрики. Однако эти системы имеют существенный недостаток, который заключается в большой трудоемкости построения базы правил. Действительно, поскольку каждый член экспертного комитета может иметь свое убеждение о важности того или иного критерия, то для повышения качества решения целесообразно для оценок каждого эксперта E_t при отработке указанных выше циклов применять его индивидуальное мнение относительно уровня важности критериев, которое выражается в весовых коэффициентах критериев W_{lt}, где $l = \overline{1,h}$, $t = \overline{1,k}$. Однако структура правил нечеткого логического вывода обуславливает учет важности критериев в самих правилах, т. е. причинно-следственные связи между входами и выходом, которые сформулированы в том или ином правиле, по своему содержанию и сути определяют зависимость величины выхода от входных значений и взаимного относительного уровня важности входных значений. Поэтому для учета индивидуальных предпочтений каждого эксперта по степени важности оценок по критериям наиболее корректным оказывается создание и использование индивидуальных баз правил для каждого эксперта [197]. Понятно, что такой подход в реальных задачах сведения метрик оказывается слишком дорогим и исключает использование СНЛВ и их преимуществ. Поэтому необходимо разработать подход к применению СНЛВ, который обеспечит учет индивидуальных предпочтений каждого эксперта по степени важности критериев оценки и будет пригодным для применения в практических задачах.

Для решения задачи предлагается применять единую для экспертов СНЛВ со взвешенной истинностью, разработанную в [57] и апробированную в [285].

Постановка задачи. К системе метрик для измерения и оценки сетей ОТС, их составляющих, качества сервисов и продуктов принадлежат количественные и качественные метрики. Использование оценок для управления сетями ОТС на практике требует методик сведения количественных и качественных метрик в сведенные метрики того же или более высоких уровней иерархии, которые бы обеспечивали получение результатов за приемлемое время. Необходимо разработать подход к применению СНЛВ, который обеспечит учет индивидуальных предпочтений каждого эксперта по степени важности критериев оценки и будет пригодным для применения в практических задачах.

Общий подход к оцениванию. Для практического оценивания сетей ОТС, их составляющих, качества сервисов и продуктов с целью управления этими сетями на основе нечеткого логического вывода с привлечением экспертного комитета предлагается применять единую для всех экспертов СНЛВ с взвешенной истинностью, разработанную в работе [57] и успешно апробированную в работе [285] для задач выбора проектов и соответствующего финансирования.

Система нечеткого логического вывода. СНЛВ традиционно включает следующие компоненты: фаззификатор; механизм логического вывода; дефаззификатор. Фаззификатор определяет степень принадлежности входных значений x_c, $c=\overline{1,h}$, к нечетким множествам входа – лингвистическим переменным с соответствующей лингвистической шкалой $T_{x_c}=\left\{T_{x_c}^1,\ T_{x_c}^2,\ ...,\ T_{x_c}^{m_{xc}}\right\}$, где m_{xc} – количество лингвистических переменных в шкале для *c*-го входа, которая является одинаковой для всех входов. Необходимость фаззификации обусловлена использованием в СНЛВ лингвистических правил, поскольку необходимо определить степень истинности каждой предпосылки каждого правила.

Ядро механизма логического вывода составляет база лингвистических правил *Rules* = {*Rule*$_j$}, $j=\overline{1,r}$, имеющих вид [55]

$$Rule_j = \text{«если } x_1 \in T_{x_1}^{any} \text{ и } x_2 \in T_{x_2}^{any} \text{ ... и } x_h \in T_{x_h}^{any}, \text{то } y^j \in T_y^{any}\text{»}. \qquad (5.24)$$

Правила *Rule*$_j$ могут быть заданы экспертами или получены на основе накопленных статистических данных. Механизм логического вывода отображает нечеткие множества входа $T_{x_c}^{any}$, $c=\overline{1,h}$, каждого правила в нечеткие множества выхода T_y^{any}, определенные для выходных лингвистических переменных $\mathrm{T}_y=\left\{T_y^1,\ T_y^2,\ ...,\ T_y^{m_y}\right\}$. Порядок выполнения правил не влияет на результат – правила анализируются и выполняются параллельно. Выходные нечеткие множества y^j каждого правила объединяются в одно нечеткое множество вывода $\tilde{y}$.

Дефаззификатор отражает нечеткое множество вывода $\tilde{y}$ в четкое число $\bar{y}$, которое можно считать результатом работы СНЛВ для входных значений x_c, $c=\overline{1,h}$. Таким образом, обработка входных данных позволяет вместо множества выходных значений получить одно удобное для дальнейшего использования числовое значение. На практике используется высотная дефаззификация, а также дефаззификация на основе методов центроидов, максимума, центра максимумов [55].

Пусть база *Rules* правил вида (5.22) построена. Процесс функционирования СНЛВ состоит в последовательном подключении ее компонентов. На этапе фаззификации определяются степени принадлежности входных значений x_c, $c=\overline{1,h}$, нечетким множествам входа, т. е. определяются степени истинности $\mu_c^j(x_c)$ для каждой предпосылки каждого правила *Rule*$_j$. Затем для каждого правила на основе степени истинности предпосылок μ_c^j определяется степень выполнения α_j. Для этого применяют композицию на основе оператора минимума:

$$\alpha_j = \min\left(\mu_1^j(x_1),\ \mu_2^j(x_2),\ ...,\ \mu_n^j(x_n)\right),\ j=\overline{1,r}. \qquad (5.25)$$

Для каждого правила на основе степени выполнения α_j, $j=\overline{1,r}$, рассчитывается результат его выполнения – выходное нечеткое множество с усеченной

АУТСКИ

функцией принадлежности $\ddot{\mu}^j(y)$. Определение усеченной функции принадлежности, т. е. операция импликации, осуществляется с помощью оператора минимума (по Мамдани [285]):

$$\ddot{\mu}^j(y) = \min\left(\alpha_j, \mu^j(y)\right),\ j = \overline{1,r}, \tag{5.26}$$

или на основе оператора произведения (по Ларсену [56]):

$$\ddot{\mu}^j(y) = \alpha_j \cdot \mu^j(y),\ j = \overline{1,r}. \tag{5.27}$$

Работа механизма логического вывода завершается тем, что выходные нечеткие множества выполненных правил с помощью оператора максимума агрегируются в нечеткое множество выхода $\tilde{y}$, функция принадлежности которого имеет вид

$$\mu_{\tilde{y}} = \max\left(\ddot{\mu}^1(y), \ddot{\mu}^2(y), ..., \ddot{\mu}^r(y)\right). \tag{5.28}$$

Метод дефаззификации нечеткого множества, который применяется для нахождения окончательного четкого результата $\bar{y}$, выбирается в зависимости от конкретной задачи. Для оценивания сетей ОТС очень удобным и эффективным оказался центроидный метод. Он заключается в нахождении центра тяжести (центроида) нечеткого множества $\tilde{A}$, который и выбирается в качестве результата *A*:

$$A = \frac{\int_{C_1}^{C_n} x f_{\tilde{A}}(x)\,dx}{\int_{C_1}^{C_n} f_{\tilde{A}}(x)\,dx}. \tag{5.29}$$

В популярной СНЛВ Сугено база правил *Rules* = {*Rule*$_j$}, $j = \overline{1,r}$, состоит из правил, выходом которых является четкое значение, определяемое линейной функцией:

$$Rule_j = \text{«если } x_1 \in T_{x_1}^{any} \text{ и } x_2 \in T_{x_2}^{any} \text{ ... и } x_n \in T_{x_n}^{any}, \text{ то } \overline{y^j} = \sum_{i=1}^{n} a_i^j x_i + c^j \text{»}, \tag{5.30}$$

где a_i^j, $i = \overline{1,n}$, c^j – константы [359].

Окончательный результат логического вывода $\bar{y}$ рассчитывается как взвешенное среднее четких $\overline{y^j}$ результатов выполненных правил, в котором весовыми коэффициентами выступают степени выполнения правил α_j. Однако обычная СНЛВ не учитывает важные для практики ограничения на время получения нужных оценок, необходимых для управления сетью. Поэтому для оценивания сетей ОТС часто применяются СНЛВ с взвешенной истинностью.

АУТСКИ

Система нечеткого логического вывода со взвешенной истинностью. Опорная база правил такой СНЛВ является единой для всего комитета и создается консолидировано группой $E = \{E_t\},\ t = \overline{1,k}$. Важность $W_l = 1/h$ для всех критериев $C = \{C_l\},\ l = \overline{1,h}$, при составлении правил считается равной. При применении СНЛВ со взвешенной истинностью для каждого эксперта $E = \{E_t\},\ t = \overline{1,k}$, проводят корректировку процесса логического вывода с учетом индивидуальных значений весовых коэффициентов W_{ct} критериев $C_c,\ c = \overline{1,h}$.

Конечно, весовые коэффициенты W_{ct} являются нормированными, и в таких случаях целесообразно их использовать как корректирующие множители. В СНЛВ, использующих такой метод, степень истинности каждой предпосылки μ_c^j, что соответствует критерию C_c правила $Rule_j$, умножается на соответствующий весовой коэффициент W_{ct}. Тогда с учетом (5.25) степень выполнения α_j правила $Rule_j$ для эксперта E_t определяется так:

$$\alpha_j = \min\left(W_{1t} \cdot \mu_1^j(x_1),\ W_{2t} \cdot \mu_2^j(x_2), ...,\ W_{ht} \cdot \mu_h^j(x_h)\right). \qquad (5.31)$$

Для экспертных оценок метрик соответственно можно использовать выражение

$$\alpha_j = \min\left(W_{1t} \cdot \mu_1^j(S_{ict}),\ W_{2t} \cdot \mu_2^j(S_{ict}), ...,\ W_{ht} \cdot \mu_h^j(S_{ict})\right). \qquad (5.32)$$

Однако в таком случае значение оценки по менее важному критерию, которому соответствует меньшее значение весового коэффициента, будет вызывать существенное влияние на значение степени выполнения α_j всего правила. Это происходит потому, что в композиции (5.25) и импликации (5.26) логическому «И» соответствует оператор минимума [57]. Таким образом, может искусственно занижаться значение степени выполнения отдельных правил α_j. Очевидно, такой способ взвешивания степеней истинности предпосылок правил для оценивания ОТС не является корректным.

Внедрение весовых коэффициентов (1-W_{ct}) в качестве корректирующих множителей тоже не обеспечивает корректного отображения весовых соотношений между значениями оценок по критериям в расчете степеней выполнения правил. В этом случае большое влияние также может иметь оценка по менее важному критерию (при меньшей степени истинности предпосылки по этой оценке).

Для корректного учета различных значений весовых коэффициентов оценок экспертов нужно обеспечить адекватное влияние каждой оценки на формирование значения степеней выполнения правил и соответственно результатов логического вывода. При использовании единой базы правил для определении степени выполнения α_j в процедуре композиции степеней истинности предпосылок в правилах (5.25) и (5.31) вместо оператора минимума предлагается использовать специальный аппарат взвешенной агрегации значений степеней истинности $\mu_c^j(x_c)$ предпосылок правила. При этом правило выполняется, если значение степени истинности $\mu_c^j(x_c)$ каждой предпосылки правила больше нуля. Предложенная СНЛВ с взвешенной истинностью фактически соединяет положительные моменты идей Мамдани, Ларсена и Сугено.

Иногда вычислительная простота делает целесообразным применение в процедуре композиции (5.25) взвешенной суммы значений оценок, т. е. линейной комбинации значений оценок по всем критериям и их весовых коэффициентов. В этом случае степень выполнения α_j правила $Rule_j$ при применении СНЛВ для эксперта E_t определяется следующим образом:

$$\alpha_j = \sum_{c=1}^{h} W_{ct} \cdot \mu_c^j(x_c),\ \mu_c^j(x_c) > 0. \qquad (5.33)$$

Для задачи сведения метрик операция композиции для правила $Rule_j$ в процедуре нечеткого логического вывода СНЛВ с взвешенной истинностью для метрики M_i и эксперта E_t формулируется так:

$$\alpha_j = \sum_{c=1}^{h} W_{ct} \cdot \mu_c^j(S_{ict}),\ \mu_c^j(S_{ict}) > 0. \qquad (5.34)$$

Необходимо отметить, что СНЛВ с взвешенной истинностью по существу построена на идее использования прописанных правилам $Rule_j$ весов w_j, $j = \overline{1,r}$ (каждому правилу отвечает свой вес) с целью интеграции оценок экспертов.

Пример применения СНЛВ со взвешенной истинностью. Рассмотрим применение СНЛВ со взвешенной истинностью для оценки удовлетворенности пользователя уровнем технического обслуживания Интернет-провайдером сервиса «Интернет». Параметр обслуживания (см. рис. 5.7) оценивается такими метриками:

- M_1 – удобство пользования;
- M_2 – техническая поддержка;
- M_3 – стабильность и непрерывность обслуживания;
- M_4 – время определения причины неисправности, о которой сообщил пользователь;
- M_5 – время устранения неисправности, о которой сообщил пользователь.

Таким образом, нужно обрабатывать пять метрик $M = \{M_i\}$, $i = \overline{1,5}$. Экспертный комитет состоит из четырех экспертов $E = \{E_t\}$, $t = \overline{1,4}$. Необходимо наилучшим образом свести эти метрики в сводную метрику «удовлетворенность обслуживанием».

Каждый эксперт E_t для каждой метрики M_i дает оценки S_{ict} по трем критериям: $C = \{C_c\}$, $c = 1, 2, 3$: C_1 – важность метрики для пользователя; C_2 – важность метрики для ОТС; C_3 – весомость метрики в сводной метрике. Эксперты $E = \{E_t\}$ для критериев оценки $C = \{C_c\}$ задают индивидуальные весовые коэффициенты W_{ct}.

Нормированные значения весовых коэффициентов критериев и критериальные экспертные оценки метрик приведены в табл. 5.3.

Для получения обобщенных оценок метрики от каждого эксперта критериальные оценки экспертов будем агрегировать с помощью СНЛВ со взвешенной истинностью.

Таблица 5.3. Критериальные экспертные оценки метрик

Эксперт, E_t	Критерий, C_c	Вес, W_{ct}	Оценки метрик, M_i				
			M_1	M_2	M_3	M_4	M_5
E_1	C_1	0,443	0,532	0,605	0,791	0,623	0,592
	C_2	0,372	0,421	0,451	0,482	0,411	0,403
	C_3	0,185	0,353	0,225	0,321	0,380	0,322
E_2	C_1	0,534	0,627	0,725	0,625	0,592	0,711
	C_2	0,325	0,551	0,621	0,731	0,631	0,595
	C_3	0,141	0,383	0,456	0,352	0,452	0,392
E_3	C_1	0,582	0,680	0,712	0,630	0,581	0,705
	C_2	0,275	0,610	0,580	0,710	0,632	0,570
	C_3	0,143	0,445	0,381	0,411	0,491	0,365
E_4	C_1	0,482	0,632	0,680	0,731	0,592	0,692
	C_2	0,335	0,710	0,671	0,621	0,712	0,585
	C_3	0,183	0,403	0,452	0,392	0,482	0,407

СНЛВ имеет 3 входа, соответствующие критериям оценки, и одно выходное значение – степень важности метрики. Лингвистические переменные, соответствующие входным значениям, имеют по три значения: *малая, средняя, большая*, (интенсивность значения метрики). Их графическая иллюстрация приведена на рис. 5.9,*а*. Лингвистическая переменная выхода СНЛВ также имеет три градации: *низкая, средняя, высокая* (степень важности). Их графическая иллюстрация представлена на рис. 5.9,*б*. Для соответствующих нечетких множеств входа и выхода применяются треугольные функции принадлежности.

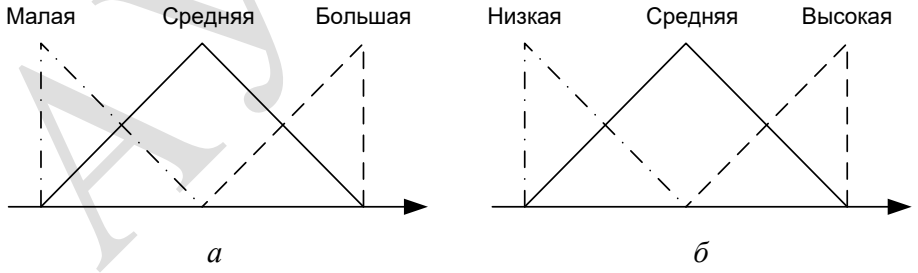


Рис. 5.9. Лингвистическая переменная:
а – входа и *б* – выхода

База правил СНЛВ приведена в табл. 5.4. Степень выполнения каждого правила определяется с помощью выражения (5.31), а процедура импликации осуществляется по Мамдани (5.25).

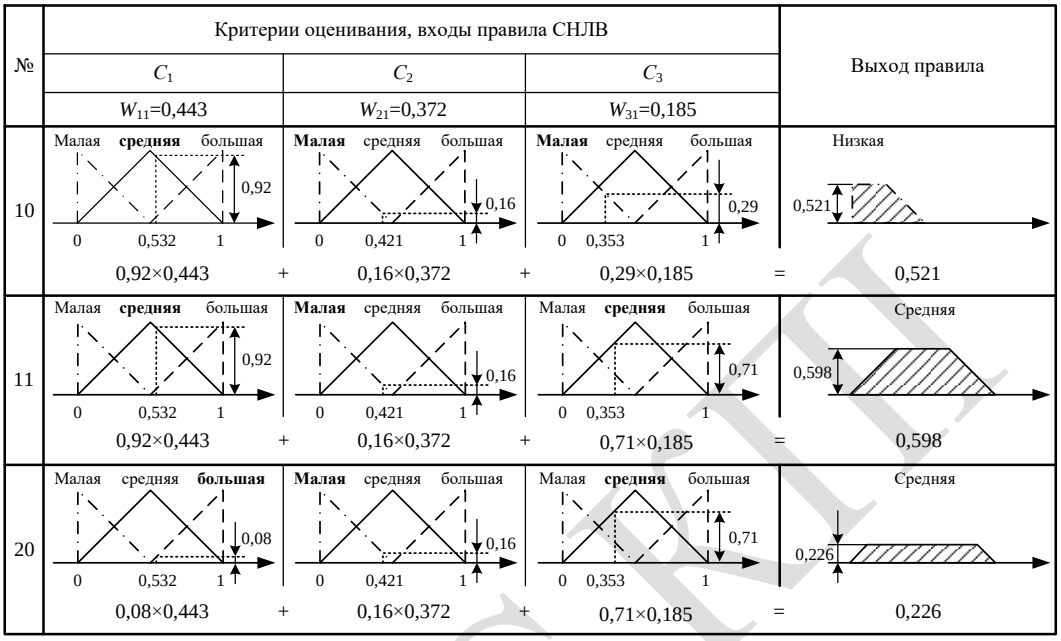
Таблица 5.4. База правил примера

№	Правила нечеткого логического вывода			
	Входные значения			Выходное значение
	Важность		Весомость в сведенной метрике	Степень важности
	для пользователя	для ОТС		
1	малая	малая	малая	низкая
2	малая	малая	средняя	низкая
3	малая	малая	большая	средняя
4	малая	средняя	малая	низкая
5	малая	средняя	средняя	средняя
6	малая	средняя	большая	средняя
7	малая	большая	малая	средняя
8	малая	большая	средняя	средняя
9	малая	большая	большая	средняя
10	средняя	малая	малая	низкая
11	средняя	малая	средняя	средняя
12	средняя	малая	большая	средняя
13	средняя	средняя	малая	средняя
14	средняя	средняя	средняя	средняя
15	средняя	средняя	большая	средняя
16	средняя	большая	малая	средняя
17	средняя	большая	средняя	средняя
18	средняя	большая	большая	высокая
19	большая	малая	малая	средняя
20	большая	малая	средняя	средняя
21	большая	малая	большая	высокая
22	большая	средняя	малая	средняя
23	большая	средняя	средняя	средняя
24	большая	средняя	большая	высокая
25	большая	большая	малая	высокая
26	большая	большая	средняя	высокая
27	большая	большая	большая	высокая

Рассмотрим работу СНЛВ со взвешенной истинностью для получения агрегированных значений оценок эксперта E_1 для метрики M_1. В СНЛВ для значений критериальных оценок табл. 4.3 эксперта E_1 для метрики M_1 сработали правила 10, 11, 13, 14, 19, 20, 22, 23. В табл. 5.5 проиллюстрирована процедура расчета значений истинности предпосылок и степени выполнения для правил 10, 11, 20.

АВТОРСКИЙ

Таблица 5.5. Пример выполнения правил 10, 11, 20



Значение агрегированных с помощью СНЛВ оценок всех экспертов для всех метрик приведены в табл. 5.6.

Таблица 5.6. Агрегированные с помощью СНЛВ экспертные оценки метрик

Эксперт, E_t	Оценки метрик M_i, A_{it}^E				
	M_1	M_2	M_3	M_4	M_5
E_1	0,149	0,219	0,266	0,224	0,223
E_2	0,207	0,331	0,329	0,206	0,312
E_3	0,292	0,323	0,304	0,185	0,317
E_4	0,348	0,309	0,350	0,248	0,286

Для решения задачи сведения метрик находим обобщенную оценку каждой метрики – степень ее важности для сводной метрики. Уровни экспертов будем считать одинаковыми, а степени важности рассчитаем как средние значения агрегированных экспертных оценок метрик с помощью выражения (5.21). Результаты представлены в табл. 5.7.

Таблица 5.7. Окончательные оценки метрик

Степень важности метрики, A_i	Оценки метрик, M_i				
	M_1	M_2	M_3	M_4	M_5
Рассчитанное значение	0,249	0,296	0,312	0,216	0,285
Нормированное значение	0,183	0,218	0,230	0,159	0,210

АНТОКИ

Согласно найденным нормированным значениям степеней важности метрик, (табл. 5.7) на основе формулы (5.22) можно определять конкретное значение сводной метрики $M_{св}$, если заданы конкретные значения $z_1,\ldots,z_5$ метрик $M_1,\ldots,M_5$.

Таким образом, предложен оригинальный метод оценки уровня обслуживания пользователей ИТ-инфраструктуры путем сведения количественных и качественных метрик. Метод отличается комбинированным применением методов экспертных оценок и подходов на основе теории нечетких множеств и обеспечивает получение оценок сети ОТС за приемлемое время. Применение СНЛВ со взвешенной истинностью обеспечивает учет индивидуальных весовых коэффициентов критериев оценки метрик для каждого эксперта при применении экспертным комитетом единой базы правил СНЛВ. Сведение метрик выполняется с помощью системы нечеткого логического вывода с привлечением экспертов. Это требует участия экспертного комитета и эффективной автоматизации процесса оценивания.

Можно также использовать предложенный в работе [152] метод агрегирования показателей оценки качества услуг. В этом случае оценки задаются в виде нечетких множеств, причем в качестве входных показателей используются лингвистические переменные оценки параметров нижнего уровня, а в качестве выходной переменной – обобщенная лингвистическая переменная для оценки верхнего уровня. В следующем цикле расчетов обобщенная оценка будет выступать в качестве входной переменной следующего, более высокого уровня. Для метода агрегирования разработан алгоритм обучения, позволяющий строить функции принадлежности входных нечетких переменных и формировать нечеткие правила сведения показателей оценки качества сервисов.

5.6. НЕЧЕТКОЕ ОЦЕНИВАНИЕ В ЗАДАЧАХ УПРАВЛЕНИЯ УРОВНЕМ ОБСЛУЖИВАНИЯ

5.6.1. Проблема агрегирования показателей оценки качества сервисов, заданных в виде нечетких множеств

Экономические показатели заставляют компании использовать внешние ресурсы при конструировании ИТ-сервисов. ИТ-организации, предоставляющие услуги связи и доступ к ресурсам, имеют достаточно большие масштабы и развитую систему связей между структурными подразделениями и службами, что обязывает их поддерживать сложную ИТ-инфраструктуру. Для эффективного управления деятельностью ИТ-организаций необходимы хорошо продуманные бизнес-процессы, а в основу принятия решений по ключевым моментам деятельности должны быть положены математические модели и методы. Возникает потребность в создании математических моделей и методов для:

1) планирования и контроля работы;
2) оценки эффективности бизнес-процессов;
3) оценки эффективности отдельных служб;
4) улучшения бизнес-процессов и др.

Традиционно ИТ-организации оценивают свою деятельность с помощью ряда метрик технологического (операционного) уровня, например, пропускная способность, если речь идет об управлении качеством предоставления услуг по

доступу к интернету, задержка при передаче пакетов или частость ошибок, если оценивается, например, технология IP поверх ATM.

Но эти метрики не учитывают влияния на бизнес современных тенденций к предоставлению услуг в виде сервисов с акцентом на максимальное удовлетворение потребностей пользователей. Поэтому метрики операционного уровня не могут быть основой для всестороннего анализа, оптимизации и управления деятельностью ИТ-организаций. Необходим переход к метрикам, связанным с оценкой сервисов, осуществлением ожиданий пользователей и влиянием на бизнес самих провайдеров сервисов. Эти метрики позволяют ИТ-организациям более полно оценить свою деятельность, уровень сервисов, удовлетворенность пользователей и влияние отдельных аспектов эксплуатации, технического обслуживания, управления процессами предоставления услуг, планирования, проектирования и развития ИТ-инфраструктуры на бизнес.

Такой переход требует разработки новых методов оценки влияния отдельных аспектов на бизнес в разрезе анализа их связей с базовыми метриками операционного уровня. Для этого необходимы модели и методы бизнес-анализа, позволяющие улучшить деятельность ИТ-организаций. Нужно решить проблему оптимизации в разрезе бизнес-анализа и построить систему поддержки принятия решений (СППР), которая позволяет оптимизировать структуру и деятельность ИТ-организаций. Поэтому актуальной является проблема агрегирования метрик одного уровня в метрики высшего уровня, когда применяются нечеткие метрики.

Осознание критической важности управления качеством телекоммуникационных сервисов, особенно в отношении доступности и надежности, для бизнеса ОТС закреплено в документах TMF – международной ассоциации лидеров ОТС. SLA определяет требования к качеству сервисов, предоставляемых ОТС через показатели их доступности, надежности, производительности. Поэтому для автоматизации бизнес-процессов ОТС с точки зрения стоимостных и временных критериев особую важность приобрел документ SLA Management Handbook [347, 348, 349]. Его появление предопределила необходимость учета изменившихся условий деятельности ОТС и прежде всего появление новых видов ОТС – провайдеров интернет, прикладных приложений и контента, а также тенденции перехода к архитектурам, делающим возможной электронную коммерцию, усложнение инфраструктуры обеспечения сервисов, акцентирование на проблемах пользователей, возрастающая роль автоматизации.

TMF предложил новую модель организации деятельности ОТС в области телекоммуникаций – eTOM. В eTOM нашли отражение бизнес-процессы, предназначенные для совершенствования управления SLA, качеством сервиса и другими направлениями деятельности ОТС. Руководители ОТС и заказчики понимают, что SLA способен подтолкнуть заказчиков использовать новые сервисы и технологии, если ОТС обеспечит необходимый уровень характеристик сервисов и гарантирует пользователям возможность вести свой бизнес эффективно с точки зрения стоимости и с минимальными рисками. Для этого, как со стороны заказчиков, так и со стороны ОТС в составе их СУ должны быть модули, способные помочь превратить SLA в инструмент формирования бизнес-продуктов. Надлежащим образом подготовленный SLA позволяет определить ожидания от каждого элемента нужного сервиса и управлять этими ожиданиями, а также поддерживает эффективные изменения в деятельности, улучшения

измерений и отчетности, прогнозирования, закладывая при этом основы для успеха заказчиков.

Необходимость автоматизированного управления уровнем сервисного обслуживания осознана разработчиками СУИ для ОТС. Исходя из общей тенденции ИТ-управления фокусироваться не на управлении сетями, а на управлении сервисами и клиентами с точки зрения целей и политик бизнеса, разработчики предложили несколько программно-аппаратных решений, (см., например, [197]).

Сложность возникших проблем обусловлена размерами сетей и организационных структур поддержки деятельности ОТС и необходимостью увязать в единой СУ много факторов, которые ранее не учитывались. Для решения этих проблем необходимо разработать, автоматизировать и внедрить комплекс адекватных процессов, способных повысить эффективность СУ ОТС в новых условиях. В свою очередь, это требует разработки новых моделей, связанных с генерацией, оцениванием, выбором и реализацией решений. Многокритериальность, неопределенность, необходимость учета рисков – это лишь некоторые из обстоятельств, которые следует учитывать при разработке этих моделей.

Концептуально автоматизированное управление уровнем сервисного обслуживания на стороне ОТС должно базироваться на показателях:

1) влияния неполадок на финансовые потери за счет недоплат клиентов, штрафов, которые выставят клиенты, перехода клиентов к другим провайдерам;

2) доходов, которые приносит сеть и ее составляющие, вплоть до отдельных элементов.

Для эффективного управления уровнем сервисного обслуживания на основе указанных показателей необходимо построить зависимости между состоянием сети и ее элементов, динамикой отказов и потерями/доходами, чтобы можно было дать бизнесу инструмент для оценки решений относительно инвестиций в сеть и принятия SLA на основе стоимостных параметров этих решений для ОТС.

Современным взглядам на структурирование крупных сетей присуще выделение в них четырех основных уровней – ресурсов, сетей, сервисов и пользователей. На каждом из них выстраивается своя система оценки качества. На уровне ресурсов используется громоздкая система технологических показателей, которая существенно зависит от технологий и оборудования. Здесь необходимо учесть показатели технологий IP, ATM, DSL, Ethernet, MPLS и др. Моделируются отдельные элементы, например маршрутизаторы, коммутаторы и линии связи, а также композиции технологий. На уровне сетей используются релевантные для этого уровня показатели функционирования технологий и оборудования – KPI. Здесь интерес вызывают оценивание качества сервиса в сети в месте подключения пользователя, задержки передачи пакетов или частость ошибок передачи пакетов между узлами сети. Но эти показатели малоинформативны относительно качества предоставления сервиса на всем пути от ресурсов до пользователя. Поэтому на уровне сервисов используются релевантные для этого уровня показатели качества – KQI, которые обеспечивают измерение специфических аспектов функционирования продуктов, сервисов и их элементов на основе KPI, а также информации из других источников. При этом выделяются KQI для сервисов, которые агрегируют KPI для оценки качества

сервисов и их элементов, и обобщенные KQI, предназначенные для оценки продуктов, предлагаемых пользователю. Именно последние KQI, которые определяются на основе сервисных KQI, часто включаются в SLA.

На уровне пользователей используются также специфические для пользователей показатели QoS. Примерами этих технологически нейтральных метрик, с помощью которых оценивают качество, являются, например, показатели готовности сервисов, среднее время между отказами сервисов и другие аспекты надежности предоставления сервисов, качество обслуживания пользователей, точность биллинга и т. д. Метрики уровня пользователей ориентированы на клиентов и позволяют оценить выполнение услуги (например, удовлетворенность пользователей качеством услуг), поддержки (например, среднее время ожидания ответа службы помощи), биллинга (например, точность биллинга) и другие аспекты. Важнейшей метрикой этого уровня является удовлетворенность пользователя, которая выступает оценкой настроенности пользователя на продолжение пользоваться услугами. С точки зрения бизнеса оцениваются метрики трех групп: метрики пользователей (например, средний уровень удовлетворенности пользователей, коэффициент «текучести» пользователей); доходности (доходность сервиса и т. п.); CAB (charging, accounting, billing).

Методической базой концепции автоматизированного управления уровнем сервисного обслуживания должна быть иерархическая система измерения, накопления, обобщения и преобразования показателей на каждом уровне и между уровнями. Воплощение нового подхода к организации деятельности ОТС требует моделей и соответствующих методик для:

1) классификации операторов, заказчиков и сервисов;
2) оценки уровня сервисов клиентами (QoE);
3) оценки уровня сервисов для клиента оператором услуг;
4) определения уровня функциональности сервисов;
5) определения технических параметров сервисов;
6) определения параметров функционирования оборудования сети ОТС.

Кроме того, необходима общая многоуровневая модель управления уровнем сервисного обслуживания, которая адекватно связывает параметры функционирования сетей и технологий с ожиданиями пользователей и показателями прибыльности бизнеса.

Особое место занимает сведение показателей одного уровня в интегрированные показатели этого же или более высокого уровня. Рост сетей, использование различных технологий и расширения приложений пользователей и сервисов создали ситуацию, когда СУ ОТС переполнены потоками информации, поступающей от сетей, оборудования, приложений, сервисов и других элементов ИТ-инфраструктуры. Метрик на низших уровнях больше, чем на высших, причем последние являются результатом агрегирования первых. Это даже дало основание некоторым исследователям [197] изобразить взаимосвязанные метрики в виде пирамиды и предложить модель дерева метрик, которая позволяет строить композиции нужных для оценки сетей метрик. Чем выше уровень анализа сетей ОТС, тем больше факторов неопределенности влияет на оценку их показателей, оправдывает применение недетерминированных методов анализа, прежде всего теории нечетких множеств и нечеткой логики. Например, общую оценку удовлетворенности пользователя сервисом или общую оценку сервиса очень удобно моделировать в терминах нечетких множеств, а агрегирования показателей осуществлять с помощью правил нечеткой логики.

Традиционно для расчета качества обслуживания в глобальных и локальных информационно-телекоммуникационных сетях применяются методы теории массового обслуживания. Согласно классическому подходу Клейнрока и Джексона [51], разработанному для ИВС, ИУС также можно рассматривать как множество независимых узлов, каждый из которых представляет СМО. Базовая модель рассматривает эти СМО как одноканальные с бесконечными очередями и потоками заявок. При определенных предположениях относительно независимости потоков в узлах, стационарности потоков заявок и распределения времени поступления заявок и их обслуживания по экспоненциальному закону для этих моделей получены аналитические выражения для оценки основных параметров ИВС, таких как среднее время обслуживания заявки, эффективная пропускная способность, вероятность отказа в обслуживании.

Применение моделей на основе СМО позволило не только построить эффективный инструментарий исследования каналов связи, магистральных сетей, но и создало условия для появления новых моделей ИВС, учитывающих ограничения на буферные устройства, приоритетное обслуживание, адаптивное распределение трафика, многоканальность [11, 12, 96].

Для отказа от предположения о независимости потока в каждом узле сети применяются модели системного анализа, построенные на средствах теории графов.

Отказаться от предположения о стационарности потока и расширить выбор законов распределения времени поступления заявок и их обслуживание позволит применить теорию нечетких множеств и нечеткой логики. В условиях нечеткого описания системы и/или нечетко сформулированных критериев эти методы можно эффективно применять для оптимизации параметров сетей и управления ими, поскольку они дают возможность более адекватно представить процессы в сетях. Сегодня методы теории нечетких множеств и нечеткой логики в основном применяются для моделирования входного потока заявок, параметров сетей, например связности, кратчайших путей [12, 85].

В задачах управления уровнем обслуживания методы теории нечетких множеств и нечеткой логики применяются достаточно редко. Здесь широко используются методы исследования операций, прежде всего теория полезности, модели и методы искусственного интеллекта, например продукционные системы и логический подход [207, 212].

В общем на уровне ресурсов и сетей существуют хорошо продуманные подходы к комплексной оценке их функционирования, (см., например, [308, 283]). Здесь определены наборы критериев, проработаны метрики для ряда технологий. Ситуация с разработкой моделей и методов определения KQI с учетом аспектов бизнеса провайдера не такая радужная. Период констатации необходимости анализа профилей пользователей и сетей в процессах принятия решений ОТС уже завершился и появились интересные подходы к управлению параметрами SLA с учетом бизнес-интересов [190, 203]. Но исследование метрик и их взаимовлияния все еще не охватывают всего спектра факторов влияния, особенностей новых видов провайдеров и пользователей. Учитывая изложенное выше и изменчивость ИТ-инфраструктуры, целесообразно развивать модели и методы агрегирования нечетких данных.

5.6.2. Постановка проблемы агрегирования нечетких показателей оценки качества сервисов

Пусть задана двухуровневая система показателей функционирования сетей, сервисов, технологий и других компонентов ИТ-инфраструктуры. Причем на нижнем уровне используется n показателей $K_1,K_2,\ldots,K_n$, которые должны быть сведены в один показатель Q верхнего уровня. Значениями этих показателей являются не числа, а слова или предложения естественного или определенного формального языка. Например, такую характеристику как качество сервиса можно определить с помощью переменной «*Качество*», значения которой могуть принадлежать множеству $S = \{$*отличное* (S_1); *очень хорошее* (S_2); *хорошее* (S_3); *среднее* (S_4); *ниже среднего* (S_5); *плохое* (S_6) и *очень плохое* (S_7)$\}$. Переменная «*Качество*» была бы обычной переменной, если бы ее значения были точными числами. Но в данном случае она превращается в лингвистическую переменную, поскольку используется в нечетких рассуждениях человека. Показатели $K_1,K_2,\ldots,K_n$ часто тоже являются лингвистическими, а не числовыми переменными. Они приобретают значения значений из подобных множеству S множеств значений $A_1,A_2,\ldots,A_n$. Лингвистические переменные характеризуют функциями принадлежности, когда каждому значению лингвистической переменной соответствует определенное нечеткое множество со своей функцией принадлежности. Нечеткие множества – естественное обобщение обычных множеств, при отказе от бинарности их характеристической функции и предположении, что она может принимать любые значения на отрезке [0, 1].

В нечеткой логике с n входами и одним выходом рассуждения осуществляются по правилам (нечеткими правилами), которые имеют вид

$$R_j\text{: Если } K_1 \in A_{1,j} \wedge K_2 \in A_{2,j} \wedge \ldots \wedge K_n \in A_{n,j}\text{, то } Q \in S_j, \qquad (5.35)$$

где $K_i,\ i=\overline{1,n}$, – входные лингвистические переменные; Q – выходная лингвистическая переменная; $A_{i,j}$ – нечеткое множество входной лингвистической переменной K_i; R_j – нечеткое правило; S_j – выходное нечеткое множество.

Необходимо разработать метод агрегирования показателей оценки качества сервисов, заданных в виде нечетких множеств. Поскольку в системах управления ИТ-инфраструктурой накоплен огромный статистический материал, этот метод позволяет строить функции принадлежности и нечеткие правила с помощью обучающих выборок данных. Необходимо также разработать соответствующий алгоритм обучения, позволяющий строить функции принадлежности входных нечетких переменных и формировать нечеткие правила сведения показателей оценки качества сервисов.

5.6.3. Общий подход к агрегированию показателей оценки качества сервисов

Описанную выше проблему можно рассматривать как частный случай более общей проблемы создания СППР ОТС на основе принципов eTOM. Такая СППР предназначена для: предварительной обработки и фильтрации потоков данных, характеризующих состояния сети и ее элементов; обеспечения администраторов, экономистов, менеджеров и специалистов различных служб необходимыми для принятия решений данными. В основу подхода к решению обобщенной проблемы целесообразно положить следующие концептуальные положения:

1. Построение системы связанных метрик.
2. Разработка системы методик сведения.
3. Разработка СППР, которую можно настраивать на конкретные методики и модели сведения.

Архитектура СППР приведена на рис. 5.10. Назначение компонентов СППР понятно из названий. Стоит отметить, что в основу функционирования системы положены система метрик и база моделей, позволяющие конструировать схему расчетов оценок. В случае нечетких оценок можно использовать предложенные здесь модель и метод.

Использование системы метрик продемонстрируем на деятельности ОТС, когда на верхнем уровне необходимо определять, управлять и совершенствовать такие бизнес-процессы:

1. Развитие и управление сервисами.
2. Развитие и управление ресурсами.
3. Развитие и управление поставками.
4. Управление маркетингом и предложением услуг.

На первом уровне можно выделить такие подуровни.

1. Управление сервисом в операционном процессе.
2. Управление отношениями с клиентами.
3. Управление ресурсом в операционном процессе.
4. Управление отношениями с поставщиками.

Если рассмотреть подуровень 1, то в нем можно выделить.

1. Поддержку готовности процессов управления сервисом.
2. Конфигурирование и активацию сервисов.
3. Управление проблемами сервисов.
4. Управление качеством сервисов.
5. Тарификацию сервисов.

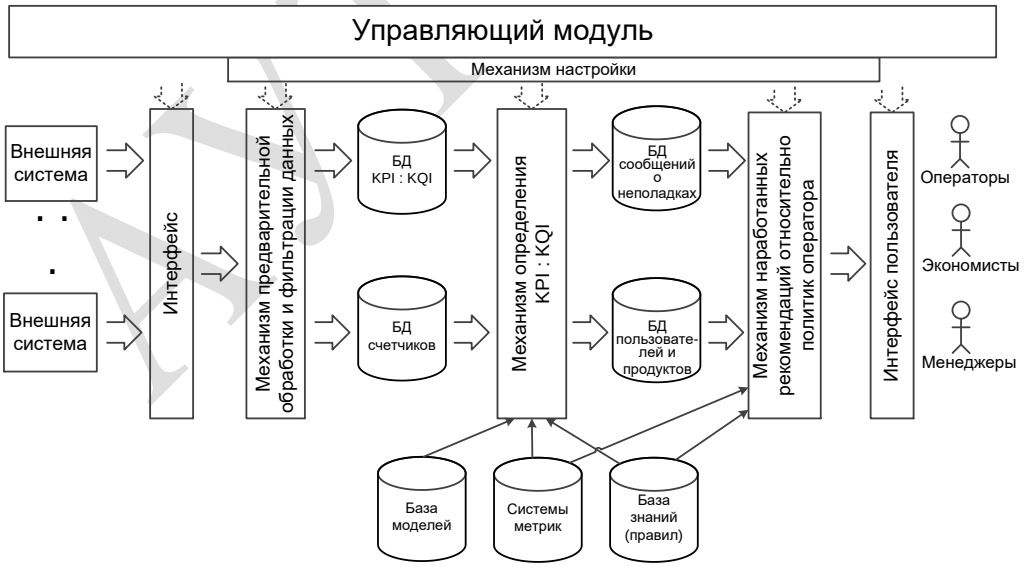


Рис. 5.10. Архитектура СППР

АЛГОРИТМ

Таким образом, выстраивается многоуровневая иерархическая система, в которой с помощью методов оценивания можно построить оценку состояния всей системы бизнес-процессов оператора.

Предложенный метод целесообразно применять тогда, когда в качестве входных показателей используют лингвистические переменные оценки параметров нижнего уровня, а в качестве выходной переменной – уже обобщенную лингвистическую переменную для оценки верхнего уровня, которая в следующем цикле расчетов будет выступать в качестве входной переменной следующего более высокого уровня.

5.6.4. Построение функции принадлежности и нечетких правил

Рассмотрим подход к построению функций принадлежности, применимый в задачах сведения ключевых показателей. В нечеткой системе с n входами и одним выходом нечеткие правила имеют вид (5.35).

Предположим, что существует m пар входов и выходов данных, которые образуют множество числовых обучающих выборок P с n входами и одним выходом $P = \{(k_{1,j},...,k_{n,j}) \mid j = \overline{1,m}\}$, где $k_{i,j}$ – значение i-й входной лингвистической переменной K_i, составляющих множество $(k_{1,j},\ldots,k_{n,j},q_i)$; q_i – значение соответствующей выходной лингвистической переменной Q, $1 \le i \le n$ и $1 \le j \le m$.

Рассмотрим нечеткую модель с несколькими входами и одним выходом. Поскольку нечеткое отношение эквивалентности является рефлексивным, симметричным и транзитивным, четкие данные можно разделить с помощью α-сечений на разные группы. На основе экспериментальных данных можно определить нечеткое отношение совместимости (рефлексивное и симметричное). Нечеткое отношение эквивалентности можно найти как максиминные замыкания нечеткого отношения. Перед тем, как строить нечеткое отношение эквивалентности множества обучающих выборок данных P, необходимо упорядочить множество значений выходной лингвистической переменной Q в порядке возрастания:

$$P' = \{(k'_{1,p},...,k'_{n,p},q'_p) \mid q'_{p_1} \le q'_{p_2}, 1,2,...,m, \text{ и } 1 \le p_1 \le p_2 \le m\}, \tag{5.36}$$

где $(k'_{1,p},k'_{2,p},...,k'_{n,p},q'_p) \in P$.

Значения $q'_1,q'_2,...,q'_m$ выходной лингвистической переменной Q будем рассматривать как упорядоченное множество обучающих выборок данных P'. Нечеткое отношение совместимости $R\left(q'_{p_1},q'_{p_2}\right)$ между выходными лингвистическими переменными Q в упорядоченном множестве обучающей выборки данных P' можно определить с помощью евклидового расстояния:

$$R(q'_{p_1},q'_{p_2}) = 1 - |q'_{p_1} - q'_{p_2}|/\delta. \tag{5.37}$$

Здесь q'_{p_1} и q'_{p_2} – значение переменной Q в упорядоченном множестве обучающей выборки P', а

$$\delta = \left(\sum_{i-1}^{m-1} |q_i - q_m|\right) / (m-1), \tag{5.38}$$

АУТСОРС

где q_m – максимальное значение лингвистической переменной Q во множестве P'. Нечеткое отношение эквивалентности $R^T(q'_{p_1}, q'_{p_2})$ между значениями q'_{p_1} и q'_{p_2} лингвистической переменной Q множества P' можно получить с помощью максиминного транзитивного замыкания отношения совместимости $R(q'_{p_1}, q'_{p_2})$. Поделим множество данных упорядоченной обучающей выборки P' на основе α-сечений отношения эквивалентности $R^T(q'_{p_1}, q'_{p_2})$ на r подмножеств G_j, $j = \overline{1,r}$, где j-е подмножество G_j множества P' можно представить в виде

$$G_j = \{(k'_{1,p}, k'_{2,p}, ..., k'_{n,p}, q'_p) \mid R^T(q'_{p_1}, q'_{p_2}) \geq \alpha,\ \alpha \in [0,1],\ 1 \leq p \leq m,\ 1 \leq p_1, p_2 \leq m. \quad (5.39)$$

Здесь α – пороговое значение, которое выбирается для разбиения множества P'; $1 \leq j \leq r$ и r – количество подмножеств, полученных из множества P'.

Предположим, что j-е множество значений O_j выходной лингвистической переменной Q и j-е множество значений $I_{i,j}$ входной лингвистической переменной K_i получены из j-й подмножества G_j множества P':

$$O_j = \{q_p \mid \forall(k_{1,p}, k_{2,p}, ..., k_{n,p}, y_p) \in G_j, 1 \leq p \leq m\}, 1 \leq j \leq r \quad (5.40)$$

и

$$I_{i,j} = \{k_{i,p} \mid \forall(k_{1,p}, k_{2,p}, ..., k_{i,p}, ..., k_{n,p}, q_p) \in G_j,\ 1 \leq p \leq m\},\ 1 \leq i \leq n \text{ и } 1 \leq j \leq r. \quad (5.41)$$

Таким образом, функцию принадлежности нечетких множеств для выходной лингвистической переменной Q, можно построить, используя множество значений O_j, где $j = \overline{1,r}$, . Поскольку на основе α-сечений отношение эквивалентности значения множества P' разбиты на r разных множеств O_j, $j = \overline{1,r}$, то каждое множество O_j переменной Q можно считать α-сечением $\underline{A_{j,\alpha}}$ выходного нечеткого множества A_j, т. е. $A_{j,\alpha} = \{q \mid q \in O_j \text{ и } \mu_{A_j}(q) \geq \alpha\}$, $j = \overline{1,r}$, где $\mu_{A_j}(q)$ – функция принадлежности выходной нечеткого множества A_j переменной Q; O_j – j-е множество выходных значений переменной Q.

Поскольку значение α-связей $A_{j,\alpha}$ выходной лингвистической переменной Q представлены в порядке возрастания ($q_i \leq q_j$ при $i < j$), то можно вычислить среднее минимума и максимума выходных значений множества α-сечений $A_{j,\alpha}$ и представить их центрами нечетких множеств A_j выходной лингвистической переменной Q, т. е.

$$b_j = (q_{\min} + q_{\max}) / 2, \quad (5.42)$$

где b_j – центр выходного нечеткого множества A_j; $q_{\min}$ и $q_{\max}$ – минимальные и максимальные элементы α-сечений $A_{j,\alpha}$, соответственно. Степень принадлежности минимальных и максимальных элементов α-сечений $A_{j,\alpha}$ является множеством пороговых значений $\alpha \in [0, 1]$. По трем точкам (b_j, 1), ($q_{\min}$, α) и ($q_{\max}$, α) интерполируем левую (a_j, 0) и правую вершины (c_j, 0) нечеткого множества A_j,

АУТСКИ

получая $a_j = b_j - (b_j - q_{\min})/(1-\alpha)$ и $c_j = b_j + (q_{\max} - b_j)/(1-\alpha)$. Теперь функция принадлежности $\mu_{A_j}(q)$ нечеткого множества A_j выходной лингвистической переменной Q может быть представлена в виде тройки (a_j,b_j,c_j):

$$\mu_{A_j}(q) = \begin{cases} (q-a_j)/(|b_j - a_j|), \text{ если } a_j \le q \le b_j; \\ (c_j - q)/(|c_j - b_j|), \text{ если } b_j \le q \le c_j; \\ 0, \text{ в противном случае.} \end{cases} \quad (5.43)$$

Множества P' значений входных лингвистических переменных K_i также разбито на r множеств $I_{i,1}$, $I_{i,2}$, …, $I_{i,r}$ с использованием α-сечений отношения эквивалентности между выходными значениями переменной Q множества P'. В j-м множестве входов и выходов $I_{i,j}$ переменной K_i значение также упорядочены в порядке возрастания. Нечеткое отношение эквивалентности между входными значениями j-го множества входов и выходов $I_{i,j}$ переменной X_i строится на основе сходства входных значений. Далее определяем нечеткое отношение совместимости между значениями множества $I_{i,j}$ входной лингвистической переменной K_i с помощью евклидового расстояния:

$$R_{i,j}(k_{i,p_1}, k_{i,p_2}) = 1 - (|k_{i,p_1} - k_{i,p_2}|)/\delta; \quad (5.44)$$

$$\delta = (\sum_{p=1}^{|I_{i,j}|} |k_{i,p} - k_{i,\max}|)/(|I_{i,j}| - 1), \quad (5.45)$$

где $k_{i,p_1} \in I_{i,j}$, $k_{i,p_2} \in I_{i,j}$, $1 \le i \le r$, $1 \le j \le n$, $1 \le p_1 \le |I_{i,j}|$ и $1 \le p_2 \le |I_{i,j}|$, $|I_{i,j}|$ – количество элементов множества $I_{i,j}$ переменной K_i, $k_{i,\max}$ – максимальное значение переменной K_i во множестве входных значений $I_{i,j}$, и $k_{i,p} \in I_{i,j}$. С помощью максиминного транзитивного замыкания отношения совместимости $R_{i,j}(k_{i,p_1}, k_{i,p_2})$ получаем нечеткое отношение эквивалентности $R^T_{i,j}(k_{i,p_1}, k_{i,p_2})$ между значениями множества входов $I_{i,j}$ переменной K_i. Далее, используя α-сечение нечеткого отношения эквивалентности, разобьем j-е множество значений $I_{i,j}$ переменной K_i на подмножества входных значений $I_{i,j,v}$: $I_{i,j,v} = \{k_{i,p} \mid k_{i,p} \in I_{i,j}, R^T_{i,j}(k_{i,p_1}, k_{i,p_2}) \ge \alpha$, $\alpha \in [0,1]$, $1 \le p \le |I_{i,j}|$, и $1 \le p_2 \le |I_{i,j}|\}$, $1 \le v \le T_j(K_i)$, где $T_j(K_i)$ – количество подмножеств значений, полученных из j-го множества входов $I_{i,j}$ переменной K_i на основе α-сечений нечеткого отношения эквивалентности, $|I_{i,j}|$ – количество элементов множества входных значений $I_{i,j}$.

Каждое подмножество значений $I_{i,j,v}$ переменной K_i также можно считать α-сечением $A_{(i,j,v),\alpha}$ входного нечеткого множества $A_{i,j,v}$, которое определяется как $A_{(i,j,v),\alpha} = \{k_{i,p} \mid k_{i,p} \in I_{i,j,v}, \mu_{A_{i,j,v}}(k_{i,p}) \ge \alpha\}$, $1 \le j \le r$, $1 \le v \le T_j(K_i)$, где $\mu_{A_{i,j,v}}(k_i)$ – функция принадлежности нечеткого множества $A_{i,j,v}$ входной лингвистической переменной K_i, α – пороговое значение, $\alpha \in [0, 1]$. Входные значения α-сечений $A_{(i,j,v),\alpha}$ также расположены в порядке возрастания (т. е. $k_{i,p_1} \le k_{i,p_2}$, $p_1 \le p_2$). Функцию принадлежности $\mu_{A_{i,j,v}}(k_i)$ нечеткого множества $A_{i,j,v}$ можно представить

тремя числами ($a_{i,j,v}$,$b_{i,j,v}$,$c_{i,j,v}$), где $a_{i,j,v}$, $b_{i,j,v}$ и $c_{i,j,v}$ – центральная, левая и правая вершины нечеткого множества $A_{i,j,v}$ переменной K_i соответственно. Значения ($a_{i,j,v}$,$b_{i,j,v}$,$c_{i,j,v}$) вычисляются как минимум и максимум элементов α-сечения $A_{(i,j,v),\alpha}$ нечеткого множества $A_{i,j,v}$:

$$b_{i,j,v} = (k_{i,\min} + k_{i,\max}) / 2; \tag{5.46}$$

$$a_{i,j,v} = b_{i,j,v} - (b_{i,j,v} - k_{i,\min})/(1-\alpha); \tag{5.47}$$

$$c_{i,j,v} = b_{i,j,v} + (k_{i,\max} - b_{i,j,v})/(1-\alpha), \tag{5.48}$$

где $k_{i,\min}$ и $k_{i,\max}$ – минимальные и максимальные элементы α-сечений ($A_{(i,j,v),\alpha}$) нечеткого множества $A_{i,j,v}$ входной лингвистической переменной K_i, $1 \le j \le r$, $1 \le v \le T_j(K_i)$ и $T_j(K_i)$ – подмножество значений $I_{i,j,v}$, полученное из j-го множества значений $I_{i,j}$ лингвистической переменной K_i. Функция принадлежности нечеткого множества $A_{i,j,v}$ представляется тремя числами ($a_{i,j,v}$,$b_{i,j,v}$,$c_{i,j,v}$) в виде

$$\mu_{A_{i,j,v}}(k_i) = \begin{cases} (k_i - a_{i,j,v})/(\,|\, b_{i,j,v} - a_{i,j,v} \,|\,), \text{ если } a_{i,j,v} \le k_i \le b_{i,j,v}; \\ (c_{i,j,v-} k_i)/(\,|\, c_{i,j,v} - b_{i,j,v} \,|\,), \text{ если } b_{i,j,v} \le k_i \le c_{i,j,v}; \\ 0, \text{ в противном случае,} \end{cases} \tag{5.49}$$

где $a_{i,j,v}$, $b_{i,j,v}$ и $c_{i,j,v}$ – центральная, левая и правая вершины входного нечеткого множества $A_{i,j,v}$ переменной K_i, $1 \le j \le r$, $1 \le v \le T_j(K_i)$, а $T_j(K_i)$ – подмножество входных значений $I_{i,j,v}$, полученное из j-го множества значений $I_{i,j}$ входной лингвистической переменной K_i.

Используя выражения (5.36)–(5.49), разобьем пары входов и выходов множества P' на подмножества входных значений $I_{i,j,v}$ переменной K_i и множество исходных значений O_j переменной Q, где $1 \le i \le n$, $1 \le j \le r$ і $1 \le v \le T_j(K_i)$. Таким образом, на основе подмножества входных значений $I_{i,j,v}$ переменной K_i и множества выходных значений O_j переменной Q получаем функцию принадлежности $\mu_{A_{i,j,v}}(k_i)$ входного нечеткого множества $A_{i,j,v}$ и функцию принадлежности $\mu_{A_j}(q)$ нечеткого множества A_j выходной лингвистической переменной Q соответственно.

Используя иерархические взаимосвязи между подмножествами значений входной лингвистической переменной K_i и множествами значений выходной лингвистической переменной Q, обобщим нечеткие правила. Для иллюстрации иерархических взаимосвязей между множествами выходных значений и подмножествами входных значений рассмотрим систему с двумя входными лингвистическими переменными K_1, K_2 и одной выходной лингвистической переменной Q. При этом j-е множество значений исходной лингвистической переменной Q и j-е множество значений $I_{i,j}$ переменной K_i получим на основе α-сечений отношения эквивалентности между выходными значениями переменной Q множества P', где i = 1, 2; $1 \le j \le r$ и r – число подмножеств, полученных из множества P'. Таким образом, v-му подмножеству входных значений $I_{i,j,v}$ переменной K_i получаем на основе α-сечений отношения эквивалентности между значениями j-го множества входных значений $I_{i,j}$ переменной K_i, где i = 1, 2; $1 \le j \le r$ и $1 \le v \le T_j(K_i)$ и $T_j(K_i)$ –

АУТСКИ

число подмножеств значений $I_{i,j,v}$, полученных из j-го множества значений $I_{i,j}$ входной переменной K_i.

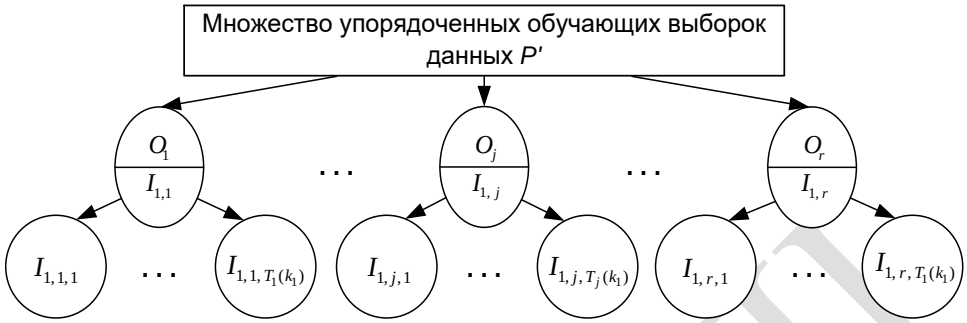


Рис. 5.11. Взаимосвязь между множеством значений A_j выходной лингвистической переменной Q и подмножеством значений $I_{1,j,v}$ входной лингвистической переменной K_i

Существует иерархическое отношение между j-м множеством значений O_j переменной Q и v-м подмножеством входных значений $I_{1,j,v}$ переменной K_1, изображенное на рис. 5.11, где $1 \le j \le r$ и $1 \le v \le T_j(K_i)$. Предположим, что пара входов и выходов $(k_{1,p},k_{2,p},q_p) \in P'$, где $1 \le p \le m$. Если значение $k_{1,p}$ переменной K_1 принадлежит подмножеству входных значений $I_{i,j,v}$, то соответствующее значение q_p переменной Q будет принадлежать множеству выходных значений O_j. На основе иерархических взаимосвязей между j-м множеством выходных значений O_j переменной Q и v-м подмножеством входных значений $I_{1,j,v}$ переменной K_1 можно найти иерархическую взаимосвязь между j-м выходным нечетким множеством A_j и v-м входным нечетким множеством $I_{1,j,v}$, изображенную на рис. 5.12, где $1 \le j \le r$ и $1 \le v \le T_j(K_1)$.

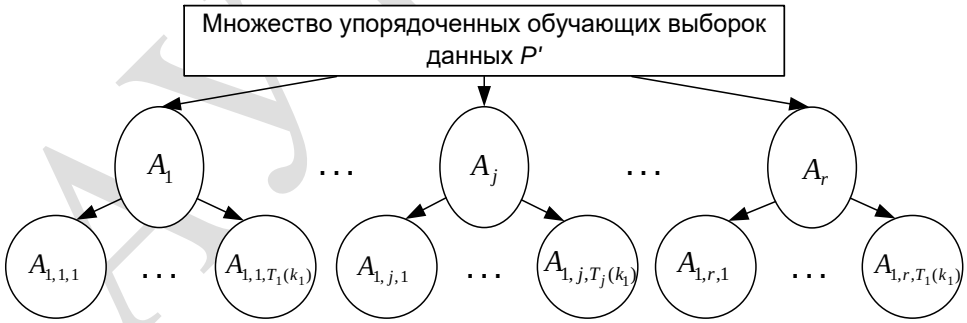


Рис. 5.12. Иллюстрация взаимосвязей между выходными нечеткими множествами A_j выходной лингвистической переменной Q и входными нечеткими множествами $A_{1,j,v}$ выходной лингвистической переменной K_1

Из рис. 5.12 можно получить нечеткие правила, которые связывают нечеткое множество A_j выходной переменной Q и нечеткое множество $I_{1,j,v}$ входной переменной K_1:

$$\begin{array}{ll}
\text{если } K_1 \in A_{1,1,1}, & \text{то } Q \in A_1; \\
\text{если } K_1 \in A_{1,1,T_1(k_1)}, & \text{то } Q \in A_1; \\
\text{если } K_1 \in A_{1,j,1}, & \text{то } Q \in A_j; \\
\text{если } K_1 \in A_{1,j,T_1(k_1)}, & \text{то } Q \in A_1; \\
\text{если } K_1 \in A_{1,r,1}, & \text{то } Q \in A_r; \\
\text{если } K_1 \in A_{1,r,T_r(k_1)}, & \text{то } Q \in A_r.
\end{array} \tag{5.50}$$

Таким же образом получим иерархическую взаимосвязь между j-м множеством значений O_j переменной Q и v-м подмножеством входных значений $I_{2,j,v}$ переменной K_2 (рис. 5.12), где $1 \le j \le r$ и $1 \le v \le T_j(K_2)$, а нечеткие правила можно обобщить так:

$$\begin{array}{ll}
\text{если } K_2 \in A_{2,1,1}, & \text{то } Q \in A_1; \\
\text{если } K_2 \in A_{2,1,T_1(k_2)}, & \text{то } Q \in A_1; \\
\text{если } K_2 \in A_{2,j,1}, & \text{то } Q \in A_j; \\
\text{если } K_2 \in A_{2,j,T_1(k_2)}, & \text{то } Q \in A_1; \\
\text{если } K_2 \in A_{2,r,1}, & \text{то } Q \in A_r; \\
\text{если } K_2 \in A_{2,r,T_r(k_2)}, & \text{то } Q \in A_r.
\end{array} \tag{5.51}$$

Если значение $k_{1,p}$ принадлежат множеству $I_{1,j}$ входной лингвистической переменной K_1, а входные значения $k_{2,p}$ принадлежат множеству $I_{2,j}$ входной лингвистической переменной K_2, то значение q_p принадлежит множеству O_j выходной лингвистической переменной Q для обучающих пар входов и выходов $(k_{1,p},k_{2,p},q_p)$ множества P'. Следовательно, если пары $(k_{1,p},k_{2,p},q_p)$ удовлетворяют условиям $K_1 \in A_{1,j,v_1}$ и $K_2 \in A_{2,j,v_2}$ для некоторых $v_1 \in [1,T_j(K_1)]$ и $v_2 \in [1,T_j(K_2)]$, то $Q \in A_j$, где $p \in [1,m]$; $T_j(K_i)$ – количество подмножеств значений j-го множества $I_{i,j}$ входной лингвистической переменной K_i, где $i = 1, 2$.

Нечеткие правила в (5.50) и (5.51) комбинируют на основе иерархических взаимосвязей между исходным нечетким множеством A_j и нечеткими множествами A_{1,j,v_1} и A_{2,j,v_2} входной лингвистической переменной K_2 следующим образом:

$$\text{если } K_1 \in A_{1,j,v_1} \wedge K_2 \in A_{2,j,v_2}, \text{ то } Q \in A_{2,j,v_2}, \tag{5.52}$$

где $1 \le j \le r$, $1 \le v_1 \le T_j(K_1)$ и $1 \le v_2 \le T_j(K_2)$.

Обобщим $\sum_{j=1}^{r} T_j(k_1) T_j(k_2)$ нечеткие правила из данного множества числовых обучающих выборок данных. В общем случае, если в нечеткой системе существует n входных лингвистических переменных $K_1,\ldots,K_n$ с одной исходной лингвистической переменной Q, то общее количество нечетких правил от множества обучающих выборок данных P на основе предложенных нечетких алгоритмов можно записать в виде

$$\sum_{j=1}^{r} \prod_{i=1}^{n} T_j(K_i), \tag{5.53}$$

АУТСКП

где r – число нечетких множеств лингвистической переменной Q; $T_j(K_i)$ – количество нечетких множеств переменной K_i.

Рассмотрим два нечетких правила для двух переменных K_1, K_2 и одной лингвистической переменной Q:

$$\text{если } K_1 = A_{1,1,1} \wedge K_2 = A_{2,1,1}, \text{ то } Q \in A_1;$$

$$\text{если } K_1 = A_{1,2,1} \wedge K_2 = A_{2,2,1}, \text{ то } Q \in A_2,$$

где A_1 и A_2 – нечеткие множества выходной лингвистической переменной Q; $A_{1,1,1}$ и $A_{1,2,1}$ – нечеткие множества входной лингвистической переменной K_1, $A_{2,1,1}$; $A_{2,2,1}$ – нечеткие множества входной переменной K_2 (рис. 5.13).

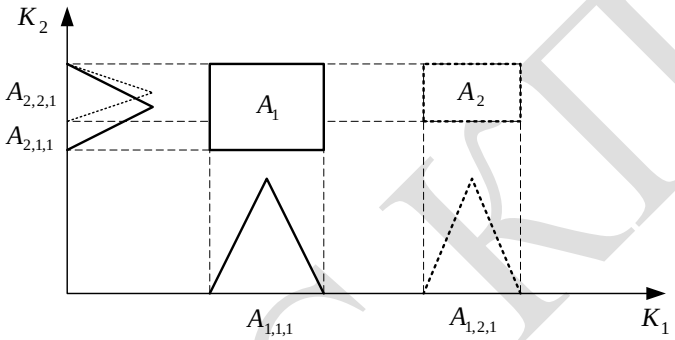


Рис. 5.13. Эквивалентность между входными нечеткими множествами

Из рис. 5.13 видно, что степень правдоподобия между входными нечеткими множествами $A_{2,1,1}$ и $A_{2,2,1}$ лингвистической переменной K_2 является высокой. Поскольку два входных нечетких множества эквивалентны, то можно построить функцию принадлежности нового входного нечеткого множества $A_{2,\text{new}}$, объединяя функции принадлежности входных нечетких множеств $A_{2,1,1}$ и $A_{2,2,1}$ входной лингвистической переменной K_2. После этого заменяем множества $A_{2,1,1}$ и $A_{2,2,1}$ в условной части нечеткого правила на множество $A_{2,\text{new}}$ и получаем нечеткие правила:

$$\text{если } K_1 \in A_{1,1,1} \wedge K_2 = A_{2,\text{new}}, \text{ то } B \in A_1;$$

$$\text{если } K_1 \in A_{1,2,1} \wedge K_2 = A_{2,\text{new}}, \text{ то } B \in A_2.$$

Отметим, что возможно применение и других методов для измерения степени правдоподобия или эквивалентности между различными нечеткими множествами.

Построим матрицы отношения эквивалентности для входных нечетких множеств переменной K_i. Степень равенства двух различных входных нечетких множеств A_{i,j_1,v_1} и A_{i,j_2,v_2} переменной K_i определяется как

$$E(A_{i,j_1,v_1}, A_{i,j_2,v_2}) = (|A_{i,j_1,v_1} \cap A_{i,j_2,v_2}|)/(|A_{i,j_1,v_1} \cup A_{i,j_2,v_2}|), \qquad (5.54)$$

где $1 \le j_1 \le r$, $1 \le j_2 \le r$, $1 \le v_1 \le T_{j_1}(k_i)$ и $1 \le v_2 \le T_{j_2}(k_i)$, $|A_{i,j_1,v_1} \cap A_{i,j_2,v_2}|$ – область пересечения двух различных входных нечетких множеств A_{i,j_1,v_1} и A_{i,j_2,v_2},

$|A_{i,j_1,v_1} \cup A_{i,j_2,v_2}|$ – область объединения двух различных входных нечетких множеств A_{i,j_1,v_1} и A_{i,j_2,v_2}.

Усреднением α-сечений равенства получаем нечеткие множества, в которых степень равенства выше, чем определенное пороговое значение α_{eq}, где $\alpha_{eq} \in [0, 1]$, и, объединяя их, получаем новую функцию принадлежности. Предположим, что $E\left(A_{i,j_1,v_1}, A_{i,j_2,v_2}\right)$ больше, чем пороговое значение α_{eq}, где значение α_{eq} определяется пользователем, $\alpha_{eq} \in [0, 1]$. Тогда можно объединить два входных нечетких множества A_{i,j_1,v_1} и A_{i,j_2,v_2} входной переменной K_i для построения нового нечеткого множества $A_{i,new}$, которое представляется тремя числами $(a_{i,new}, b_{i,new}, c_{i,new})$. Тройка $(a_{i,new}, b_{i,new}, c_{i,new})$ вычисляется усреднением $\left(a_{i,j_1,v_1}, b_{i,j_1,v_1}, c_{i,j_1,v_1}\right)$ и $\left(a_{i,j_2,v_2}, b_{i,j_2,v2}, c_{i,j_2,v_2}\right)$ входных нечетких множеств A_{i,j_1,v_1} и A_{i,j_2,v_2}, соответственно, следующим образом:

$$a_{i,new} = (a_{i,j_1,v_1} + a_{i,j_2,v_2})/2; \tag{5.55}$$

$$b_{i,new} = (b_{i,j_1,v_1} + b_{i,j_2,v_2})/2; \tag{5.56}$$

$$c_{i,new} = (c_{i,j_1,v_1} + c_{i,j_2,v2})/2. \tag{5.57}$$

Функция принадлежности $\mu_{A_i,new}(k_i)$ входного нечеткого множества $A_{i,\text{new}}$ переменной K_i представляется тройкой $(a_{i,new}, b_{i,new}, c_{i,new})$:

$$\mu_{A_{i,new}}(k_i) = \begin{cases} (k_i - a_{i,new})/(|\, b_{i,new} - a_{i,new}), \text{ если } a_{i,new} \le k_i \le b_{i,new}, \\ (c_{i,new-k_i})/(|\, c_{i,new} - b_{i,new}\,|), \text{ если } b_{i,new} \le k_i \le c_{i,new}, \\ 0, \text{ в противном случае.} \end{cases} \tag{5.58}$$

Новое нечеткое множество переменной K_i можно заменить таким входным нечетким множеством переменной K_i, степень равенства которого выше порогового значения α_{eq}.

На основе приведенного метода разработан алгоритм обучения, позволяющий строить функции принадлежности входных нечетких переменных. Обучающая выборка данных P состоит из m пар входов и выходов $(k_{1,p},\ldots,k_{n,p},q_p)$ входных лингвистических переменных $K_1,\ldots,K_n$ и выходной лингвистической переменной Q_p, $1 \le p \le m$.

Шаг 1. Упорядочить обучающие выборки данных P в порядке возрастания на основе значений выходной лингвистической переменной Q и получить множество упорядоченных обучающих выборок данных P'.

Шаг 2. Построить отношения эквивалентности R^T между значениями выходной лингвистической переменной Q множества упорядоченных выборок данных P'.

Шаг 3. Разбить упорядоченные обучающие данные множества P' на r различных множеств значений O_j выходной лингвистической переменной Q и r различных множеств значений $I_{i,j}$ входной лингвистической переменной K_i на

МУТСКП

основе α-сечений отношения эквивалентности R^T, полученных из исходных значений переменной Q множества P', где $1 \le i \le n$ и $1 \le j \le r$.

Шаг 4. Вывести функцию принадлежности $\mu_{A_j}(q)$ выходного нечеткого множества A_j переменной Q на основе α-сечений α-исходного нечеткого множества A_j в отношении множества выходных значений O_j, де $1 \le j \le r$.

Шаг 5. Упорядочить входные значения множества входных значений $I_{i,j}$ входной переменной K_i в порядке возрастания, где $1 \le i \le n$ и $1 \le j \le r$.

Шаг 6. Построить отношения эквивалентности $R^T_{i,j}$ между значениями j-го множества входных значений $I_{i,j}$, где $1 \le i \le n$ и $1 \le j \le r$.

Шаг 7. Разбить множество входных значений $I_{i,j}$ на $T_j(K_i)$ подмножеств входных значений $I_{i,j,v}$ на основе α-сечений отношения эквивалентности $R^T_{i,j}$, где $1 \le i \le n$ и $1 \le j \le r$, $1 \le v \le T_j(K_i)$; $T_j(K_i)$ – количество подмножеств значений, полученных из j-го множества входных значений $I_{i,j}$ входной лингвистической переменной K_i.

Шаг 8. Вывести функцию принадлежности нечеткого множества $\mu_{A_{i,j,v}}(k_i)$ входной лингвистической переменной K_i на основе α-сечений $A_{(i,j,v),\alpha}$ входного нечеткого множества $A_{i,j,v}$ в отношении подмножества входных значений $I_{i,j,v}$ переменной K_i, где $1 \le i \le n$ и $1 \le j \le r$, $1 \le v \le T_j(K_i)$; $T_j(K_i)$ – число подмножеств значений, полученных из j-й множества входных значений $I_{i,j}$ переменной K_i.

Шаг 9. Создать нечеткие правила на основе иерархических взаимосвязей между выходным нечетким множеством A_j выходной лингвистической переменной Q и соответствующим нечетким множеством $A_{i,j,v}$ входной лингвистической переменной K_i, где $1 \le i \le n$ и $1 \le j \le r$, $1 \le v \le T_j(K_i)$.

Шаг 10. Вычислить степень равенства $E(A_{i,j_1,v_1}, A_{i,j_2,v_2})$ между входными нечеткими множествами A_{i,j_1,v_1} и A_{i,j_2,v_2} переменной X_i на основе (5.54), где $1 \le j_1 \le r$, $1 \le j_2 \le r$, $1 \le v_1 \le T_{j_1}(K_i)$ и $1 \le v_2 \le T_{j_2}(K_i)$.

Шаг 11. Если $E(A_{i,j_1,v_1}, A_{i,j_2,v_2}) > \alpha_{\text{eq}}$, где α_{eq} определяется пользователем, $\alpha_{\text{eq}} \in [0, 1]$, $1 \le j_1 \le r$, $1 \le j_2 \le r$, $1 \le v_1 \le T_{j_1}(K_i)$ и $1 \le v_2 \le T_{j_2}(K_i)$, то:

а) построить новое нечеткое множество $A_{i,new}$ переменной K_i путем объединения входящих нечетких множеств A_{i,j_1,v_1} и A_{i,j_2,v_2} входной лингвистической переменной K_i;

б) заменить нечеткие множества A_{i,j_1,v_1} и A_{i,j_2,v_2} входной лингвистической переменной K_i в условной части нечетких правил новым входным нечетким множеством $A_{i,new}$.

Таким образом, предложено решение проблемы агрегирования метрик низшего уровня в метрики высшего уровня, когда применяются нечеткие метрики. Указанная проблема решена в контексте общей проблемы создания и обеспечения функционирования ИТ-инфраструктуры для достижения уровня требований пользователей с наименьшими затратами. Приведен метод построения функции принадлежности и нечетких правил с помощью обучающих выборок данных. Разработан соответствующий алгоритм обучения, который позволяет строить функции принадлежности входных нечетких переменных и формировать нечеткие правила сведения показателей оценки качества сервисов.

5.7. МЕТОД ОЦЕНИВАНИЯ СОСТОЯНИЯ ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Постановка задачи исследования. Оценивание состояния ОМУ усложняется необходимостью: 1) выявления связей и влияния разных по типу параметров; 2) учета зависимости между состояниями ОМУ различных иерархических уровней ИТ-инфраструктуры; 3) быстрой адаптацией к изменению структуры связей параметров и вкладу параметров в общую оценку состояния ОМУ. Поэтому целесообразным видится применение нейронных сетей (НС) для оценивания состояния ОМУ [80]. Для этого необходимо разработать метод оценивания состояния ОМУ с использованием НС и определить наиболее эффективный тип НС для решения задач автоматического оценивания состояния ОМУ.

Метод оценивания состояния элементов ИТ-инфраструктуры с помощью нейронных сетей. В условиях влияния на оценивание состояния ОМУ приведенных выше факторов целесообразно применять НС [111, 330] и задачу оценивания и прогнозирования состояния ОМУ решать путем создания технологии, в которой математические модели элементов ИТ-инфраструктуры играют роль супервизоров для искусственных НС.

Для обучения НС формируется множество $R=\{S,W,Y\}$, где $S=\{s_n\}$ – множество значений параметров ОМУ, $n=\overline{1,N}$, N – количество параметров ОМУ, влияющих на его состояние; $W=\{w_n\}$, $n=\overline{1,N}$, – множество стартовых весовых коэффициентов НС; $Y=\{y_k\}$, $k=\overline{1,K}$, – множество контрольных состояний ОМУ. В процессе обучения НС на ее входы поступают сигналы, соответствующие s_n, $n=\overline{1,N}$. Взвешенные весовыми коэффициентами w_n, $n=\overline{1,N}$, входные сигналы суммируются, проходят через передаточную функцию, генерируют результат y, поступающий на выход НС, где он сравнивается с контрольным значением из множества Y. По отклонению y от y_k, $k=\overline{1,K}$, формируется значение ошибки E, на минимизацию которой и направлено обучение НС с супервизором. Обучающая выборка должна быть сформирована так, чтобы примеры охватывали всю плоскость исследований, а ошибка E стремилась к минимуму в каждом эксперименте. В обучающей выборке использованы нормированные значения параметров ОМУ в закодированном виде, оценка которых проведена предварительно.

Значение каждого из параметров множества P_i, $i=\overline{1,I}$ (4.8) нормируется и приводится к отрезку [0, 1], причем ОМУ, функционирующему в соответствии с заданным регламентом, соответствуют значения параметров, равные 1. После этого значение параметра кодируется и подается на вход НС. При этом активизируется количество входов НС, равное количеству N_i параметров i-го ОМУ, оценка состояния которого осуществляется.

Оценка состояния ОМУ заканчивается появлением на выходе НС закодированного значения состояния Y:

$$Y=f\left(s_{n,i}, n=\overline{1,N}, i=\overline{1,I}\right), \tag{5.59}$$

где $s_{n,i}$, $n=\overline{1,N}$, $i=\overline{1,I}$, – значение параметра $p_{n,i}$.

Для кодирования состояния ОМУ используется предложенный в п. 4.1 метод. Пример кодирования состояния ОМУ для случая пятипорогового ($L_{2,i} = 5$) ранжирования состояний приведен в табл. 5.8.

Таблица 5.8. Кодирование состояния при пятипороговом ранжировании

Код состояния i-го ОМУ при $n = 5$	Показатель производительности i-го элемента	Лингвистические переменные описания состояния элемента
$A_4^{(5)}$	$P_{3,i}^{(l_3)} \le s_{1,i} \le 1$	«Отлично»
$A_3^{(5)}$	$P_{2,i}^{(l_2)} \le s_{1,i} \le P_{3,i}^{(l_3)}$	«Хорошо»
$A_2^{(5)}$	$P_{1,i}^{(l_1)} \le s_{1,i} \le P_{2,i}^{(l_2)}$	«Удовлетворительно»
$A_1^{(5)}$	$P_{0,i}^{(l_0)} \le s_{1,i} \le P_{1,i}^{(l_1)}$	«Неудовлетворительно»
$A_0^{(5)}$	$0 \le s_{1,i} \le P_{0,i}^{(l_0)}$	«Критично»

Кодовая комбинация для обозначения состояния S_i i-го ОМУ, $i = \overline{1, I}$, имеет вид: $A_3^{(5)} A_4^{(5)} A_4^{(5)} A_3^{(5)} A_4^{(5)}$. Комбинация поступает на входы обученной НС. На ее выходе состояние элемента характеризуется одним символом алфавита $A = \left\{ A_{b1}^{(M_{1,i})}, m_1 = \overline{1, M_{1,i}} \right\}$. Значения порогов устанавливаются и изменяются экспертами. НС выдает код состояния, в котором находится ОМУ. Схема реализации метода продемонстрирована на рис. 5.14.

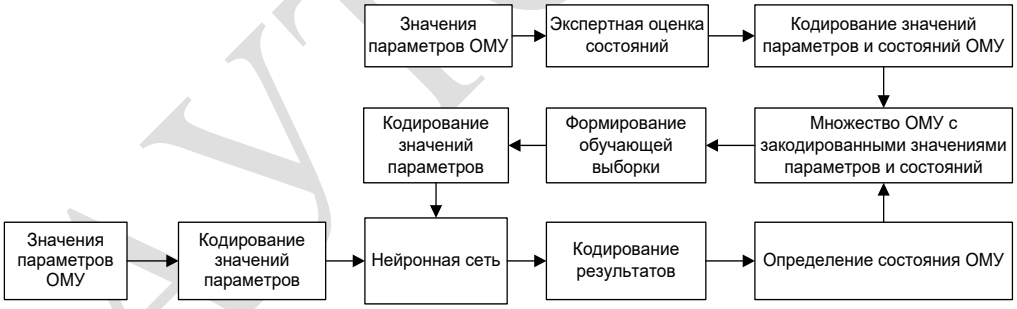


Рис. 5.14. Схема реализации метода оценивания состояния элементов ИТ-инфраструктуры

Результаты исследований. Обучение НС заканчивается построением гиперплоскостей, разделяющих область возможных состояний ОМУ на непересекающиеся зоны, каждая из которых соответствует отдельному состоянию, определяемому, например, по табл. 4.8. Для оценивания состояния ОМУ использовались НС трех типов: Хопфилда (СХ), радиально базисная (РБС) и на основе машины опорных векторов (МОВ). Для каждой НС проводилось обучение с супервизором с учетом индивидуальных особенностей работы НС [63, 129, 167, 196].

В качестве примера для проверки предлагаемого метода выбран ОМУ типа «сервер», состояние которого оценивается такими параметрами: 1) время ожида-

ния доступа к носителям; 2) загрузка процессора/оперативной памяти; 3) свободная оперативная память; 4) загруженность сетевых интерфейсов; 5) температура. По параметрам 1)–4) оценивается работоспособность сервера, а пятый позволяет выявить возможную причину неисправностей. Для первого исследования метода использовалась выборка по элементам ИТ-инфраструктуры типа «сервер» с параметрами 1)–4) (табл. 5.9), а для второго – с параметрами 1)–5) (табл. 5.10). Количество испытаний 100 и 1000.

Таблица 5.9. Количество ошибок СХ, РБС и МОВ при четырех параметрах

Тип НС	Состояние, в котором находится ОМУ тестовой выборки											
	$A_0^{(5)}$		$A_1^{(5)}$		$A_2^{(5)}$		$A_3^{(5)}$		$A_4^{(5)}$		Случайное	
	100	1000	100	1000	100	1000	100	1000	100	1000	100	1000
СХ	2	17	4	32	3	34	3	39	1	9	3	38
РБС	0	3	1	3	0	12	1	9	0	2	1	7
МОВ	0	0	0	2	1	2	0	2	0	0	0	2

Таблица 5.10. Количество ошибок СХ, РБС и МОВ при пяти параметрах

Тип НС	Состояние, в котором находится ОМУ тестовой выборки											
	$A_0^{(5)}$		$A_1^{(5)}$		$A_2^{(5)}$		$A_3^{(5)}$		$A_4^{(5)}$		Случайное	
	100	1000	100	1000	100	1000	100	1000	100	1000	100	1000
СХ	3	23	5	52	3	42	5	37	2	11	4	36
РБС	0	4	1	3	0	12	1	8	0	5	1	14
МОВ	0	1	0	4	1	3	0	4	0	0	0	3

Как видно из табл. 5.9 и 5.10, НС на основе СХ показала худшие результаты. При определении состояния ОМУ погрешность достигает 5%. НС на основе РБС показала результаты на порядок лучше НС на основе СХ. Это обусловлено выполнением операции округления на выходе НС. Максимальная погрешность до операции округления составляла 0,4. Постоянная погрешность в 85% случаев находилась в диапазоне до 0,1. НС на основе МОВ для оценивания состояния элементов ИТ-инфраструктуры представляет собой линейный пороговый классификатор, обучающийся по прецедентам. Этот классификатор для множества P_i параметров i-го ОМУ, принимающих значения из множества S_i значений параметров и множества контрольных значений $Y = \{y_k\}$, строит алгоритм $\alpha_i : s_i \to y_k$, $i = \overline{1, I}$, аппроксимирующий целевую зависимость на всем пространстве изменения значений S_i, $i = \overline{1, I}$ [21]. Точность построения гиперплоскости зависит от объема и качества обучающей выборки. При проведении экспериментов погрешность не превышала 0,01–0,02%.

Из графиков, приведенных на рис. 5.15, видно, что НС на основе СХ допускает 2–5% ошибок при оценке состояния элементов ИТ-инфраструктуры, находящихся в приграничной области состояний. Применение РБС подразумевает написание подпрограммы, которая будет осуществлять операцию математического округления на выходе НС, поскольку в РБС выход является аналоговым. НС на основе РБС допускает в среднем 1% ошибок. Все допущенные НС ошибки связаны с недостаточной точностью построения разделяющей гиперплоскости. Как показывают результаты исследований, наиболее точно гиперплоскости определяются с помощью НС на основе МОВ.

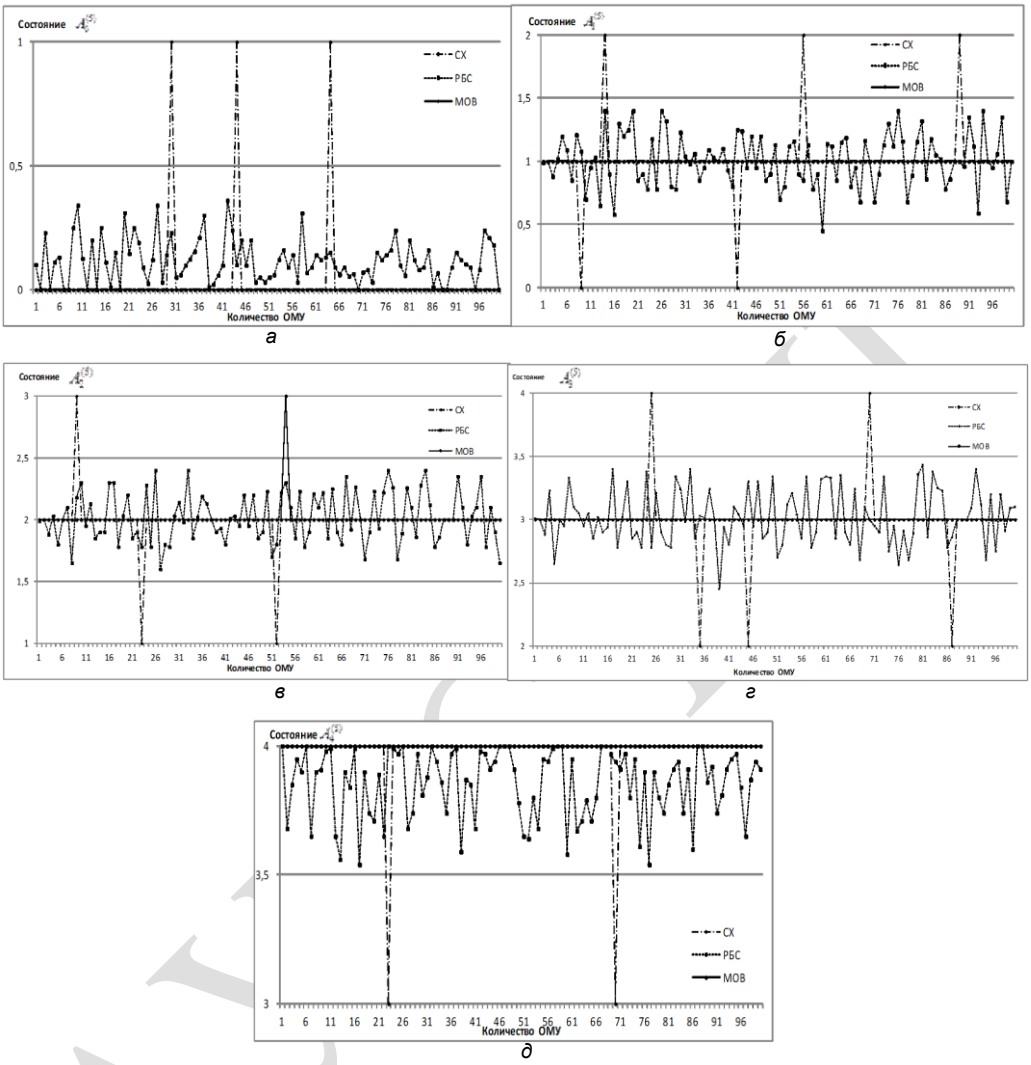


Рис. 5.15. Результаты тестирования НС на основе СХ, РБС и МОВ при n = 5 для ОМУ, находящихся в состоянии: *а* – $A_0^{(5)}$; *б* – $A_1^{(5)}$; *в* – $A_2^{(5)}$; *г* – $A_3^{(5)}$; *д* – $A_4^{(5)}$

По показателям простоты, скорости обучения, быстродействия и точности целесообразно использовать НС на основе МОВ, в которой определение состояния ОМУ сводится к определению принадлежности состояния ОМУ к одному из двух классов. В этом случае необходимо реализовать $M_{n,i}$ параллельных МОВ – линейных пороговых классификаторов, обучающихся по прецедентам. Каждая из параллельных МОВ строит собственную разделяющую гиперплоскость по уравнению $< W, S >= W_0$, где $W_0 = (w_{00}, w_{01}, ..., w_{0m})$, $m = \overline{1, M_{n,i}}$, – скалярный порог между классами. Главной задачей обучения является построение разделяющей гиперплоскости максимальной ширины.

МОВ соотносит объект с определенным классом по следующей схеме [21]:

$$Y_k = \text{sign}(\sum_{i=1}^{N} w_i s_i - w_0) = \text{sign}(< W, S > - W_0)\,. \qquad (5.60)$$

Структура НС для оценки состояния на основе МОВ приведена на рис. 5.16.

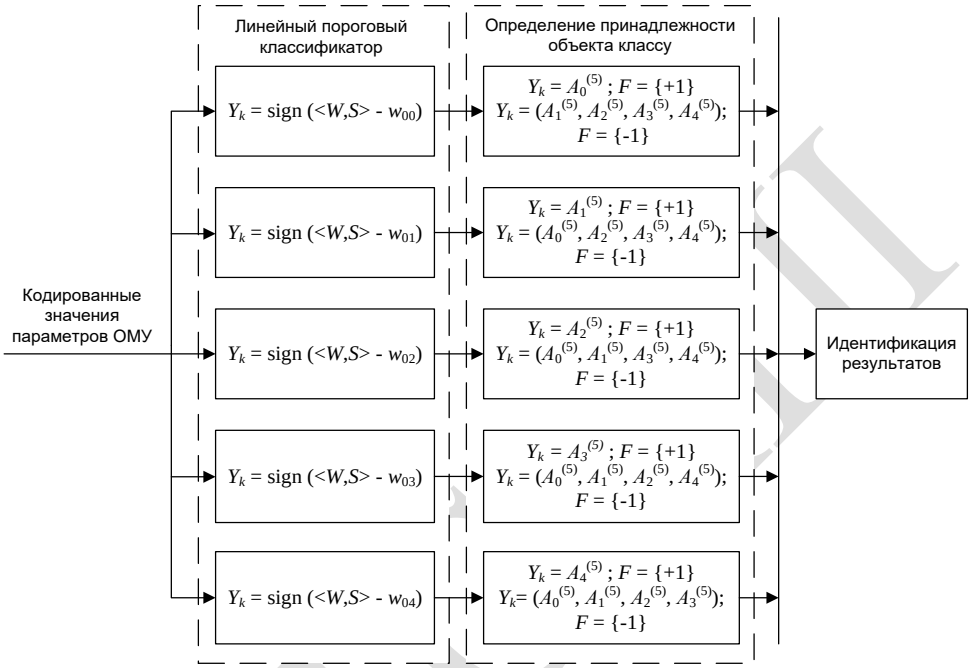


Рис. 5.16. Структура нейронной сети на основе МОВ

Таким образом, экспериментально подтверждено, что именно НС на основе МОВ целесообразно использовать в ИТ УИТИ для решения задачи классификации состояний элементов ИТ-инфраструктуры. При этом разработанный метод позволяет осуществлять автоматический анализ состояния ОМУ с ошибкой, не превышающей 0,01–0,02%.

5.8. МЕТОД РАСПОЗНАВАНИЯ ОБРАЗОВ НА ОСНОВЕ ПОТЕНЦИАЛЬНЫХ ФУНКЦИЙ В ЗАДАЧАХ ОЦЕНКИ УРОВНЯ МУЛЬТИМЕДИЙНЫХ УСЛУГ

ИТ УИТИ предъявляет достаточно жесткие требования к оперативности, точности и стоимости методов оценивания качества мультимедийных услуг. Именно поэтому эталонные методы [253, 305] нельзя использовать в ИТ УИТИ для непрерывного мониторинга качества услуг. По той же причине нельзя применять субъективные методы, основанные на использовании MOS-подобных оценок [306, 296, 291] качества сервиса, которые требуют присутствия экспертов в месте оценки качества. Поэтому для оценки качества мультимедийных услуг используются методы объективного оценивания, такие как NIQA [305], LCQA [247] и др. В частности для оценивания качества IPTV без использования эталона применяются алгоритмы EPSNR, TVQM, MPQM [277].

В плане реализации объективного оценивания представляет интерес исследование возможности оценки качества услуг на основе значений показателей трафика, проходящего через сетевые узлы. При этом возможна существенная погрешность, однако для ОТС важнее получать оценку качества с минимальной стоимостью, чем с большой точностью.

Предельные значения ключевых метрик, при которых качество мультимедийных услуг еще соответствует допустимому уровню, регламентируется документами [370, 308, 292, 361, 286]. Однако стандартизируются пороговые значения только отдельных метрик. Для решения задач управления уровнем услуг в крупных ОТС представляет интерес установление зависимости качества услуг от значений параметров трафика [110, 115].

Постановка задачи исследований. Необходимо разработать метод оценки уровня мультимедийных услуг на основе анализа значений совокупности ключевых метрик, измеренных в узлах ТКС, при несбалансированных выборках.

Применение метода потенциальных функций в задачах оценки уровня сервисов. Задачу оценки уровня услуг на основе анализа значений ключевых метрик, измеренных в узлах ТКС, можно рассматривать как задачу распознавания образов. В этом случае классами выступают оценки уровня услуг: «неудовлетворительный», …, «отличный», признаками – значения метрик соответствующего уровня сети, а в качестве объектов – зафиксированные ситуации, характеризующиеся значениями метрик и относящиеся к одному из классов с определенным уровнем услуг.

Обучив классифицирующие модели, уже на этапе мониторинга ТКС по значениям метрик можно определять ожидаемый уровень предоставляемых услуг в подсети без использования аппаратуры, установленной на клиентской стороне.

Предлагаемый метод построен на использовании накопленного комплекса значений метрик [110]. Опросами и сбором жалоб пользователей, а также проведением тестов с участием экспертов фиксируются периоды, когда был известен уровень сервиса. Сформированные выборки используются для обучения моделей, сопоставляющих уровень услуг со значениями метрик ТКС. В последующем текущие значения метрик используются для оценки уровня мультимедийных услуг с использованием моделей распознавания ситуации.

Для решения поставленной задачи целесообразно использовать метод потенциальных функций [1], применяемый при распознавании образов с учителем и без учителя. Потенциальная функция $K(x, y)$ – функция двух переменных, где x и y – точки пространства X [1], определена на всем пространстве X и положительна $(K(x,y) > 0$ для всех $x, y \in X)$. Закрепив источник потенциала в точке $y = x'$, получим функцию $K(x, x')$ одной переменной x. При этом $\max K(x, x') = K(x', x')$ и $\rho(x_1, x') > \rho(x_2, x') \rightarrow K(x_1, x') < K(x_2, x')$, где ρ – расстояние между точками.

При распознавании образов A_j, j = 2,3,…, суммируются функции $K(x, x_i)$ в точках $x_i \in A_j$, определяемых учителем как относящиеся к j-му образу. В этом случае потенциал $\Phi_{A_j}(x)$ каждого образа определяется так:

$$\Phi_{A_j}(x) = \sum_{x_i \in A_j} K(x, x_i). \qquad (5.61)$$

При распознавании точка x^* относится к тому образу A_j, $j = 2,3,\ldots$, потенциал которого $\Phi_{A_j}(x^*)$ в заданной точке максимальный – $\max\limits_j \Phi_{A_j}(x^*)$.

При распознавании двух образов, например, A и B, целесообразно ввести разделяющую функцию f, причем $f \geq 0$ при $x \in A$ и $f < 0$ при $x \in B$. Для разделяющей функции n-е приближение f^n определяется таким образом:

$$f^n = \sum_{x_i \in A} K(x, x_i) - \sum_{x_j \in B} K(x, x_j). \tag{5.62}$$

Метод является итерационным. Точки предъявляются по одной в случайном порядке, и после каждого показа происходит коррекция разделяющей функции. Рекуррентная процедура нахождения очередного приближения f^{n+1} разделяющей функции определяется так [1]:

$$f^{n+1} = q^n f^n + r^n K(x, x^{n+1}), \tag{5.63}$$

при этом параметр q^n чаще всего принимается равным единице, а параметр r зависит от значений f^n и f^* в данной точке. Причем

$$r^n = \gamma_n \left(r(f^n, f^*) + \varepsilon \right), \tag{5.64}$$

где ε – помеха; коэффициент γ_n является либо константой, либо стягивающим множителем; $r(f^n, f^*)$ – функция, указывающая направление корректировки:

$$r(f^n, f^*) \begin{cases} < 0, & f^* < f^n; \\ > 0, & f^* > f^n; \\ = 0, & f^* = f^n. \end{cases} \tag{5.65}$$

Разделяющая функция принимает такие значения:

$$f^*(x_i) \begin{cases} \geq 0, & x_i \in A; \\ < 0, & x_i \in B. \end{cases} \tag{5.66}$$

Рекуррентная формула определения очередного приближения функции имеет вид

$$f^{n+1} = f^n + r^n K(x, x^{n+1}), \tag{5.67}$$

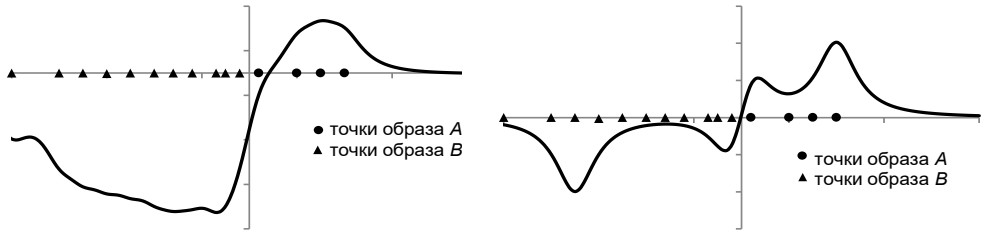
где

$$r^n = \begin{cases} 0, \text{при отсутствии ошибки}; \\ +1, f^n(x^{n+1}) < 0, \quad x \in A; \\ -1, f^n(x^{n+1}) \geq 0, \quad x \in B. \end{cases} \tag{5.68}$$

Аутсорсинг

Корректировка функции выполняется только в тех точках, в которых происходят ошибки распознавания (рис. 5.17,*а*), что позволяет проводить эффективное обучение на несбалансированных выборках (рис. 5.17,*б*). Игнорирование точек, в которых функция правильно осуществляет классификацию, существенно снижает время обучения. В итоге после *n*-го приближения разделяющая функция представляет собой сумму потенциальных функций в данных точках:

$$f^n = \sum_{x_i \in A} K(x,\, x_i) - \sum_{x_i \in B} K(x,\, x_i)\,. \qquad (5.69)$$



а *б*

Рис. 5.17. Вид разделяющей функции: *а* – с учетом всех объектов обучающей выборки; *б* – с учетом только тех объектов, для которых происходят ошибки распознавания

Поскольку метод итерационный, все старые данные сохраняются, а исправляются только ошибки, возникающие при классификации новых данных. Поскольку модель заново не обучается, переобучение происходит быстро.

Алгоритм решения задач методом потенциальных функций:

1. Если вид функции f^* известен, то выбирается $K(x,y)$, раскладывающаяся по той же системе функций, что и f^*. Если вид функции f^* неизвестен, то $K(x,y)$ подбирается интуитивно или экспериментально.

2. Задается первое приближение функции: $f^0 \equiv 0$.

3. Вычисляется следующее приближение: вычисляется значение функции в новой точке $f^n\left(x^{n+1}\right)$ и определяется *r*.

4. Если коэффициент r^n не равен нулю, то сохраняются значения x^{n+1} и r^n.

5. Проводится показ очередной точки x^{n+1} и вычисляется следующее приближение.

6. Остановка может проводиться через *L* показов после последнего исправления (количество *L* задается), либо после показа всей обучающей выборки.

7. Качество полученной модели проверяется показом проверочной выборки.

Пример применения метода потенциальных функций. Рассмотрим разделение двух образов в одномерном пространстве. Пусть образу *A* соответствуют точки, у которых значение координаты $x \geq 0$, образу *B* – точки, у которых значение координаты $x < 0$. В качестве потенциальной функции выберем

$$K\left(x,\, x^{*}\right)=\frac{1}{1+10(x-x^{*})^{2}}\,. \tag{5.70}$$

На рис. 5.18 показаны результаты обучения модели: сплошной линией обозначена разделяющая функция, пунктирной – потенциалы, создаваемые точками.

После 13 исправлений разделение происходит в точке $x=-0,04$ (см. рис. 5.18,*в*), а при использовании такой разделяющей функции f происходят ошибки классификации только в полуоткрытом интервале $-0,04\le x<0$.

Применение метода потенциальных функций для оценки уровня сервиса IPTV. Рассмотрим применение метода потенциальных функций для решения задачи оценки уровня сервиса IPTV в зависимости от значений таких ключевых метрик как джиттер, потеря пакетов и задержка. Согласно [323, 370, 308, 292, 361, 286] эти показатели оказывают наибольшее влияние на качество сервиса IPTV.

При проведении экспериментальных исследований уровень сервиса оценивался оценкой MOS [306, 296, 291]. Таблица данных, в которой каждому набору значений метрик сопоставлялась оценка уровня сервиса, делилась на две выборки – обучающую и проверочную. Проверочная выборка использовалась для определения адекватности, устойчивости и стабильности полученных моделей, а также времени классификации. Для сравнения были построены модели с помощью логистической линейной регрессии, нелинейной регрессии, наивного байесовского классификатора, метода потенциальных функций, деревьев решений с решающим правилом, построенным на основе логистической линейной регрессии, и метода потенциальных функций. В табл. 5.11 приведены время обучения, требуемая память, адекватность и стабильность моделей.

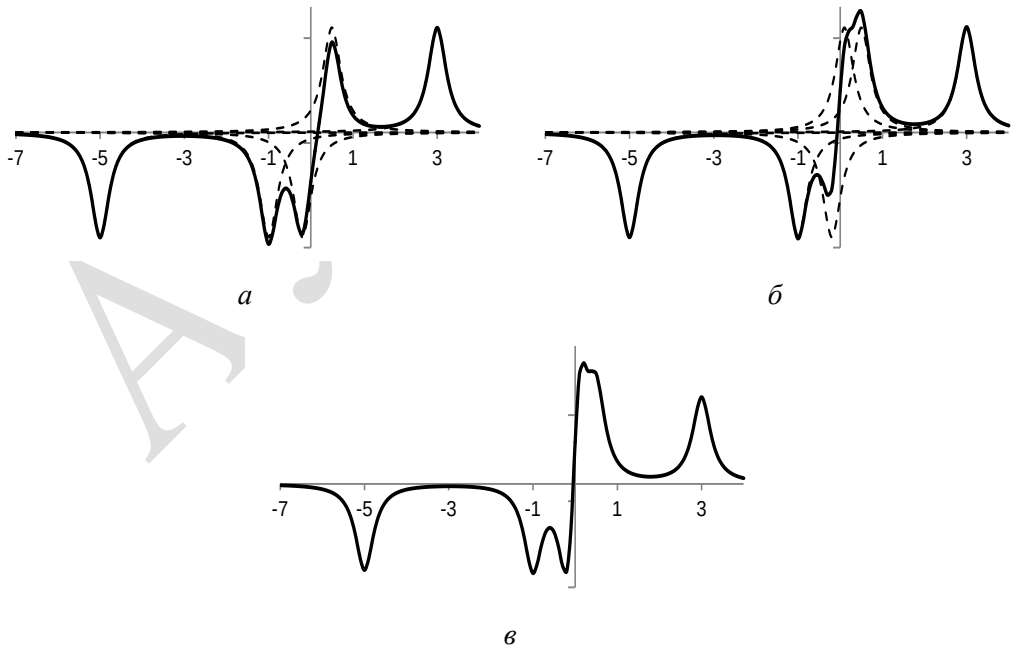


Рис. 5.18. Вид разделяющей функции после: *а* – пяти: *б* – шести;
в – тринадцати исправлений

Таблица 5.11. Результаты эксперимента

Метод распознавания	Время обучения, мс	Требуемая память, кб	Адекватность	Стабильность
Наивный байесовский классификатор	60	569	0,78	0,55
Логистическая линейная регрессия	55	3539	0,79	0,58
Нелинейная регрессия	236	6497	0,65	0,30
Метод потенциальных функций	3	104	0,93	0,80
Деревья решений (решающее правило – метод потенциальных функций)	7	247	0,87	0,78

При оценке качества сервиса IPTV лучшие результаты показали метод потенциальных функций, дерево решений на основе метода потенциальных функций и наивный байесовский классификатор.

Разделяющие поверхности, полученные методом потенциальных функций для разных обучающих выборок, приведены на рис. 5.19. Оценка уровня сервиса проводилась для градаций «удовлетворительно» (область под поверхностью) и «неудовлетворительно» (область над поверхностью). Метрики нормировались относительно 60 мс для джиттера, 2300 мс для задержки и 0,06 для потери пакетов.

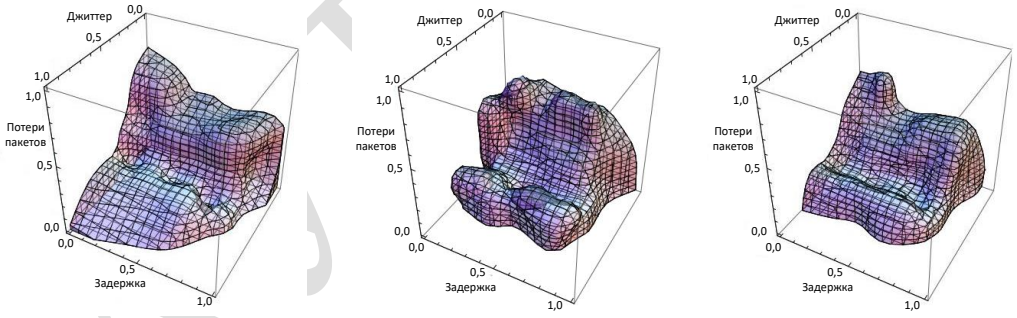


Рис. 5.19. Примеры разделяющих поверхностей, полученных методом потенциальных функций для различных обучающих выборок и разных экспертов

Продолжая разбивать область с удовлетворительным уровнем сервиса на подобласти, можно достигнуть большего количества градаций уровня сервиса. Многослойная фигура (рис. 5.20) соответствует отображению значения метрик в шкалу оценки уровня сервиса от «неудовлетворительно» до «отлично».

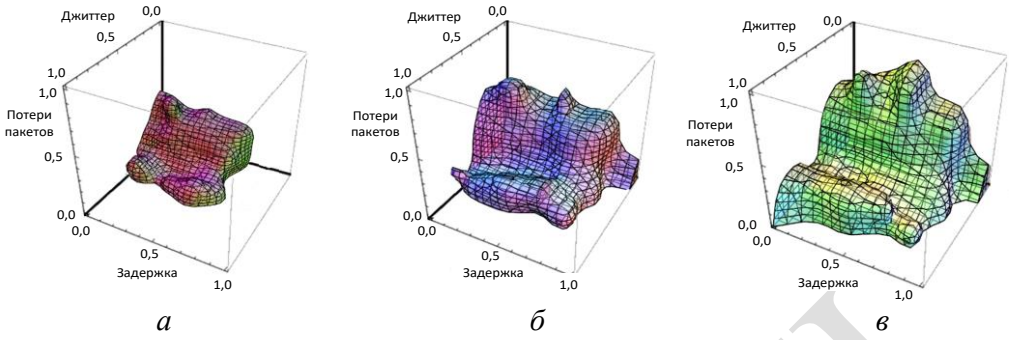


Рис. 5.20. Разделяющие поверхности мультиклассового классификатора, отвечающие оценкам MOS: *а* – 1–2 и 3; *б* – 3 и 4; *в* – 4 и 5. По осям отложены нормированные значения метрик

Жалобы пользователей на качество сервиса создают дисбаланс в сторону точек с неудовлетворительным качеством. Метод потенциальных функций при обучении исправляет разделяющую функцию только в тех точках, в которых происходят ошибки. Это делает метод менее чувствительным к несбалансированным выборкам.

Точки, принадлежащие к разным классам, из обучающей выборки должны поступать бессистемно [17]. В противном случае многие точки, полезные для обучения модели, окажутся проигнорированными. Исследования показали, что двух- и более кратным повторным показом проигнорированных при обучении точек можно улучшить адекватность моделей. На рис. 5.21,*а* и *б* показаны результаты переобучения модели, обученной с точкой-выбросом.

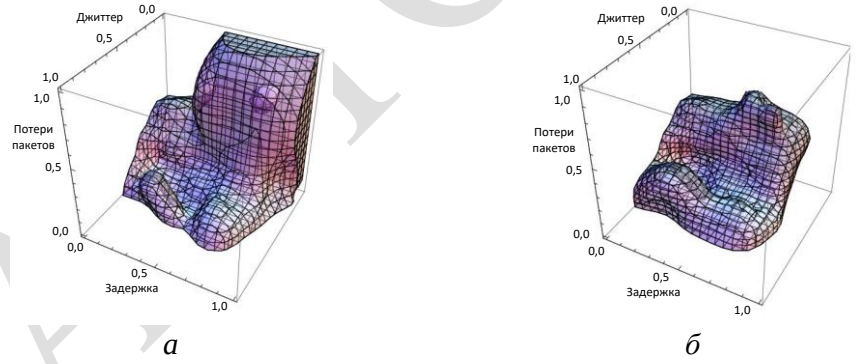


Рис. 5.21. Модель, обученная с выбросом (*а*); исправление ошибок выброса путем переобучения (*б*)

Появление точки-выброса посредине процесса обучения и его неокончательное последующее исправление привело к тому, что адекватность модели составила всего 0,75 (рис. 5.21,*а*). «Раковина» на рисунке соответствует искажению разделяющей функции вследствие появления точки-выброса. «Раздутость» вокруг соответствует появлению исправляющих точек в окрестностях точки-выброса, что еще сильнее исказило разделяющую поверхность. После переобучения путем показа ранее проигнорированных точек адекватность модели возросла до 0,9 (рис. 5.21,*б*).

Таким образом, обоснована возможность оценки качества мультимедийных сервисов на основании анализа значений совокупности ключевых метрик с применением методов теории распознавания образов, в частности метода потенциальных функций. Предложенная реализация метода обладает большой точностью и нечувствительна к несбалансированным выборкам.

5.9. СВЕДЕНИЕ МЕТРИК КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ С ПОМОЩЬЮ АППАРАТА НЕПАРАМЕТРИЧЕСКОЙ СТАТИСТИКИ

5.9.1. Обоснование применимости аппарата непараметрической статистики для сведения метрик

Для эффективного управления ИТ-инфраструктурой необходимо определять показатели качества функционирования распределенных элементов ИТ-инфраструктуры, привлеченных к выполнению возложенных ИУС задач. Такими показателями являются обобщенные метрики, характеризующие определенные аспекты деятельности функциональных единиц ИТ-инфраструктуры, а именно: отдельные задачи, комплексы задач или же целые функциональные подсистемы. Для расчета обобщенных метрик необходимо иметь иерархию сведения оценок, которая содержит ориентированную на реально выполняемые задачи структуру зависимостей между элементами ИТ-инфраструктуры. Зависимости указывают: какие параметры, каких элементов ИТ-инфраструктуры и каким образом влияют на определенную характеристику определенного ее элемента.

Тогда необходимо разработать методы получения показателей функционирования элементов и их сведение на основе иерархического представления ИТ-инфраструктуры в обобщенных метриках, которые комплексно описывают качество выполнения возложенных на ИУС задач. Особый интерес представляют методы расчета и сведения метрик качества функционирования групп элементов, а также обобщение этих метрик до уровня всей ИТ-инфраструктуры единым универсальным способом.

Один из таких методов расчета сводных количественных характеристик для выделенных множеств ОМУ, характеризующих надежность, эффективность и другие аспекты бесперебойного выполнения функций ИТ-инфраструктуры на основе значений параметров функционирования ОМУ, предложен в [124]. Суть метода заключается в сведении группы метрик ОМУ, связанных функциональными зависимостями, к единой обобщенной величине, однозначно характеризующей состояние ОМУ определенной функциональной единицы ИТ-инфраструктуры. В основу метода положен аппарат непараметрической статистики, применяемой до представления текущего состояния ОМУ в евклидовом пространстве их параметров с поэтапным обобщением оценок до уровня ИТ-инфраструктуры в целом согласно ее иерархического представления.

Отсутствие явно выраженной иерархической структуры зависимостей между элементами в современных многоэлементных и многофункциональных ИТ-инфраструктурах ведет к необходимости выделения функциональных групп (ФГ). ФГ – это группа элементов, выделенных из полного состава по признаку участия в выполнении определенной функции. ИТ-инфраструктура может содержать одну или несколько таких групп. В зависимости от решаемых задач группы могут

оставаться неизменными длительное время или подвергаться частым изменениям, как, например, группы отвечающие за дискретно-выполняемые функции. В корпоративных ИТ-инфраструктурах часто выделяют ФГ стратегического управления (выполняющую комплексы задач финансового менеджмента, маркетинга, управления документооборотом и др.), бухгалтерского учета, управления производством и т. д. Каждая из этих ФГ характеризуется уникальностью структуры и особенностью подхода к оценке эффективности функционирования.

Серверные узлы, ПК, системы кабелей и коммутирующего оборудования, офисная периферия, общесистемное и специальное программное обеспечение образуют аппаратно-программные комплексы ИТ-инфраструктуры, которые рассматриваются как ФГ. Интегрированность ФГ в пределах современных предприятий, ориентированных на минимизацию задействованных ресурсов, приводит к смешиванию аппаратно-программных частей подсистем, использованию одной и той же вычислительной и телекоммуникационной техники несколькими ФГ. Например, на одном сервере или ПК могут базироваться приложения для управления документооборотом и ФГ бухгалтерского учета.

Множественность функциональных связей целесообразно представлять графом зависимостей, вершинами которого являются ОМУ – абстрактные сущности, представляющие части ИТ-инфраструктуры, которые помимо вербальной характеристики могут быть описаны набором параметров. На нижнем уровне иерархии ИТ-инфраструктуры ОМУ могут быть аппаратные и программные элементы ИУС, на верхнем – функциональные и технологические подсистемы, выполняющие возложенные на ИУС задачи. Каждый ОМУ содержит сведения о конфигурации и текущем состоянии отдельного элемента ИТ-инфраструктуры. Так, для отдельных компьютеров или серверных узлов параметрами состояния могут быть процессорная емкость, использование оперативной памяти, свопфайла, жестких носителей, текущая скорость чтения/записи контроллера жесткого диска и др. Для веб-серверов в качестве параметров состояния используются пропускная способность, количество удачных объединений между клиентами и веб-сервером в секунду, среднее время отклика веб-сервера, относительная быстродействие интерфейса программирования приложений, количество ошибок в секунду и др. И, наконец, для единицы программного обеспечения можно характеризовать количеством строк кода, циклической сложностью, количеством методов на класс, глубиной наследственности и др. Выделение ОМУ в процессе декомпозиции системы состоит из введения и формального описания логических сущностей в соответствии со значимыми функциональными единицами и формирования наборов свойств, определяющих их состояние.

Совокупность ОМУ и их взаимосвязей образуют ориентированный граф зависимостей между ОМУ. В этом графе ОМУ изображаются вершинами, а направленные ребра характеризуют взаимное влияние объектов, показывая, какие параметры и каких ОМУ нижних уровней иерархии влияют на каждый ОМУ верхнего уровня. В идеальном случае граф ОМУ должен быть деревом, отвечая требованиям связности и ацикличности. Каждой вершине графа ОМУ необходимо поставить в соответствие определенный набор параметров ОМУ. Граф зависимостей определяет множество объектов, возможно влияющих на данный ОМУ, и их параметров, отражает функциональную инфраструктуру. В целом этот граф должен охватить максимум параметров ОМУ различных уровней и соответствовать внутреннему строению ИТ-инфраструктуры. Именно на его основе можно выходить на обобщенные целевые показатели. При этом необходимо

осуществить отображение полученных показателей на качественную шкалу. Именно на графе ОМУ обозначаются ФГ – множества однородных объектов, предназначенных для решения одной или нескольких совместных задач и описываемые одинаковым набором параметров.

Для разработки и реализации метода сведения метрик на основе аппарата непараметрической статистики необходимо в виде графа ОМУ согласовать априорную информацию о взаимодействии зависимых ОМУ, очертить множество параметров, существенных для оцениваемого параметра рассматриваемого объекта, и определить алгоритмы расчета количественных показателей функционирования объекта и их проектирования на качественную ось. Для формирования множества существенных параметров выберем случай, когда оцениваемый объект характеризуется только одной метрикой, которая зависит от всех параметров ОМУ нижнего уровня иерархии. Этот случай легко обобщается на несколько сводных метрик, каждая из которых вычисляется независимо от других. Общность метода требует одинаковости наборов параметров в пределах оцениваемой метрики, т. е. у влияющих ОМУ учитываются наборы одинаковых параметров, значение каждого из которых при необходимости приводится к единому масштабу.

Концептуальные основы сведения метрик с использованием аппарата непараметрической статистики. Качественная оценка выражает понятийную характеристику процесса, привязана к его природе и ожидаемому результату. В каком-то смысле она является субъективной и поэтому трудно поддается формализации и обобщению. Однако качественная оценка должна характеризовать взаимосвязь между элементами ИТ-инфраструктуры и актуальными значениями их метрик. Следовательно, ей все же присуща статистическая природа, что содействует формализации.

Непредсказуемость результатов воздействия широкого спектра факторов на элементы ИТ-инфраструктуры и сложные взаимосвязи между ними не позволяют применить известные параметрические статистические модели. Так, самоподобие моделей систем, поведение которых определяется интернет-активностью [220, 308], множественность, непредсказуемость и разнообразие задач, решаемых средствами ИТ-инфраструктуры, не позволяют как применять параметрические модели, так и прогнозировать поведение элементов на определенные промежутки времени.

Однако среди статистических моделей есть претенденты, которые отвечают описанным условиям применения. В частности, для сведения метрик интерес представляют непараметрические статистические модели, которые обычно основываются на некоторых предположениях о функции распределения и одновременно применяют всю имеющуюся информацию. Среди них наиболее интересна функция статистической глубины [210, 393], поскольку именно она дает оценку подобия, которая сходится к вероятностным показателям. Идеология подхода, основанного на применении функции глубины и ее глубинно-упорядоченных регионов к представлению ФГ в параметрическом евклидовом m-мерном пространстве, где координатными осями служат параметры функционирования элементов ИТ-инфраструктуры, изложена в [373]. Показатель глубины универсален и может быть применен для анализа работы компонентов ИУС, ФПС или ФГ на основе различных наборов параметров.

С помощью функции статистической глубины можно свести множество метрик, характеризующих состояние объектов ФГ, к единому показателю. Этот показатель предоставляет традиционные преимущества работы с одномерными данными – простоту упорядочения или ранжирования, причем независимо от текущего положения группы элементов ИТ-инфраструктуры в параметрическом пространстве. При этом распределение объектов группы в данном пространстве определяется множеством их глубинно-упорядоченных регионов, что позволяет оценивать состояния групп, идентифицировать и локализовать неисправности. При этом возможности анализа расширяются за счет возможности сравнения разных групп объектов того же пространства.

Теперь условие нормального функционирования ФГ можно задать с помощью глубинного региона, например ограничением на его форму, объем или положение. Ограничения на форму позволяют выразить относительную неизменность определенной взаимосвязи между объектами группы, а ограничения на объем должны сдерживать их различия по параметрам, тогда как фиксацией положения некоторого подпространства можно задать зону нормального функционирования ФГ. В последнем случае можно утверждать, что группа исправно выполняет поставленные задачи в случае принадлежности всего региона определенной глубины к зоне ее штатного функционирования.

Особое внимание заслуживают ОМУ, которые в данном параметрическом пространстве находятся вдали от главного облака. Можно предположить, что они являются претендентами на посторонние точки. Для их идентификации предлагается сначала выделять такие допустимые пределы, что объекты, которые им не принадлежат, почти наверняка есть посторонними. Такое подпространство можно отделить, например, гиперплоскостями, перпендикулярными к координатным осям, что обеспечит простоту их задания оператором и высокое быстродействие дифференциации. Кроме того, существуют нотации глубины, способные выделить посторонние точки самостоятельно, например полупространственная [373] и проекционная глубины [225, 353].

Для сведения метрик при управлении ИТ-инфраструктурой выбор нотации функции статистической глубины пришелся на глубину зоноида [275, 302]. Наличие целого ряда свойств делают глубину зоноида очень удобной для оценки функционирования элементов ИТ-инфраструктуры. Среди них в первую очередь выделим аффинную инвариантность, обеспечивающую фильтрацию влияния изменения положения в параметрическом пространстве оцениваемой группы целиком на оценку качества функционирования группы, и непрерывность по глубине и точками. Сами же глубинно-упорядоченные регионы, которые в данной нотации именуются зоноидами, являются выпуклыми. Кроме того, существует аппарат построения выпуклых оболочек зоноидов в пространстве любой размерности [301].

Суть предлагаемого метода сведения метрик определяется следующей последовательностью действий, одни из которых являются подготовительными, а другие выполняются с разной периодичностью.

1. Предварительное группирование элементов, выделение зависимых ОМУ и организация связей влияния ОМУ нижнего уровня иерархии на верхние. Функциональной единицей, к которой применяется метод, является ОМУ O_l, $l = \overline{1, L}$ где L – количество вершин в графе ОМУ.

2. Оценка параметров состояния ОМУ. Состояние каждого l-го ОМУ характеризуется параметрами $P_{l,k},\ k=\overline{1,K_l}$, где K_l – количество параметров l-го ОМУ, каждый из которых характеризует определенный аспект функционирования ОМУ O_l. Значение k-го параметра $P_{l,k}$ ОМУ $O_l,\ l=\overline{1,L}$, определяется значениями параметров множества ОМУ нижнего уровня иерархии, которая составляет оцениваемую группу $S_{l,k}$ параметра $P_{l,k}$, содержащую $N_{l,k}$ однородных ОМУ $X_{l,k,n}$, $n=\overline{1,N_{l,k}}$. Среди параметров группы $S_{l,k}$ выделяются те, которые оказывают влияние на $P_{l,k}$: $P_{l,k,d},\ d=\overline{1,D_{l,k}}$, где $P_{l,k,d}$ – количество параметров оценки функционирования объектов, составляющих оцениваемую группу $S_{l,k}$. Однородность ОМУ означает идентичность набора параметров ОМУ в рамках оцениваемой группы, т. е. состояние параметра $P_{l,k}$ определяется набором $D_{l,k}$ $N_{l,k}$ элементов ИТ-инфраструктуры.

Такое представление обеспечивает общность аппарата, которая предусматривает зависимость различных параметров оцениваемого ОМУ от разных групп параметров одного и того же или разных ОМУ нижних уровней иерархии, при этом чрезмерно не усложняет его, поскольку предельным частным случаем является наличие у ОМУ лишь одного параметра, который определяется всеми параметрами ОМУ единой оцениваемой группы.

3. Интегральная оценка состояния ОМУ. Важнейшую роль в оценке функциональной группы ОМУ играет обобщенный показатель состояния ее объектов, который должен просто и единым образом характеризовать поведение каждого отдельного ОМУ по отношению к другим ОМУ этой ФГ. В параметрическом m-мерном пространстве в качестве такого показателя целесообразно использовать глубину – близость к центру облака в плане эмпирического распределения.

Существует несколько методов оценки глубины – полупространственная глубина, симплексная глубина и глубина зоноида [272]. Для применения в СУИ наиболее привлекательным является последний метод, отличающийся наглядностью и предоставляющий дополнительные сведения, которые могут быть использованы в СУИ для решения различных задач.

Функция глубины – отображение

$$d: E^m \to [0,1], \tag{5.71}$$

обладающее такими свойствами: стремление к нулю в бесконечности от центра облака, непрерывность на множестве эмпирических данных и по распределению, максимальность в центре облака, монотонность относительно всех точек и упорядоченных регионов, а также аффинная эквивариантность [275].

Глубина $\alpha_{l,i}$ точки $x_{l,i}, i=\overline{1,n_l}$, соответствующей объекту, где n_l – количество объектов в ФГ $S_l,\ l=\overline{1,L}$, L – количество ФГ, в пространстве E^m принимает значения из отрезка [0, 1].

Упорядоченный регион – зоноид $Z(\alpha_k, S_l)$ для произвольной глубины α_k может быть определен как

$$Z(\alpha_k, S_l) = conv\left\{ \frac{1}{\alpha_k n}\sum_{j=1}^{k} x_{l,i_j} + \left(1-\frac{k}{\alpha_k n}\right) x_{l,i_{k+1}} : \{i_1,...,i_{k+1}\} \subset N \right\}, \tag{5.72}$$

где $\alpha_k \in \left[\frac{k}{n_l}, \frac{k+1}{n_l}\right]$, $k = \overline{1, n_l - 1}$, $N = \{1, ..., n_l\}$ [272].

Для $\alpha_0 \in \left[0, \frac{1}{n}\right]$

$$Z(\alpha_0, S_l) = conv\left\{x_{l,1}, ..., x_{l,n_l}\right\}. \tag{5.73}$$

Тогда глубина точки $x_{l,i}$, соответствующей объекту ФГ S_l, определяется так:

$$\alpha_{l,i} = \begin{cases} \sup\left\{\alpha : x_{l,i} \in Z(\alpha, S_l)\right\} \forall x_{l,i} : \exists Z(\alpha, S_l) : x_{l,i} \in Z(\alpha, S_l), \\ 0 \text{ — в остальных случаях.} \end{cases} \tag{5.74}$$

С помощью выражений (5.72)–(5.74) многообразие параметров, характеризующих состояние объектов ФГ, сводится к единому показателю $\alpha_{l,i}$, поддающемуся упорядочению и не зависящему от местоположения центральной точки ФГ. Это дает возможность получить распределение объектов в *m*-мерном параметрическом пространстве с последующим выделением подмножеств объектов с наиболее или наименее выраженными состояниями относительно текущего состояния всей ФГ. Распределение ОМУ в пределах ФГ однозначно определяется набором упорядоченных регионов. Это позволяет достаточно легко решать задачи локализации неисправностей в ФГ, сравнивая ее с другими ФГ того же пространства.

Таким образом, метод сведения метрик оценки качества функционирования ОМУ с применением аппарата непараметрической статистики позволяет сводить разнотипные параметры, влияющие на качество функционирования ОМУ, к универсальным обобщенным показателям. Задача обобщения показателей группы ОМУ решается путем их представления в параметрическом пространстве с возможностью дальнейшего проецирования на качественную ось.

5.9.2. Метод нечеткой непараметрической оценки качества функционирования элементов ИТ-инфраструктуры

Определение состояния ОМУ существенно усложняется необходимостью учета значений большого количества параметров, влияющих на качество работы ОМУ. Опасность получения за счет больших затрат точных оценок, которыми можно не успеть распорядиться, способствует использованию обобщенных качественных оценок функционирования ОМУ. Возникает потребность в методах получения обобщенных качественных оценок преобразованием значений комплекса параметров, характеризующих отдельные аспекты функционирования ОМУ, в один или несколько показателей качества работы. При этом часто отсутствуют четкая априорная информация о характере зависимостей между ОМУ и строгие оценки качества их функционирования.

Сбор информации о значениях параметров функционирования ОМУ может быть выполнен с помощью специализированных протоколов SNMP, CMIP и т. п., посредством агентской технологии, встроенного в операционные системы инструментария управления, например WMI, индикаторов производительности –

Performance Counters и пр. В работе [104] предложено оценивать работу элементов и подсистем ИУС по интегральному показателю качества функционирования. Недостатком данного метода является сложность поиска оценочных функций параметров, необходимых для вычисления численного значения показателя качества. В [145] предложен подход к сведению количественных и качественных метрик для оценки сетей ОТС. Недостатком предложенного метода является то, что при оценке не рассматривается история состояний сетевых элементов, а учитывается только текущее состояние. В [87] использован поход к оценке состояния отдельных элементов ИУС или их функциональных групп с помощью единственного параметра, однако не учитывается логическая структура взаимозависимостей компонентов ИУС.

В современных СУИ автоматизируются в основном функции мониторинга и анализа функционирования элементов ИТ-инфраструктуры, в то время как управление обычно осуществляется администраторами вручную [104]. Существенно повысить эффективность управления можно, используя в СУИ модели и методы, основанные на актуальной оценке параметров функционирования ОМУ.

Поэтому необходимо разработать и использовать в ИТ УИТИ методы получения обобщенной качественной оценки функционирования ОМУ с учетом истории значений как собственных параметров, так и параметров зависимых ОМУ, оказывающих влияние на качество работы рассматриваемого ОМУ. Ниже описан один из таких методов – метод оценки качества функционирования ОМУ в режиме реального времени с учетом логической структуры зависимостей между ОМУ, предложенный в [109].

Метод нечеткой непараметрической оценки качества. В течение некоторого промежутка времени параметры функционирования ОМУ принимают различные значения. Отклонениями от нормы считаются систематические выходы значений за границы зон качественного функционирования. Случаи единичного или нерегулярного выхода значений за установленные пределы полагаются временным ухудшением качества работы элементов, не требующим управляющих воздействий.

При этом для реагирования ИТ УИТИ только на серьезные неполадки необходимо, чтобы СУИ анализировала не текущее состояние элементов ИУС, а историю значений параметров качества функционирования в течение некоторого промежутка времени T. В этом случае для определения необходимости формирования управляющего воздействия целесообразно использовать метод качественной оценки функционирования элементов ИУС [109].

Формирование множества выходных величин как оценок параметров функционирования ОМУ. При оценке работы отдельного ОМУ o_a, $a = \overline{1, A}$, на протяжении определенного интервала времени T учитываются собственные параметры ОМУ $P_{a,d}$, $d = \overline{1, D_a}$, влияющие на качество функционирования o_a, и обобщенные оценки функционирования других ОМУ, оказывающих влияние на состояние данного ОМУ. Подобного рода зависимости описываются логической структурой ИТ-инфраструктуры, иерархия которой представляется в виде ориентированного графа. Расчет качества функционирования по такой иерархической структуре проводится снизу вверх, когда для определения качества функционирования ОМУ верхних уровней предварительно определяется качество функционирования ОМУ нижних уровней, оказывающих влияние на данный ОМУ.

Совокупность значений выбранных параметров наряду с обобщенными оценками образует множество выходных величин Y_n, (n – номер контура управления), на основании анализа которых СУИ, при необходимости, вырабатывает вектор управляющих воздействий U_n.

В течение заданного интервала времени T проводится M замеров значений параметров подмножества P_{Y_n} множества Y_n, сформированного собственными параметрами элементов, в результате чего формируется множество $\{P_{Y_n}^m\}$, $m = \overline{1, M}$. Подмножество $\{Q_n\} = Y_n \setminus P_{Y_n}$ состоит из значений обобщенных оценок работы ОМУ, полученных расчетным путем за интервалы функционирования, адаптированные к режимам деятельности ОМУ. Все значения выходных параметров нормируются и приводятся к отрезку [0, 1].

Концептуальные положения метода. При отсутствии сведений о характере зависимостей между ОМУ и качественных оценок параметров ОМУ, а также возможности существования любого рода отношений между ОМУ целесообразно применять приближенные экспертные оценки, основанные на личном опыте администраторов, руководителей ИТ-подразделений и других экспертов. Оценки такого рода зависят от функциональной нагрузки ИТ-инфраструктуры, определяются при развертывании СУИ и ее настройке и задаются посредством пользовательского интерфейса с помощью функций принадлежности аппарата нечеткой логики [390]. Для анализа временных зависимостей между параметрами отдельных ОМУ используются геометрические оценки аппарата непараметрической статистики, проецирование которых на зоны, определяющие качество функционирования, порождает обобщенную оценку параметров ОМУ. При этом с помощью аппарата нечеткого вывода с учетом влияния элементов нижних уровней иерархии вычисляется качественная оценка функционирования (КОФ). Предлагаемый в [109] двухэтапный метод позволяет получить статистическую оценку функционирования ОМУ без каких-либо предположений о вероятностных характеристиках процессов, протекающих в ИТ-инфраструктуре, а также обобщить ее на основе опыта экспертов, спроецировав на качественную ось.

Представление ОМУ в параметрическом пространстве состояний требует аналитической обработки данных в евклидовом пространстве, которая позволяет выделить зону наиболее характерного функционирования, соответствующую центральному региону. Из определений [393] центральных упорядоченных регионов наиболее подходящей является нотация зоноида [302]. Во-первых, аффинная ее эквивариантность привязывает оценку положения к ОМУ. Во-вторых, полнота информации обеспечивает уникальность оценки. В-третьих, непрерывность по глубине, определяющая центральность региона, и распределению обеспечивает устойчивость решения к входным данным. В-четвертых, выпуклость упрощает расчет долей принадлежности. И, наконец, наличие алгоритмического аппарата построения самого зоноида, являющегося выпуклым многогранником, составляет набор полезных свойств нотации зоноидов, предопределивших ее выбор.

Расчет обобщенной оценки параметров функционирования ОМУ. Представление ОМУ в параметрическом пространстве состояний требует аналитической обработки данных в евклидовом пространстве, которая позволяет выделить

зону наиболее характерного функционирования, соответствующую центральному региону. Предположим, на качество функционирования ОМУ o_a, $a=\overline{1,A}$ где A – количество ОМУ, оказывают влияние элементы множества

$$E_a=\{o_{a,1},o_{a,2},...,o_{a,C}\} \tag{5.75}$$

ОМУ нижних иерархических уровней и D_a собственных параметров ОМУ o_a:

$$P_a=\{p_{a,1},p_{a,2},...,p_{a,Da}\}. \tag{5.76}$$

Диапазоны значений КОФ всех элементов o_a и собственных параметров $p_{a,d}$, $d=\overline{1,D_a}$, разбиваются на поддиапазоны, для каждого из которых определяется значение лингвистической переменной. Каждому параметру соответствует $S_{a,d}$ лингвистических значений $l^P_{a,d,s}$, $s=\overline{1,S_{a,d}}$, а каждому диапазону значений КОФ сопоставляется B_a лингвистических значений $l^E_{a,b}$, $b=\overline{1,B_a}$ Заполняется нечеткая база знаний, в которой набору лингвистических значений $\{l^E_{i,b},l^P_{a,d,s}\}$, $i:e_i\in E_a$ ставится в соответствие лингвистическое значение из множества $\{l^E_{a,b}\}$.

Определим КОФ ОМУ o_a нижнего уровня иерархической структуры, на работу которого не влияют другие ОМУ, т. е. $E_a=\varnothing$. В этом случае КОФ получается обобщенной оценкой параметров функционирования o_a. Для этого в течение интервала T значения параметров из множества P_a замеряются M раз, в результате чего формируется M множеств вида

$$P_{a,m}=\{p_{a,1,m},p_{a,2,m},...,p_{a,Da,m}\},\ \ m=\overline{1,M}. \tag{5.77}$$

Рассмотрим D_a-мерное пространство E^{D_a}, в котором каждая ось соответствует отдельному собственному параметру функционирования o_a, а каждому $P_{a,m}$ соответствует точка пространства, отображающая состояние собственных параметров o_a во время m-го замера. Для эмпирического распределения на пространстве E^{D_a} зоноид определяется с помощью выражений (5.72)–(5.74) и выглядит так:

$$Z_a(\alpha)=\begin{cases} conv\{\dfrac{1}{\alpha\cdot M}\sum\limits_{j=1}^{\beta}P_{a,i_j}+(1-\dfrac{1}{\alpha\cdot M})P_{a,i_{\beta+1}}\},\ \text{для}\ \alpha\in[\dfrac{\beta}{M},\dfrac{\beta+1}{M}]; \\ conv\{P_{a,1},P_{a,2},...,P_{a,M}\},\ \text{для}\ \alpha\in(0,\dfrac{1}{M}], \end{cases} \tag{5.78}$$

где α – глубина зоноида, $\beta\in\{1,...,\ M-1\}$, а $\{i_1,...,i_{\beta+1}\}\subset\{1,...,M\}$.

Полученный регион выделяет в E^{D_a} область, в которой со статистической точки зрения наибольшая вероятность пребывания точек, характеризующих состояние ОМУ, а глубина региона пропорциональна риску выхода таких точек за пределы $Z_a(\alpha)$. Так как все оси рассматриваемого D_a-мерного пространства разбиты на диапазоны, то гиперплоскости, перпендикулярные координатным осям

АУТСКИЙ

и проходящие через точки границ таких диапазонов, разбивают полученный зоноид на выпуклые замкнутые D_a-мерные доли. Каждой доле соответствует определенное высказывание из нечеткой базы знаний, на основании которых и определяется обобщенная оценка параметров o_a.

Для всех $b = \overline{1, B_a}$ рассчитывается сумма относительных D_a-мерных объемов долей зоноида, принадлежащих подпространствам, соответствующим нечетким высказываниям:

$$V_{a,b} = \frac{\sum_{j=1}^{R_{a,b}} V_{a,b,j}}{V_a}, \tag{5.79}$$

где $P_{a,b}$ – количество неповторяющихся нечетких высказываний вида $\bigcap_{d=\overline{1,D}} l^P_{a,d,s}$, $s = \overline{1, S_{a,d}}$, правил базы знаний, выходной величиной которых является лингвистическое значение $l^E_{a,b}$, $V_{a,b,j}$ – D_a-мерный объем выпуклой замкнутой доли зоноида, соответствующей j-му высказыванию, а V_a – D_a-мерный объем $Z_a(\alpha)$.

$Q_a \in [0,1]$ – численное значение КОФ a-го элемента, определяется центроидным методом:

$$Q_a = \frac{\int_0^1 q \cdot F_a(q) dq}{\int_0^1 F_a(q) dq}, \tag{5.80}$$

где $F_a(q)$ – выходное нечеткое множество, результат объединения функций принадлежности выхода, которые соответствуют лингвистическим значениям $l^E_{a,b}$, $b = \overline{1, B_a}$, усеченным с помощью операции импликации [285]:

$$F_a = \underset{i=\overline{1,Ba}}{agg} \left(\int_0^1 imp(V_{a,i}, \mu_{a,i}(q)) dq \right). \tag{5.81}$$

Здесь *agg* – агрегирование нечетких множеств, наиболее часто реализуемое операцией нахождения максимума, *imp* – импликация, обычно реализуемая как операция нахождения минимума, $\mu_{a,i}(q)$ – функция принадлежности выхода, соответствующая $l^E_{a,i}$.

Определение КОФ элемента логической структуры ИТ-инфраструктуры. Определим КОФ o_a с учетом влияния других ОМУ, т. е. $E_a \neq \varnothing$, приняв обобщенные оценки элементов множества E_a равными $Q^*_a = \{Q_{a,1}, Q_{a,2}, ..., Q_{a,C}\}$.

Тогда значения КОФ данного ОМУ o_a рассчитывается по формуле (5.80), но с той разницей, что

АУТСКИЙ

$$F_a(q) = \underset{i=\overline{1,B_a}}{agg}\left(\int_0^1 imp(V_{a,i}, \mu_{a,i}(q), G_{a,i}(Q_a^*))dq\right), \quad (5.82)$$

где $G_{a,i}(Q_a^*)$ – степени принадлежности Q_a^* лингвистическим значениям $l_{a,b}^E$, $b = \overline{1, B_a}$, которые рассчитываются по формуле

$$G_{a,i}(Q_a^*) = \underset{\varpi=\overline{1,R_{a,i}}}{\vee}\left(\underset{j=\overline{1,C}}{\wedge}[\mu_{j,\gamma}(Q_{a,j})]\right). \quad (5.83)$$

Здесь $\wedge$ – операция *t*-нормы [175], $\vee$ – операция *t*-конормы, $R_{a,i}$ – количество неповторяющихся нечетких высказываний $\underset{c=\overline{1,C}}{\cap} l_{z,b}^E$, $z : o_z = o_{a,c}$ для всех $b = \overline{1, B_z}$ в правилах базы знаний, выходной величиной которых является $l_{a,i}^E$, γ – номер лингвистического значения $l_{z,\gamma}^E$, участвующего в таком правиле.

Пример применения предлагаемого метода определения КОФ. Рассчитаем качественную оценку функционирования Q_1 ОМУ o_1, на состояние которого влияют два параметра функционирования: «Загруженность процессора» ($p_{1,1}$) и «Объем свободной ОЗУ» ($p_{1,2}$). Каждому из параметров поставлено в соответствие по три лингвистических значения $\{l_{1,1,1}^P, l_{1,1,2}^P, l_{1,1,3}^P\}$ и $\{l_{1,2,1}^P, l_{1,2,2}^P, l_{1,2,3}^P\}$, а ОМУ o_1 – $\{l_{1,1}^E, l_{1,2}^E, l_{1,3}^E\}$. Двумерное пространство параметров изображено на рис. 5.22.

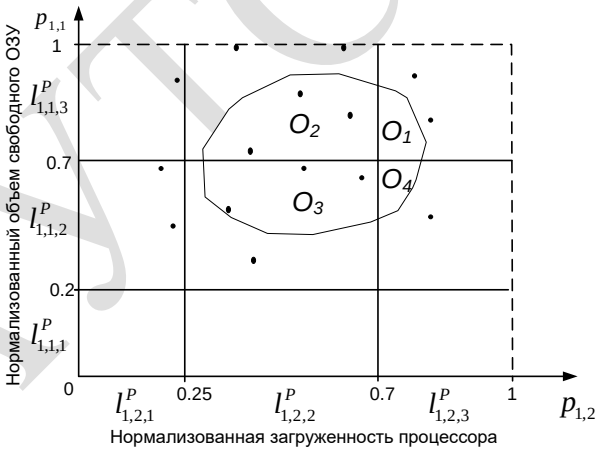


Рис. 5.22. Пример зоноида для расчета КОФ в пространстве параметров

Рассчитывается Z_1 по (5.78) [301] и разбивается гиперплоскостями, изображенными на рис. 5.22 прямыми, нормальными к осям и проходящими через точки границ диапазонов значений параметров $p_{1,1}$ и $p_{1,2}$, на доли $\{O_1, O_2, O_3, O_4\}$, где доле O_1 соответствует нечеткое высказывание $l_{1,1,3}^P \cap l_{1,2,3}^P$, доле O_2 – $l_{1,1,3}^P \cap l_{1,2,2}^P$ и т. д.

АУТСКИЙ

Таблица 5.12. Нечеткая база знаний элемента e_1

Нечеткие высказывания	Лингвистические значения e_1
$l^P_{1,1,3} \cap l^P_{1,2,2} \cup l^P_{1,1,3} \cap l^P_{1,2,3} \cup l^P_{1,1,2} \cap l^P_{1,2,3}$	$l^E_{1,3}$
$l^P_{1,1,3} \cap l^P_{1,2,1} \cup l^P_{1,1,2} \cap l^P_{1,2,2} \cup l^P_{1,1,3} \cap l^P_{1,2,3}$	$l^E_{1,2}$
$l^P_{1,1,2} \cap l^P_{1,2,1} \cup l^P_{1,1,1} \cap l^P_{1,2,1} \cup l^P_{1,1,1} \cap l^P_{1,2,2}$	$l^E_{1,1}$

В табл. 5.12 приведена нечеткая база знаний, на основании которой определяется КОФ o_1. Согласно (5.79) $V_{1,1}=0$, $V_{1,2}=\dfrac{S(O_3)}{V_1}$, $V_{1,3}=\dfrac{S(O_1)+S(O_2)+S(O_4)}{V_1}$, где $S(O_h)$, $h=\overline{1,4}$ – двумерный объем (площадь) h-й доли Z_1, а V_1 – двумерный объем Z_1. Выходное нечеткое множество $F_1(q)$ получается подстановкой полученных величин в формулу (5.81), а значение Q_1 для o_1 рассчитывается по формуле (5.80).

Таким образом, метод качественной оценки функционирования ОМУ основан на анализе истории значений их параметров с помощью аппарата непараметрической статистики и нечеткой логики. Представление ОМУ в евклидовом параметрическом пространстве состояний наряду с применением алгоритмов статистической геометрии позволяет вырабатывать нечеткие выводы с учетом наиболее характерных временных интервалов деятельности ОМУ. Использование механизмов нечеткого вывода делает возможным гибкий учет влияния ОМУ нижних уровней иерархического представления ИТ-инфраструктуры на оцениваемый ОМУ и обобщение его поведения во времени.

Предлагаемый метод позволяет СУИ формировать управляющее воздействие с учетом качества функционирования ОМУ, что делает целесообразным его использование в контурах управления, ориентированных на повышение качественных показателей производительности ИТ-инфраструктуры.

ВЫВОДЫ К ГЛАВЕ 5

1. Для принятия обоснованных управленческих решений по поддержанию согласованного уровня услуг разработан комплекс моделей и методов анализа состояния элементов ИТ-инфраструктуры и сведения метрик показателей качества. Разработанный комплекс позволяет выявлять элементы с атипичным состоянием или атипичной динамикой изменения состояния, определять состояние выше расположенных по иерархии элементов ИТ-инфраструктуры на основании состояния элементов нижних уровней, производить качественную оценку состояний элементов, определять фактический уровень сервисов.

2. С целью создания в СУИ простой унифицированной системы сбора, накопления и обработки информации о ИТ-инфраструктуре предложен метод кодирования состояний ее элементов на основе общей схемы кодирования различных признаков. Это позволило систематизировать оценку качества функционирования элементов ИТ-инфраструктуры, сделать оценку качества широкого класса элементов величиной, допускающей измерение.

АУИТКН

3. Для фрагментов ИТ-инфраструктуры, в которых структура связей и степень влияния ОМУ не изменяется во времени, предложен аналитический метод оценки состояния ОМУ по интегральному показателю качества функционирования. Это позволило существенно упростить оценку состояния ОМУ.

4. Предложен метод оценки уровня обслуживания пользователей ИТ-инфраструктуры путем сведения количественных и качественных метрик, отличающийся комбинированным применением методов экспертных оценок и подходов на основе теории нечетких множеств, обеспечивающий получение результатов за приемлемое время. Применение СНЛВ со взвешенной истинностью обеспечивает учет индивидуальных весовых коэффициентов критериев оценки метрик для каждого эксперта при применении экспертным комитетом единой базы правил СНЛВ.

5. Разработан метод оценки уровня телекоммуникационных сервисов по результатам анализа значений совокупности метрик, измеренных в узлах телекоммуникационной сети. Оценка уровня сервиса проводится с применением теории распознавания образов, в частности, метода потенциальных функций. Применение предлагаемого метода позволяет оценить уровень телекоммуникационных сервисов по обобщенным показателям функционирования телекоммуникационной сети. Тем самым создается возможность повышения конкурентоспособности предоставления телекоммуникационных услуг за счет возможности управления уровнем услуг с минимальными затратами.

6. Предложен метод сведения метрик оценки качества функционирования элементов с применением аппарата непараметрической статистики, что позволяет сводить разнотипные параметры, влияющие на качество функционирования элементов ИТ-инфраструктуры, к обобщенному показателю. Обобщение показателей элементов функциональной группы осуществляется посредством их представления в едином параметрическом пространстве. При этом сохраняется возможность дальнейшего проецирования обобщенных показателей на качественную ось с применением непараметрических моделей. Применение предложенного метода сведения метрик позволяет контролировать качество функционирования элементов ИТ-инфраструктуры и своевременно реагировать на изменения состояния элементов.

7. Предложен метод качественной оценки функционирования элементов ИТ-инфраструктуры на основе анализа истории значений их параметров с применением аппарата непараметрической статистики и нечеткой логики. Представление элементов в евклидовом параметрическом пространстве состояний наряду с применением алгоритмов статистической геометрии позволяет с помощью механизма нечеткого вывода учесть характерные временные интервалы деятельности элементов, игнорируя случаи нерегулярного выхода значений за установленные пределы. Предлагаемый метод дает возможность формировать управляющее воздействие с учетом качества функционирования элементов ИТ-инфраструктуры, игнорируя случаи единичного или нерегулярного выхода значений за установленные пределы.

ГЛАВА 6

МЕТОДЫ ОЦЕНКИ ВЛИЯНИЯ НЕИСПРАВНОСТЕЙ НА КАЧЕСТВО ПРЕДОСТАВЛЕНИЯ ИТ-УСЛУГ

В главе решается важная для управления уровнем услуг задача обнаружения неисправностей в ИТ-инфраструктуре и оценки влияния неисправностей на качество предоставления услуг пользователям. Для обнаружения неисправностей применяется метод на основе глубинно-упорядоченных регионов, а для оценки влияния неисправностей на качество услуг используется оригинальный продукционно-алгоритмический подход.

6.1. ПРИМЕНЕНИЕ ГЛУБИННО-УПОРЯДОЧЕННЫХ РЕГИОНОВ ДЛЯ ВЫЯВЛЕНИЯ НЕИСПРАВНЫХ ЭЛЕМЕНТОВ ИТ-ИНФРАСТРУКТУРЫ

6.1.1. Анализ методов поиска неисправностей в ИТ-инфраструктурах

Быстрое обнаружение проблем и оперативное устранение неисправностей в ИТ-инфраструктуре базируется на информации о состоянии не только отдельных элементов, но и групп элементов, образующих функциональные, технологические и другие подсистемы ИУС. Чтобы в условиях жестких временных требований на основе этой информации осуществлять анализ состояния элементов ИТ-инфраструктуры, СУИ должна обладать соответствующим комплексом моделей, методов и алгоритмов. Для анализа состояния ОМУ, сгруппированных по функциональному признаку, применяется метод на основе глубинно-упорядоченных регионов.

При определении управляющих воздействий, поддерживающих ИТ-инфраструктуру в работоспособном состоянии, обеспечивающих распределение ограниченных ИТ-ресурсов и решающих другие задачи, СУИ должна различать ухудшение параметров функционирования, происходящее в результате неполадок в ИТ-инфраструктуре от периодических функциональных отказов, которые являются следствием причин естественного характера. Так, увеличение количества пользователей приложения приводит к увеличению времени обработки стандартных запросов, которое СУИ фиксирует как снижение производительности серверов приложений. Точно также увеличение количества пользователей, генерирующих запросы, осуществляющих обмен файлами или мультимедийной информацией, приводит к увеличению загруженности каналов связи, которая может стать причиной перегрузки узлов сети, что тоже будет зафиксировано СУИ. В таких случаях для дифференциации отказов серверов или элементов сети от функциональных отказов анализируются дополнительные признаки, такие как количество пользователей, время суток и пр. В результате анализа этих признаков становится понятным, следует ли оперативно реагировать на ухудшение ка-

чества функционирования ИУС или просто зафиксировать этот факт для последующего его использования в качестве аргумента при подготовке руководителем ИТ-подразделения обоснований необходимости модернизации ИТ-инфраструктуры.

В [104, 116] предложен подход к анализу состояния элементов, основанный на кооперации значений отдельных параметров ОМУ с учетом их важности. При этом требуется анализ дополнительных признаков при диагностировании и локализации неисправностей, что существенно усложняет алгоритмы работы СУИ.

Для выявления неисправных ОМУ в UNIX-подобных системах может быть использован способ сведения метрик производительности [89], осуществляющий выделение метрик производительности, подготовку матрицы ковариации, выделение исходных факторов, смещение факторов, идентификацию и сохранение значимых метрик производительности. Однако этому способу присущи недостатки, ограничивающие его сферу применения. Во-первых, сведение метрик указанным способом не учитывает информацию об условиях функционирования ИТ-инфраструктуры. Во-вторых, факторы как результат анализа не несут физического смысла и не могут быть использованы при оценке ОМУ.

Способ анализа параметров функционирования ОМУ, предложенный в работе [90], включает последовательно выполняемые выбор параметров ОМУ, измерение значений выбранных параметров, статистическую обработку значений параметров, проверку результатов обработки на принадлежность к установленным пределам и выявление неправильно функционирующих ОМУ. В данном способе на этапе статистической обработки значений параметров выполняют совместный анализ отклонений данных, полученных на этапе измерения значений выбранных параметров. Предпочтение отдается среднему значению, математическому ожиданию, медиане, минимальному и максимальному значениям и стандартному отклонению, определяемым на заданном временном промежутке. На этапе проверки результатов обработки на принадлежность к установленным границам средние значения по определенным подмножеством параметров сравнивают со стандартным отклонением, определяемым по нескольким группам за промежуток времени, значительно превышающий промежуток, принятый для этапа статистической обработки значений параметров. Считается, что принадлежность средних значений параметров определенных подмножеств одному или двум стандартным отклонениям свидетельствует о нормальной работе системы. ОМУ, не удовлетворяющие этим требованиям, выделяют на этапе выявления неправильно функционирующих ОМУ.

Этот способ имеет ряд преимуществ. Во-первых, может быть учтена информация об условиях функционирования ИТ-инфраструктуры и отфильтрованы данные, вызванные общесистемными проблемами. Во-вторых, результаты статистической обработки значений параметров несут определенную физическую информацию. Метод обладает существенными недостатками, которые ограничивают его применение в СУИ. В частности, значения параметров ОМУ не включают информацию об ОМУ низшего уровня иерархии. Кроме того, на этапе выявления неправильно функционирующих ОМУ не происходит их упорядочения по критичности.

Перспективным является предложенный в работах [86, 116] подход к определению состояний элементов ИТ-инфраструктуры, в котором акцент сделан не

на анализе изменения значений показателей работы отдельных ОМУ, а на выявлении элементов группы, динамика изменения состояния которых отличается от динамики изменения состояния основной массы подобных элементов.

Именно разработке такого подхода посвящен данный параграф. По существу речь идет о разработке подхода к анализу состояния ОМУ и поиску неисправностей на основе выявления объектов и групп объектов с атипичной динамикой изменения состояния.

6.1.2. Применение глубинно-упорядоченных регионов для выявления объектов с атипичной динамикой изменения состояния

СУИ обеспечивает мониторинг состояния и управление ОМУ, образующими иерархическую структуру. Представление ИТ-инфраструктуры в виде иерархии ОМУ позволяет применять объектный подход к анализу состояния ОМУ. При этом ФГ однородных ОМУ образуют агрегированные ОМУ, представляющие собой совокупности элементов с одинаковыми наборами свойств и характерной динамикой поведения. Такие ФГ включают, например, серверы ФПС и ТПС, телекоммуникационные узлы и другие группы оборудования, состояния которых характеризуются такими параметрами как среднее время обслуживания запросов, загруженность каналов связи и пр. В любой модели управления важным фактором является качественная и количественная оценка состояния ОМУ с обобщением до уровня подсистем.

Для описания ОМУ, являющихся функциональными группами, S_l, $l=\overline{1,L}$, где L – количество ФГ в ИТ-инфраструктуре, в параметрическом пространстве целесообразно применить аппарат статистической геометрии, позволяющий сводить состояние ОМУ к единственному показателю, легко задавать ограничения относительно оцениваемых параметров, фиксировать изменения поведения объектов и ФГ. Задание состояний ОМУ на множестве глубинно-упорядоченных регионов приводит к определению порогов на языке эвклидовой геометрии. При нечетко заданных критериях или неполной информации о режиме нормального функционирования пороги можно адаптировать к различным режимам функционирования группы ОМУ, опираясь только на понятия регионов и глубины. Функциональные группы ОМУ составляются так, чтобы их объекты обладали одинаковыми наборами свойств для размещения в одном параметрическом пространстве. Для ФГ учитываются особенности поведения в параметрическом пространстве, характерном для входящих в группу объектов.

Предложенный в [87, 116] метод анализа параметров функционирования ОМУ предполагает:

1) вычисление значений параметров ОМУ верхнего уровня иерархии ОМУ как глубины α менее глубокого зоноида $Z(\alpha_k)$, не выходящего за ограничения T, заданные в параметрическом пространстве оценочного множества параметров;

2) выполнение проверки результатов обработки на принадлежность к установленным границам как проверки вхождения всех вершин зоноида $Z(\alpha_k)$ в установленные для α_k границы T и упорядочения неправильно функционирующих ОМУ по критичности в соответствии с показателем глубины в параметрическом пространстве оценочного множества параметров.

Благодаря этому появляются возможности учета информации об ОМУ ниже расположенного уровня иерархии при расчете значений параметров ОМУ и взаимосвязи внутри оцениваемой группы ОМУ во время их анализа; упорядочения неправильно функционирующих объектов по критичности.

В моделях управления важный фактор – качественная и количественная оценка состояния ОМУ с обобщением до уровня ФГ и ИТ-инфраструктуры в целом, когда функциональная группа ОМУ также является объектом с набором обобщенных признаков, свойственных множеству ОМУ группы и определяемых на основании параметров составляющих ее объектов. При этом происходит не только селекция параметров, но и фильтрация их трендов, таких как суточные колебания загруженности ресурсов, переходные процессы начала и окончания рабочего дня, сезонные девиации использования ресурсов и пр. Отклонения от нормального режима, наряду с обобщением влияния отклонения на верхние уровни иерархии агрегирующих ФГ вплоть до уровня ИТ-инфра-структуры в целом, должны быть обнаружены на ранней стадии.

Параметрическое пространство функциональной группы ОМУ – евклидово m-мерное пространство E^m, осями которого являются выбранные для данной ФГ параметры. Состояние каждого ОМУ в определенный момент времени однозначно характеризуется точкой этого пространства, а множество ОМУ ФГ представляется облаком точек, описывающим состояние ФГ относительно выбранных параметров. Состояние ФГ в целом может быть аппроксимировано математическим ожиданием точек-объектов – центральной точкой группы. Траектория движения этой точки – история изменения средних значений параметров ОМУ выделенной группы.

На выбросы – ОМУ, точки которых в параметрическом пространстве отдалены от основного облака ФГ, можно посмотреть с противоположных точек зрения. С одной стороны, учет выбросов при оценке состояния ФГ дает объективное представление о текущем состоянии всей группы, с другой – вносит значительную погрешность в оценку центральной точки группы. Поэтому оба аспекта учета выбросов имеют право на независимое рассмотрение и, соответственно, требуют инструменты анализа. Существуют нотации глубины, способные выделить выбросы самостоятельно, например полупространственная [373] и проекционная глубины [225, 353].

Для определения принадлежности объекта к выбросам предлагается сначала выделять множество объектов группы, состояния которых находятся в допустимых пределах. Такие объекты не относятся к выбросам, а входят в подпространство нормального функционирования – подпространство m-мерного параметрического пространства, ограниченного гиперплоскостями, проведенными перпендикулярно к параметрическим осям. Все объекты, не относящиеся к этому подпространству, являются выбросами. Такой способ обеспечивает высокую скорость дифференциации ОМУ по их состоянию.

Кроме оценки состояния ФГ с точки зрения работоспособности или корректности выполнения определенных задач, когда оценка состояния ОМУ и ФГ проводится по анализу значений m параметров, можно подобным образом обеспечить контроль эффективности, надежности и других показателей функционирования ИТ-инфраструктуры. Для этого необходимо выделить параметры, на основе которых будет проводиться оценка состояния ОМУ и ФГ. Количество выбранных параметров определит размерность соответствующего евклидового

пространства. Затем следует задать подпространство нормального функционирования. Подпространство нормального функционирования можно очерчивать гиперплоскостями сложной конфигурации, а при рассмотрении нелинейных зависимостей влияния значений параметров на состояние ОМУ – гиперповерхностями соответствующей формы.

Применение такого подхода к центральной точке позволяет идентифицировать отклонение работы ФГ от нормального режима при выходе последней из подпространства.

6.1.3. Использование глубины пространства состояния объектов

Условие нормального функционирования ФГ S_l можно определить как принадлежность зоноида глубины $\alpha_{н.ф.}$ подпространству нормального функционирования $\Theta_{н.ф.}$. Причем $\alpha_{н.ф.}$ определяется как

$$\alpha_{н.ф.} = \arg\left\{ \min_{\alpha_k \in [\frac{1}{n};1]} \left\{ Z(\alpha_k) : Z(\alpha_k) = conv\left\{ \frac{1}{\alpha_k n} \sum_{j=1}^{k} x_{i_j} + \left(1 - \frac{k}{\alpha_k n}\right) x_{i_{k+1}} : \right.\right.\right. \\ \left.\left.\left. : \{i_1, ..., i_{k+1}\} \subset N \right\}, Z(\alpha_k) \in \Theta \right\} \right\}. \quad (6.1)$$

Пусть подпространство $\Theta_{н.ф.}$ задано ограничивающими гиперплоскостями в m-мерном пространстве параметров. Исходя из выпуклости зоноида по определению, можно утверждать, что группа S_l нормально функционирует, если все вершины зоноида находятся внутри подпространства нормального функционирования, т. е. выполняется условие

$$Z(\alpha_{н.ф.}, S_l) \in \Theta_{н.ф.} \quad (6.2)$$

В режиме нормального функционирования группы S_l для проекций v_{ij} каждой i-й вершины, $i = 1,2,\ldots$, зоноида $Z(\alpha_{н.ф.}, S_l)$ на j-ю ось, $j = \overline{1,m}$, будет выполняться условие

$$v_{ij} \in \left[Q_{j\min}, Q_{j\max} \right], \quad (6.3)$$

где $Q_{j\min}$ и $Q_{j\max}$ – значения координат гиперплоскостей на j-й оси, ограничивающих подпространство нормального функционирования снизу и сверху соответственно.

Для локализации и управления устранением неисправностей в пределах ФГ S_l целесообразно использовать особенности пространства параметров. Объекты, чьи точки выходят за пределы подпространства $\Theta_{н.ф.}$, будем считать неисправными. Их упорядочение по глубине позволяет упорядочить восстановительные мероприятия – в первую очередь внимание должно уделяться объектам, у которых глубина меньше.

Учет градиента глубины позволяет на ранней стадии обнаружить движение отдельного объекта ФГ S_l в сторону от центра подпространства $\Theta_{н.ф.}$.

АУТСКП

Если абсолютное значение градиента для каждого ОМУ не превышает допустимой величины $\Delta\dot{\alpha}_{l,i\,доп.}$, т. е. показатель глубины не претерпевает существенных изменений во времени, то градиент глубины позволяет отследить динамику подпространства $\Theta_{н.ф.}$. При этом границы, выделяющие подпространство $\Theta_{н.ф.}$, должны смещаться вместе с центральной точкой группы S_l и ее глубинно-упорядоченными регионами.

Если глубина $\alpha_{l,i}$, определяемая из выражения (5.74), не выходит за пределы допустимого отрезка изменения глубины для каждого объекта группы S_l, т. е. выполняется условие

$$\alpha_{l,i} \in \left[\alpha_{l,i\min}, \alpha_{l,i\max}\right], \tag{6.4}$$

то считается, что ФГ S_l сохраняет состояние нормального функционирования. При этом возможно значительное изменение значений параметров объектов, не влекущее за собой существенного изменения глубины. Это позволяет обеспечить инвариантность по отношению к суточным и сезонным колебаниям в работе ИТ-инфраструктуры и переходным процессам различного рода.

Показатель глубины является универсальным и его можно применять для анализа работы объектов ФГ ИТ-инфраструктуры с различными наборами параметров.

6.1.4. Метод выявления неисправных элементов ИТ-инфраструктуры

Как отмечалось выше, выявлению неисправностей предшествует сбор данных о функционировании элементов ИТ-инфраструктуры. Собранные на протяжении периода T, определяемого текущей динамикой группы, данные представляются точками пространства размерности $D_{l,k}$, осями которого служат параметры объектов группы.

Выделим два вида ОМУ – с рассчитываемыми и с измеряемыми параметрами. При этом часть параметров ОМУ первого вида может измеряться. Значения параметров ОМУ первого вида рассчитываются и сохраняются в памяти. Значения параметров ОМУ второго вида, а также параметров ОМУ первого вида, которые не являются расчетными, получаются измерением или выполнением предусмотренных тестовых проверок. Здесь не рассматривается механизм отсеивания посторонних точек, поскольку он зависит от предназначения элементов ИТ-инфраструктуры и текущей реализации и может быть определен администратором.

Схема реализации метода, поясняющая его суть, приведена на рис. 6.1. Последовательность этапов достаточно традиционна, но их содержание определяется описанными выше концептуальными положениями.

Следует заметить, что представлению ОМУ в $D_{l,k}$-мерном параметрическом пространстве предшествует их приведение к единому масштабу и координатной сетке или проекция на относительную шкалу, например отрезок [0, 1] реальной оси.

Фундаментальная особенность метода состоит в выделении подпространства $\Theta_{l,k}$ нормального функционирования ОМУ группы $S_{l,k}$. Для этого отсекается в

виде ограничений-гиперповерхностей некоторая зона $D_{l,k}$-мерного параметрического пространства. В простейшем случае границами могут быть гиперплоскости, ортогональные координатным осям. Они и задают зону, выход за пределы которой является критическим, поскольку означает нарушение штатного режима функционирования ОМУ.

Рис. 6.1. Схема реализации метода выявления неисправных ОМУ

Расчет состояния параметра функционирования $S_{l,k}$ выполняется как оценка вероятности выхода параметров функционирования произвольного ОМУ $X_{l,k,n}$ группы $S_{l,k}$ за пределы подпространства нормального функционирования.

Пусть $X_{l,k,n} = \left(x_{l,k,n,1}, x_{l,k,n,2}, ..., x_{l,k,n,D_{l,k}} \right)^T$, где $x_{l,k,n,d}$ – значение d-го параметра n-го ОМУ группы $S_{l,k}$. Тогда состояние параметра рассчитывается по формуле

$$x_{l,k} = \begin{cases} \inf \left\{ \alpha_m : Z\left(\alpha_m, S_{l,k}\right) \in \Theta_{l,k} \right\} \forall \alpha_m \in [0,1] : \exists Z\left(\alpha_m, S_{l,k}\right) \in \Theta_{l,k}, \\ \gamma_{l,k,н.в.} \text{ — в других случаях,} \end{cases} \quad (6.5)$$

где $x_{l,k}$ – значение параметра $P_{l,k}$, $\alpha_m \in \left[\frac{m}{n}, \frac{m+1}{n}\right]$, $m = 1,...,n-1$, $\gamma_{l,k,н.в.}$ – отклонение группы $S_{l,k}$ от зоны нормального функционирования и значения $\alpha_{l,k,\max}$ $Z\left(\alpha_m, S_{l,k}\right)$ – зоноид, который определяется следующим образом.

Если выполняется условие $\alpha_m \in \left[\frac{1}{N_{l,k}}, 1\right]$, то для определения $Z\left(\alpha_m, S_{l,k}\right)$ на основе выражения (5.72) используем выражение

$$Z\left(\alpha_m, S_{l,k}\right) = \operatorname{conv}\left\{\frac{1}{\alpha_m N_{l,k}} \sum_{j=1}^{m} X_{l,k,i_j} + \left(1 - \frac{m}{\alpha_m N_{l,k}}\right) X_{l,k,i_{m+1}} : \{i_1,...,i_{m+1}\} \subset \{1,...,N_{l,k}\}\right\}, \quad (6.6)$$

а если $\alpha_m \in \left[0, \frac{1}{N_{l,k}}\right]$, то выражение (5.73).

Определенные значения параметров $P_{l,k}$ проверяются на принадлежность установленным пределам. Для этого можно использовать гарантированное качество обслуживания, вероятность доступа к ресурсам или иные характеристики, предусмотренные в SLA. Точно также можно использовать показатели, определение эмпирическим путем в рамках оцениваемой группы $S_{l,k}$. По наличию отклонения, т. е. при превышении параметром $x_{l,k}$ верхнего предела $x_{l,k} > \alpha_{l,k,\max}$ на установленное процентное соотношение, необходимо задействовать механизмы устранения возможных неисправностей. В частности, для этого применяется балансировка нагрузки в пределах группы. Предварительно нужно получить рекомендации о порядке осуществления таких действий. Тогда для каждого ОМУ $X_{l,k,n} \notin \Theta_{l,k}$ на основе выражения (5.74) вычисляется значение

$$\beta_{k,l,n} = \begin{cases} \sup\left\{\beta_{k,l,n} : X_{l,k,n} \in Z\left(\beta_{k,l,n}, S_{l,k}\right)\right\} \forall X_{l,k,n} : \exists Z\left(\beta_{k,l,n}, S_{l,k}\right) : X_{l,k,n} \in Z\left(\beta_{k,l,n}\right), \\ 0 – \text{в других случаях.} \end{cases}$$

(6.7)

Действия по устранению неисправностей в первую очередь применяются к ОМУ, которые имеют меньшие значения $\beta_{k,l,n}$, что особенно критично в случае выполнения условия $x_{l,k} = \gamma_{l,k,н.в.}$.

Сведение метрик выполняется однократно и повторяется только в случае изменений в ИТ-инфраструктуре, при этом часто достаточно применять структурирование только к измененной части ИТ-инфраструктуры. Это касается и алгоритмов, построенных на определении масштаба или относительной оси. Операции измерения и численного преобразования требуют выполнения на каждой итерации алгоритма реализации метода. Ограничение подпространства нормального функционирования $\Theta_{l,k}$, а также величин $\gamma_{l,k,н.в.}$ и $\alpha_{l,k,\max}$ могут уточняться в соответствии с целями и задачами, возложенными на ИТ-инфраструктуру.

АУТСКИЙ

6.1.5. Особенности алгоритмической реализации метода

Применение математического аппарата глубинно-упорядоченных регионов стало возможным благодаря разработанным алгоритмам расчета глубины [227], а также регионов для плоскости [226] и пространства любой размерности [301].

Наиболее затратным по объему вычислений является этап определения $Z\left(\alpha_m, S_{l,k}\right)$, поскольку он может потребовать многократного выполнения расчетов. Однако вопрос применения алгоритмов оптимизации остается открытым и во многом зависит от требуемой точности расчетов и доступного времени. Поэтому обычно ограничиваются проверкой выполнения условия

$$Z\left(\alpha_m, S_{l,k}\right) \in \Theta_{l,k}. \tag{6.8}$$

В простейшем случае границы подпространства нормального функционирования $\Theta_{l,k}$ могут быть ортогональными к координатным осям параметрического пространства гиперплоскостями, соответствующими верхним и (или) нижним ограничениям на ресурсы без учета возможной связи между ними. Тогда подпространство нормального функционирования является прямоугольным $D_{l,k}$-мерным параллелепипедом, а проверка вхождения фигуры $Z\left(\alpha_m, S_{l,k}\right)$ в него упрощается, так как $Z\left(\alpha_m, S_{l,k}\right)$ является выпуклым многогранником. Это позволяет определять вхождение $Z\left(\alpha_m, S_{l,k}\right)$ путем проверки выполнения каждого неравенства вида

$$\theta_{l,k,d,\min} \le z\left(\alpha_m, S_{l,k}\right)_{q,d} \le \theta_{l,k,d,\max} \; \forall q = 1,\ldots,Q\left(Z\left(\alpha_m, S_{l,k}\right)\right),\; d = 1,\ldots,D_{l,k}, \tag{6.9}$$

где $z\left(\alpha_m, S_{l,k}\right)_{q,d}$ – проекция *q*-й вершины зоноида $Z\left(\alpha_m, S_{l,k}\right)$ на *d*-ю ось параметрического подпространства, $Q\left(Z\left(\alpha_m, S_{l,k}\right)\right)$ – количество вершин фигуры $Z\left(\alpha_m, S_{l,k}\right)$, $\theta_{l,k,d,\min}$ и $\theta_{l,k,d,\max}$ – соответственно минимальная и максимальная границы по *d*-й оси прямоугольного $D_{l,k}$-мерного параллелепипеда нормального функционирования.

Если существует связь между параметрическими ограничениями подпространства нормального функционирования, $\Theta_{l,k}$ может принимать сложные формы. Тогда алгоритм проверки выполнения условия (6.8) необходимо разрабатывать отдельно для каждого такого случая.

Если $D_{l,k} = 2$, то расчет вершин и граней $Z\left(\alpha_m, S_{l,k}\right)$ может быть выполнен с помощью алгоритмов, приведенных в [301] или [226]. В первом случае оценка временной сложности алгоритма примет вид

$$O\left(D_{l,k}^2 \cdot \left(D_{l,k}^2 + N_{l,k}\right) \cdot F\left(N_{l,k}, D_{l,k}\right)\right),$$

где $F\left(N_{l,k}, D_{l,k}\right)$ – среднее число граней зоноида, а оценка памяти – вид

$$O\left(F\left(N_{l,k}, D_{l,k}\right) \cdot D_{l,k} + C\right),$$

где C – константа. Во втором случае временную сложность алгоритма можно оценить как $O\left(N_{l,k}^2 \cdot \log N_{l,k}\right)$. Вычисление глубины может быть выполнено на основе современных методов внутренней точки или с помощью декомпозиции Данцига-Вульфа [226], эффективной в этом случае.

Особое внимание заслуживает вопрос определения периодичности повторения расчетов. Параметры $P_{l,k}$ следует пересчитывать при изменении значений измеряемых или расчетных параметров ОМУ группы $S_{l,k}$. Учитывая, что измеряемые параметры требуют постоянного мониторинга и интеграции в течение периода T, эффективными оказываются интеллектуальные механизмы определения частоты расчета [81].

Реализация метода и примеры. Алгоритм метода реализован и включен в виде модуля в ИТ УИТИ. Для представления ИТ-инфраструктуры в виде графа ОМУ, группирования объектов и их параметров, а также установления границ подпространства нормального функционирования разработаны эффективные средства. При этом расчетами можно легко управлять, используя интуитивный интерфейс клиентской части. Для автоматизации сбора данных с РОУ использовано агентское ПО. Испытания показали применимость метода и подтвердили правильность оценок затрат времени и памяти.

На рис. 6.2 приведена оценка состояния ФГ, состоящей из 15 ПК с установленными агентами ИТ УИТИ. Малое количество переданных агентом пакетов указывает на отклонение ОМУ от режима нормального функционирования. Чрезмерное использование сетевого интерфейса в течение длительного времени свидетельствует о возможной несанкционированности трафика, неполадках в работе агентов или наличие стороннего ПО на рабочей станции.

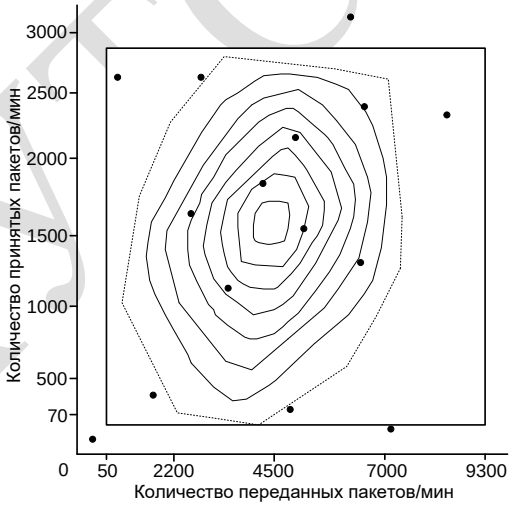


Рис. 6.2. Оценка состояния ФГ рабочих станций с установленными агентским модулями относительно сетевого интерфейса

Рис. 6.3 демонстрирует оценку той же группы по балансу использования свопа и ОЗУ. На рисунках сплошной линией определены границы штатного функционирования, а пунктирной – зоноид наибольшей глубины, принадлежащий установленным пределам.

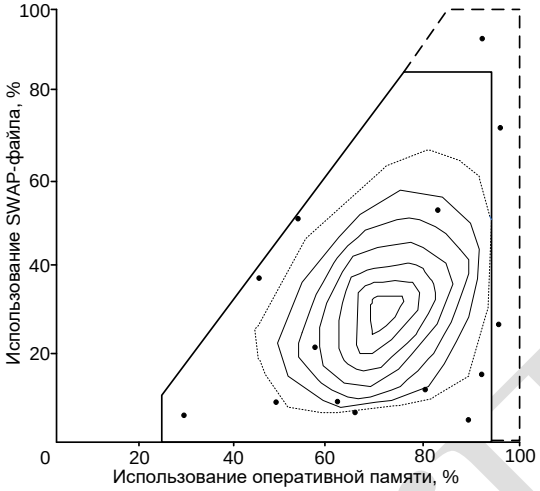


Рис. 6.3. Оценка состояния ФГ рабочих станций относительно баланса использования свопа и оперативной памяти

Оценка загруженности группы серверов технологии SaaS приведена на рис. 6.4.

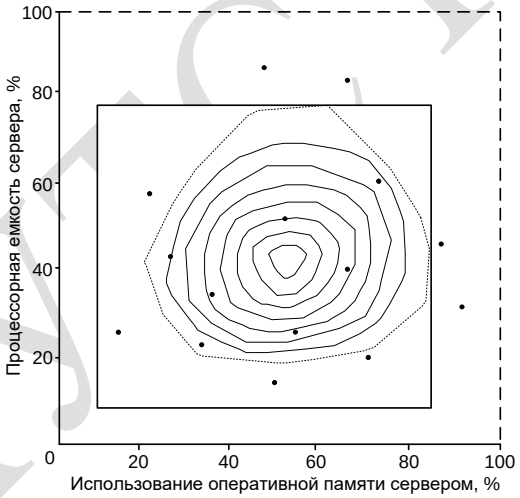


Рис. 6.4. Оценка состояния группы серверов SaaS

Низкая загруженность ресурсов позволяет перенести сервер на менее производительную ВМ или свернуть его, а высокая – указывает на необходимость разнесения выполняемых на нем задач или выделения дополнительных ресурсов.

Таким образом, рассмотренный метод выявления неисправных ОМУ по анализу динамики изменения состояния элементов функциональной группы позволяет достаточно просто выделить ОМУ с атипичной динамикой состояния для последующего управления устранением неисправностей.

АУЭС РП

6.2. АЛГЕБРАИЧЕСКИЙ ПОДХОД К ОПРЕДЕЛЕНИЮ РАСПРОСТРАНЕНИЯ ВЛИЯНИЯ НЕИСПРАВНОСТЕЙ В СЕТИ ДОСТУПА НА КАЧЕСТВО ПРЕДОСТАВЛЯЕМЫХ СЕРВИСОВ

6.2.1. Влияние неисправностей в сетях связи на качество телекоммуникационных услуг

В сфере телекоммуникационных услуг на доходы ОТС непосредственное влияние оказывает качество обслуживания абонентов. Провайдеры коммуникационных услуг, кабельного и спутникового телевидения, провайдеры доступа в интернет или услуг в интернете, операторы мобильной связи несут существенные финансовые убытки из-за потери клиентов, недовольных обслуживанием, или штрафных санкций, накладываемых при нарушении непрерывности предоставления услуг или временном предоставлении некачественного сервиса. В тоже время стабильно высокое качество обслуживания, разнообразие предоставляемых услуг, дифференциация обслуживания клиентов и другие факторы способствуют повышению среднемесячного дохода на абонента. Этот показатель определяется средним доходом от абонента за фиксированный интервал времени, который обычно равен одному месяцу, и используется ориентированными на потребителя ИТ-организациями, предоставляющими не только телекоммуникационные, но и ИТ-услуги.

Одним из ключевых компонентов телекоммуникационной инфраструктуры являются сети доступа, посредством которых обеспечивается доступ абонентов к телекоммуникационным сервисам. Крупные ОТС могут насчитывать миллионы клиентов в сетях доступа: затраты на создание, модернизацию и поддержание работоспособности сетей доступа составляют значительную часть бюджета ОТС.

Большое количество абонентов и необходимость снижения затрат на обслуживание своей инфраструктуры вынуждают ОТС производить дифференциацию обслуживания абонентов, уделяя больше внимания наиболее важным из них. В этом случае при обнаружении СУИ неисправностей в коммуникационной системе определяется степень влияния этих неисправностей на качество предоставления услуг абонентам, и в отношении наиболее важных из них предпринимаются дополнительные мероприятия по обслуживанию, в то время как остальные клиенты довольствуются номинальным обслуживанием.

Для реализации эффективного дифференцированного подхода к обслуживанию абонентов в случае возникновения неисправностей в коммуникационной системе необходимо обеспечить в СУИ поддержание жизненного цикла снижения потерь из-за ухудшения качества предоставляемых услуг, состоящего из следующих этапов:

- выявление и локализация неисправностей;
- определение их влияния на качество предоставления услуг;
- выделение наиболее важных абонентов, пострадавших в результате получения некачественных услуг;
- проведение дополнительных мероприятий по отношению к этим абонентам.

6.2.2. Проблемы определения влияния неисправностей в сетях доступа на качество предоставления сервиса абонентам

Сети ОТС можно разделить на два больших класса – магистральные сети передачи данных и сети доступа. Основу глобальной информационной инфраструктуры составляют магистральные сети передачи данных, предназначенные для обеспечения связи между крупными телекоммуникационными узлами или региональными подсетями. Магистральные сети, являясь ядром больших сетевых инфраструктур, связывают крупные телекоммуникационные узлы или региональные подсети. Магистральные сети используют каналы связи с большой пропускной способностью, объединяют информационные потоки большого количества подсетей и отличаются повышенной надежностью.

Не менее важным компонентом телекоммуникационной инфраструктуры являются сети доступа операторов, посредством которых абонентам предоставляются разнообразные информационные и телекоммуникационные услуги. В сетях доступа крупных ОТС могут насчитываться миллионы абонентов, а затраты на создание, модернизацию и поддержание работоспособности таких сетей составляют значительную часть бюджета ОТС [161]. Каждый из клиентов имеет определенный набор параметров, принадлежит к определенному классу, имеет определенный набор требований к уровню обслуживания, зафиксированному в SLA.

Когда в сети доступа появляются неполадки, у клиентов снижаются показатели качества обслуживания. Для того чтобы реализовать политику оператора по отношению с клиентами, когда появляются неполадки, службе работы с клиентами нужны показатели снижения уровня обслуживания для каждого клиента или наиболее важных из них. Расчеты указанных показателей зависят от многих параметров, прежде всего сервисов, технологий, применяемых ОТС, типа оборудования, структуры сети. При этом логике осуществления расчетов присуща изменчивость. А время, за которое необходимо выполнить указанные расчеты, достаточно ограничено.

Возникает проблема представления сетей доступа и организации расчетов показателей ухудшения качества обслуживания клиентов вследствие неполадок в сети в реальных условиях функционирования [333].

Многообразие и большое количество оборудования, существенная протяженность линий связи, высокая стоимость резервирования, жесткие требования к экономичности обслуживания и другие факторы делают сети доступа существенно менее надежными, чем магистральные сети. Большие штрафные санкции и угроза потери клиентов вынуждают операторов немедленно реагировать на неисправности, оказывающие непосредственное влияние на качество предоставляемых услуг. Наряду с оперативным обнаружением и локализацией неисправностей со стороны СУИ и их устранением службой эксплуатации сетей доступа подразделение оператора, занимающееся обслуживанием клиентов, проводит предупредительные мероприятия по отношению к важным клиентам, призванные нивелировать негативные последствия неисправностей, влияющих на качество услуг. Эти мероприятия включают в себя своевременное информирование клиентов о возникших неполадках, переключение на резерв или предоставление альтернативных источников сервиса и другие действия с целью максимально сгладить недовольство и негативный эффект от некачественного обслуживания у наиболее важных клиентов.

Поэтому очень важно быстро определять влияние неисправностей в сети доступа на качество абонентского обслуживания. В то же время большое количество абонентов ОТС, иногда насчитывающее миллионы юридических и физических лиц, громоздкость и сложность коммуникационных систем, содержащих до десяти и более иерархических уровней в сетях доступа, большое количество типов и единиц коммутационного оборудования, непростые зависимости влияния неисправностей на качество сервиса, сложная интерференционная картина воздействия совокупности неисправностей и другие факторы превращают задачу импакт-анализа для отдельного абонента в чрезвычайно трудоемкую задачу, решение которой требует больших вычислительных ресурсов. Трудность обусловлена большими размерами инвентаризационной базы данных оператора, в которой поиск взаимосвязей может потребовать существенного времени, в то время как подразделение обслуживания клиентов должно начать реагировать через считанные минуты после возникновения неисправностей, предпринимая предопределенные действия по отношению к наиболее важным клиентам.

Поэтому необходимо создать методы и простые модели импакт-анализа, позволяющие ПИА максимально быстро получать картину распространения неисправностей по телекоммуникационной сети от проблемного узла до клиентского оборудования.

Учитывая большое количество разнообразных сервисов в сети, создать универсальную модель для оценки влияния неисправностей на все сервисы одновременно не представляется возможным, поэтому следует задействовать совокупность различных моделей, каждая из которых учитывает особенности отдельных или группы родственных сервисов.

Однако, целесообразно разработать общий подход, который позволит определить структурно-функциональную схему взаимодействия различных моделей импакт-анализа. По меньшей мере этот подход должен помочь разработчикам и пользователям ПИА определить последовательность действий, построить механизм подключения моделей, уточнить характеристики, влияющие на распространение неблагоприятных последствий неисправностей.

Ниже описывается предложенный в [153] подход к решению задачи влияния неисправностей в сетях доступа на качество предоставления услуг абонентам, который построен на определенных правилах распространения влияния неисправностей, учитывающих особенности структуры сети и сервисов, используемые технологии и другие важные факторы.

6.2.3. Суть алгебраического подхода к оценке влияния неисправностей на качество предоставляемого сервиса

Из всех неисправностей, выявленных в сети доступа, ПИА выделяет только те, которые оказывают влияние на качество предоставления сервиса. Остальные неисправности, которые обнаруживаются СУИ, игнорируются при определении влияния на качество сервиса, но информация о них используется службой эксплуатации для их устранения в соответствии с определенными регламентами и с учетом их значимости для коммуникационной системы. Например, отказ резервного узла в кластерной системе или ненагруженного резервного канала связи снижает надежность коммуникационной системы ОТС, но не отражается на качестве предоставления сервиса клиентам.

АЛТГТУ

Качество предоставления каждого сервиса определяется на основе значений группы параметров, оговоренных в SLA. Потому сначала создаются упрощенные схемы сети доступа, построенные с точки зрения возможного влияния неисправностей в узлах на значения отдельных параметров, зафиксированных в SLA. Эти схемы учитывают особенности распространения влияния неисправностей на эти параметры от места возникновения неисправности через промежуточные узлы до абонентского оборудования.

Далее для каждого типа узлов прописываются правила, нивелирующие особенности отдельных сетевых технологий, связывающие тип неисправности, произошедшей в узле, со степенью влияния на значения отдельных групп или группы однородных параметров. Для каждого типа узлов определяются правила распространения влияния неисправностей на значение рассматриваемого параметра, характеризующего качество сервиса, в направлении от узлов, расположенных выше в иерархии сети, к узлам, расположенным ниже в иерархии сети или к абонентскому оборудованию. После этого последовательно от узла верхнего уровня, в котором произошла неисправность, для каждого из зависимых узлов расположенных ниже уровней рассчитываются коэффициенты снижения значений показателей качества сервиса до тех пор, пока не будет достигнут уровень абонентского оборудования. Для этого оборудования рассчитываются значения коэффициентов, определяющих степень снижения качества предоставляемых услуг относительно нормативного значения.

Если в сети доступа возникла только одна неисправность, то для анализа распространения ее влияния не рассматриваются пути, исходящие с верхнего уровня на смежный нижний уровень, в которых значением коэффициентов снижения качества предоставляемых услуг можно пренебречь. В таких случаях говорят о затухании влияния неисправностей.

Если значение коэффициента, определяющего степень снижения качества предоставляемого ОТС, на выходе абонентского оборудования превышает пороговую величину, задаваемую для каждого абонента или класса абонентов в зависимости от их важности, то ПИА вырабатывает информационные сообщения, поступающие в подразделение обслуживания клиентов для реагирования в соответствии с заранее прописанным регламентом.

6.2.4. Оценка влияния неисправностей на показатели производительности сети доступа

Рассмотрим случай, когда в качестве параметров, влияние на которые будет оцениваться при возникновении неисправностей, выбираются показатели производительности сети доступа. К таким параметрам относится, например, скорость доступа абонентов к сети интернет. Этот параметр относится к категории важнейших в связи с тем, что снижение скорости доступа негативно влияет на качество предоставления других услуг, таких как качество передачи голосовых сообщений сервиса VoIP.

Для импакт-анализа, когда в качестве параметров используются показатели производительности, достаточно использовать единственную схему, аналогичную схеме, приведенной на рис. 6.5. Эта схема представляет собой ориентированный граф, вершины U которого соответствуют узлам сети доступа, таким как маршрутизаторы, коммутаторы, мультиплексоры доступа и др., ветви – линиям связи, а листья A – абонентскому оборудованию.

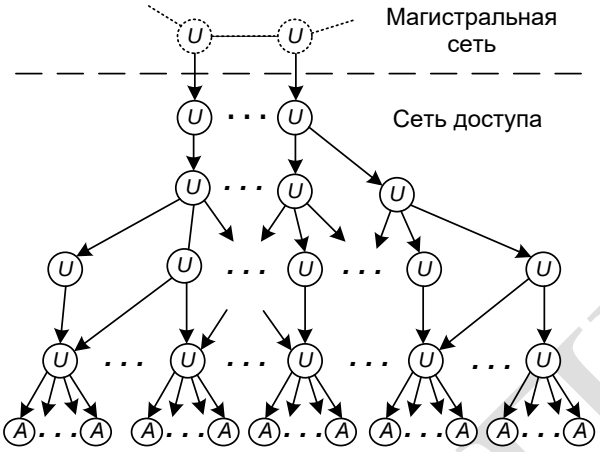


Рис. 6.5. Схема соединений узлов в сети доступа

Схема, приведенная на рис. 6.5, позволяет установить пути прохождения потоков данных от узлов к абонентскому оборудованию, определить поражаемые участки сети в результате распространения последствий неисправностей, получить интерференционную картину при возникновении нескольких неисправностей и т. п.

Эта схема, по сути, определяет лишь связность узлов сети доступа для передачи трафика. Такой схемы достаточно для решения задачи импакт-анализа при рассмотрении сетей передачи данных, функционирующих на уровнях не выше второго уровня модели OSI. Однако ее не достаточно при сервисно-ориентированном подходе к определению влияния неисправностей. Так, в настоящее время превалируют мультисервисные сети, построенные с использованием IP-протокола, и для оценки влияния неисправностей в сети доступа на качество предоставления сервиса абонентам необходимо учесть состояние различных служб, обеспечивающих работоспособность сервисов. В последнем случае дополнительно к схеме, приведенной на рис. 6.5, необходимо использовать схему, аналогичную приведенной на рис. 6.6. На этой схеме перечислены сервисы, которыми пользуется абонент, и службы, обеспечивающие работу этих сервисов.

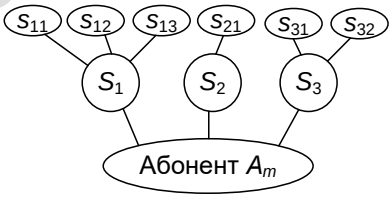


Рис. 6.6. Используемые абонентом A_m сервисы

В качестве примера на рис. 6.6 сервисы S_1, S_2, S_3, которыми пользуется m-й абонент A_m, поддерживают службы $s_{11},\ldots,s_{32}$. Такая схема составляется для каждого важного абонента A_m, $m=\overline{1,M}$, где M – количество важных абонентов ОТС, или для группы абонентов, и позволяет быстро определить влияние неисправностей в элементах сетевой инфраструктуры, обеспечивающих работу отдельных

служб. Так, в случае отказа элемента, который обеспечивает работоспособность службы s_{21}, например почтового сервера, прекращается предоставление сервиса S_2, например электронной почты, что немедленно ощутят абоненты $A_{i(S_2)}$, $i(S_2) = 1,\ldots,N(S_2)$ пользующиеся этим сервисом (рис. 6.7).

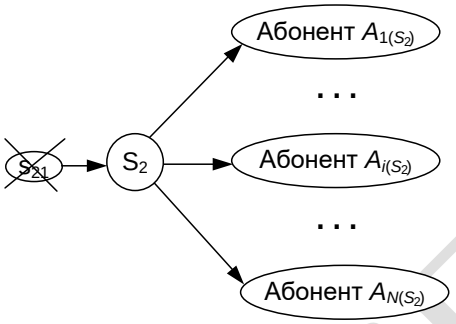


Рис. 6.7. Распространение неисправностей в элементах, поддерживающих работу служб, на качество сервисов для абонентов

Перечень абонентов услуг, например клиентов электронной почты, целесообразно хранить в таблицах, которые уже отсортированы по важности абонентов. Тогда при снижении качества предоставления сервиса вызывается соответствующая таблица и предпринимаются необходимые мероприятия по работе с абонентами.

Таким образом, для оценки влияния неисправностей в сети доступа на качество предоставления услуг абонентам в мультисервисных сетях необходимо одновременно использовать обе схемы, приведенные на рис. 6.5. и 6.6. Действительно, для оценки влияния неисправностей на значения отдельных параметров, характеризующих, например, продуктивность сети, таких как скорость доступа в интернет или реальная полоса пропускания, схемы, приведенной на рис. 6.5, достаточно. Но, если в качестве метрики оценки влияния неисправностей в сети на качество предоставления услуг используется средняя величина или диапазон изменения джиттера при передаче голосовых сообщений, то неисправность в узлах и линиях связи может привести к снижению качества передаваемого голоса или сделать обмен голосовыми сообщениями неосуществимым. Поэтому оценить качество голоса при использовании технологии VoIP с помощью только схемы, приведенной на рис. 6.5, не представляется возможным.

В силу изменчивости структуры и состава ТКС нецелесообразно строить совокупность моделей, которые будут непосредственно связывать все возможные неисправности с изменением значений параметров качества услуг *m*-го пользователя. Более рациональным выглядит подход к решению задачи импакт-анализа на основе структуры, состава ТКС и принципов распространения влияния неисправностей на качество услуг. В основу подхода следует положить модели распространения воздействия неисправного узла на зависимые узлы нижних уровней до тех пор, пока не будет достигнут уровень пользовательского оборудования. После этого определяется степень влияния неисправности на показатели качества для каждого пользователя. В этом случае для каждого типа узлов из U определяются правила распространения влияния неисправностей из H на значе-

ние q_{kl}, $k=\overline{1,M_l}$, $l=\overline{1,K}$, в направлении от узлов, расположенных выше в иерархии сети, к узлам, расположенным ниже. После этого последовательно от узла $U_{i,j}$, в котором произошла неисправность, для каждого из зависимых узлов $U_{i+1,j}$ ниже расположенных уровней рассчитываются коэффициенты ухудшения уровня услуги ρ_{klm}, $k=\overline{1,M_l}$, $l=\overline{1,K}$, в виде (6.24) до тех пор, пока не будет достигнут уровень пользователей, для каждого из которых рассчитывается значение коэффициента ρ_{klm}, $k=\overline{1,M_l}$, $l=\overline{1,K}$, $m=\overline{1,L}$. Если по ходу расчетов выполняется условие

$$\rho_{kl} < \breve{\rho}_{kl}, \tag{6.10}$$

где $\breve{\rho}_{kl}$ – допустимое уменьшение значения *k*-го показателя качества *l*-й услуги, то отмечается затухание влияния неисправности, и дальнейшие расчеты для расположенных ниже уровней не проводятся. При выполнении условия (6.25) ПИА вырабатывает сообщение для реагирования в соответствии с заранее прописанным регламентом.

Для реализации алгебраического подхода задаются множества *U, N,* матрица $Q_m=\|q_{klm}\|$ для всех $m=\overline{1,L}$ и граф *G* в виде, приведенном на рис. 6.5 или 6.6. Для каждого типа узлов из *U* прописываются правила, связывающие неисправность из *H* со степенью влияния на q_{kl}, $k=\overline{1,M_l}$, $l=\overline{1,K}$, на выходе узла.

6.2.5. Импакт-анализ в сетях ОТС при оценке скорости доступа

Рассмотрим применение алгебраического подхода к решению задачи импакт-анализа, когда параметрами качества услуги, влияние на которые будет оцениваться при возникновении неисправностей, являются показатели производительности ТКС; в данном случае, например, скорость доступа к сети интернет.

Для формальной оценки влияния неисправностей на узлы и абонентов сети доступа введем коэффициент снижения скорости передачи $k_{i,j(n)}$, $i=\overline{1,I}$, $j=\overline{1,J_i}$, $n=\overline{1,N_{i,j}}$, где *j* – номер узла в *i*-м уровне иерархии сети доступа, *I* – количество уровней иерархии, выделяемых в рассматриваемой сети доступа, J_i – количество узлов в *i*-м иерархическом уровне, *n* – номер исходящей линии связи из *i,j*-го узла сети, $N_{i,j}$ – количество линий связи, исходящих из *i,j*-го узла сети.

Коэффициент

$$k_{i,j(n)} \in [0,1] \tag{6.11}$$

определяет, во сколько раз уменьшается скорость передачи в *n*-й линии связи, исходящей из *i,j*-го узла сети, относительно нормативного показателя.

Если воздействие всех неисправностей в узлах и линиях связи представить таким образом, чтобы их влияние сводилось к изменению значения коэффициентов $k_{i,j(n)}$, то фактическая скорость доступа $Vp_{i,j(n)}$ в *n*-й линии, исходящей из *i,j*-го узла, будет определяться так:

$$Vp_{i,j(n)} = k_{i,j(n)} \times Vн_{i,j(n)}, \tag{6.12}$$

где $Vн_{i,j(n)}$ – нормативная скорость доступа в *n*-й линии на выходе *i,j*-го узла.

АУТСКИ

С помощью соотношения (6.12) определяется фактическая скорость доступа Vp_m, для каждого m-го абонента A_m, $m=\overline{1,L}$. Если полученная скорость Vp_m больше порогового значения, оговоренного в SLA, т. е. при выполнении условия

$$Vp_m \geq Vr_m, \tag{6.13}$$

где Vr_m – пороговое значение допустимого снижения скорости для m-го абонента, то в отношении m-го абонента никакие меры не предпринимаются. В противном случае действия, предпринимаемые ОТС, определяются степенью снижения скорости доступа и значимостью абонента.

Введем зависящий от состояния узла коэффициент $K_{i,j}$, определяющий снижение относительно нормативного значения параметра производительности j-го узла, $j=\overline{1,J_i}$, i-го, $i=\overline{1,I}$, уровня иерархии узлов сети доступа, J_i – количество узлов U на i-м уровне.

Для рассматриваемого случая коэффициент $K_{i,j}$ определяет степень снижения скорости передачи на выходах узла $U_{i,j}$ относительно нормативного, а значения коэффициента $K_{i,j}$ зависят от состояния узла, обусловленного неисправностями в сети доступа. Естественно применить диапазоном значения этого коэффициента отрезок [0, 1]. Таким образом,

$$K_{i,j} \in [0,1]. \tag{6.14}$$

При этом значение $K_{i,j}$ устанавливается равным 0, если возможные неисправности приводят к тому, что трафик через узел $U_{i,j}$ не проходит. Значение $K_{i,j}$ устанавливается равным 1 при полностью исправном узле $U_{i,j}$, или, если произошедшая неисправность в узле $U_{i,j}$ не влияет на скорость передачи данных. Для всех остальных неисправностей значения $K_{i,j}$ принимают промежуточные значения в отрезке [0, 1].

Коэффициент $K_{i,j}$ можно определять на основании правил, прописываемых для каждого типа узлов в зависимости от структуры сети доступа. Правила должны адекватно учитывать особенности коммуникационных технологий, настройки оборудования, зависимость от величины коэффициентов снижения скорости передачи во входящих линиях, наличие резервирования и другие параметры. При обнаружении неисправностей в сети проводится их локализация, определяется степень серьезности и выполняется сопоставление состояния узла $U_{i,j}$ конкретному значению коэффициента $K_{i,j}$ по соответствующему правилу. Для повышения быстродействия ПИА правила должны быть максимально простыми, однако обеспечивать достаточную для принятия решения точность.

Указанные правила в первую очередь должны учитывать структурно-технологические особенности сети доступа. Исходя из анализа существующих сетей доступа, принципов их функционирования и эксплуатации, используемых в них коммуникационных технологий и оборудования, целесообразно положить в основу определения правил следующие возможные способы соединения узлов сети доступа (см. рис. 6.8).

Способ соединения а. При исправном узле $U_{2,1}$ скорость доступа в исходящей линии определяется входной скоростью и ограничениями, накладываемыми на скорость передачи и приема, устанавливаемую для портов узла $U_{2,1}$ при его конфигурации.

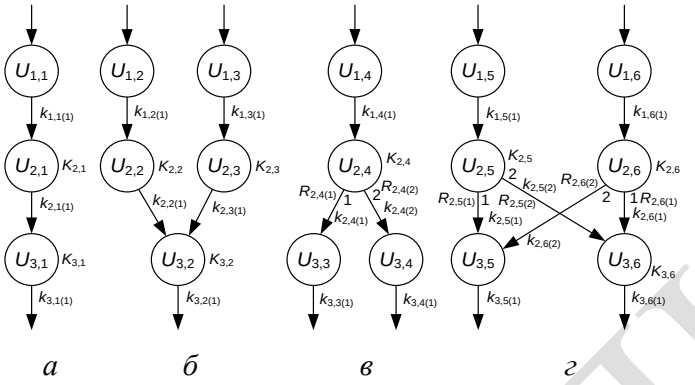


Рис. 6.8. Возможные способы (шаблоны) соединения узлов сети доступа

Если узел $U_{2,1}$ неисправен, то возможны два случая. В первом случае, когда коэффициент $K_{2,1} = 0$, скорость в исходящей ветви равна 0 независимо от скорости на входе. Во втором – $0 < K_{2,1} < 1$ и неисправность в узле $U_{2,1}$ приводит к снижению скорости в исходящей линии в $K_{2,1}$ раз.

Тогда коэффициент снижения скорости относительно номинальной в исходящей линии связи узла $U_{2,1}$ определяется так:

$$k_{2,1(1)} = K_{2,1} \times k_{1,1(1)}. \tag{6.15}$$

Коэффициент $k_{3,1(1)}$ снижения скорости на выходе узла $U_{3,1}$ с учетом (6.15) определяется с помощью выражения

$$k_{3,1(1)} = K_{3,1} \times k_{2,1(1)} = K_{3,1} \times K_{2,1} \times k_{1,1(1)}. \tag{6.16}$$

Способ соединения б. Если в один узел, например в узел $U_{3,2}$, ведут несколько линий связи, которые используются для распределения нагрузки, поступающей от абонентов, а не для целей резервирования, то при $K_{3,2} = 1$ для оценки влияния нужно использовать выражение

$$k_{3,2(1)} = \begin{cases} k_{2,2(1)} + k_{2,3(1)}, \text{ если } k_{2,2(1)} + k_{2,3(1)} \le 1, \\ 1, \text{ если } k_{2,2(1)} + k_{2,3(1)} > 1. \end{cases} \tag{6.17}$$

Если коэффициент $K_{3,2}$ не равен 1, то проводится соответствующая коррекция коэффициента $k_{3,2(1)}$.

Способ соединения в. Введем коэффициент $R_{i,j(n)}$, определяющий распределение полосы пропускания или скорости передачи между n-ми исходящими линиями связи j-го узла i-го иерархического уровня.

Значения коэффициента $R_{i,j(n)}$ принадлежит следующему отрезку:

$$R_{i,\,j(n)} \in [0,1]. \tag{6.18}$$

Если скорость доступа во входящей линии полностью распределяется между портами исходящих линий без дополнительных ограничений, устанавливаемых при конфигурации узла $U_{2,4}$, то принимается, что

$$\sum_{n=1}^{N_{i,j}} R_{i,j(n)} = 1. \qquad (6.19)$$

Коэффициент снижения скорости в 1-й и 2-й линиях связи на выходе узла $U_{2,4}$ при возникновении неисправностей определяется по формуле (6.15).

Способ соединения г. Для повышения надежности в сети доступа используются резервные линии связи. На рис. 6.8,*г* такими линиями являются линии 2, исходящие из узлов $U_{2,5}$ и $U_{2,6}$, в то время как линии 1 являются основными. Переключение на резерв в IP-сетях происходит автоматически при выходе из строя основной линии связи или основного узла, а распределение скоростей в основной и резервной линиях прописывается в узловых элементах при их конфигурации. Тогда для оценки коэффициента $k_{3,6(1)}$ нужно использовать выражение

$$k_{3,6(1)} = \begin{cases} K_{3,6} \times k_{2,6(1)}, \text{ если } K_{2,6} \text{ равен 1,} \\ K_{3,6} \times k_{2,5(2)}, \text{ если } K_{2,6} \text{ равен 0, а } K_{2,5} \text{ равен 1,} \\ 0, \text{ если } K_{2,6}=K_{2,5}=0. \end{cases} \qquad (6.20)$$

Для оценки коэффициента $k_{2,5(2)}$ следует использовать выражение

$$k_{2,5(2)} = \begin{cases} R_{2,5(2)} \times K_{2,5} \times k_{1,5(1)}, \text{ при } K_{2,6} = 0, \\ 0, \text{ если } K_{2,6} \neq 0. \end{cases} \qquad (6.21)$$

Естественно, что вследствие выделения части канальных ресурсов на поддержание резервной линии может снизиться скорость в основной линии. Тогда для оценки коэффициента $k_{2,5(1)}$ нужно использовать выражение

$$k_{2,5(1)} = \begin{cases} R_{2,5(1)} \times K_{2,5} \times k_{1,5(1)}, \text{ при } K_{2,6} = 0, \\ K_{2,5} \times k_{1,5(1)}, \text{ если } K_{2,6} \neq 0. \end{cases} \qquad (6.22)$$

Таким образом, состояние одного из узлов, например $U_{2,6}$, может оказывать влияние на коэффициенты снижения скорости в выходных линиях других узлов, в данном случае $U_{2,5}$, находящихся на том же или на расположенных выше уровнях относительно рассматриваемого узла и не связанных с ним физически.

Такую логическую взаимосвязь состояний можно установить на основе анализа резервных линий и узлов на схемах, подобных приведенным на рис. 6.5 или 6.8. Учитывая низкую динамику создания и удаления резервных узлов и линий связи, информацию о взаимовлиянии можно отдельно фиксировать в качестве свойств или параметров узлов, отказы в которых оказывают косвенное влияние на другие узлы, или задавать в виде соответствующих правил. Для сохранения единства подхода взаимовлияние состояний узлов задается изменением состояния зависимого узла при изменении состояния основного.

На самом нижнем уровне иерархии сети доступа коэффициент $k_{i,j(n)}$ определяет, во сколько раз уменьшилась скорость передачи на входе или выходе абонентского оборудования относительно скорости, заданной в SLA. Точка контроля скорости на входе или выходе абонентского оборудования определяется организационно или ограничивается возможностями ПИА контролировать выход абонентского оборудования.

МГТС КП

Фактическая скорость доступа Vp_m каждого m-го абонента A_m, $m = \overline{1, L}$, определяется с помощью следующего выражения, полученного по аналогии с выражением (6.12):

$$Vp_m = k_m \times Vн_m,, \qquad (6.23)$$

где k_m коэффициент, соответствующий $k_{i,j(n)}$ на выходе абонентского оборудования, $Vн_m$ – нормативная скорость доступа m-го абонента.

Теперь можно принимать решения, о которых говорилось выше. Если скорость Vp_m меньше порогового значения, оговоренного в SLA, т. е. при выполнении условия (6.13), то в отношении m-го абонента никакие меры не предпринимаются по причине отсутствия существенного влияния на уровень услуг. В противном случае действия, предпринимаемые подразделением поддержки клиентов оператора связи, определяются степенью снижения скорости доступа и значимостью абонента.

При использовании нескольких пороговых значений Vr_m можно дифференцировать действия подразделения поддержки клиентов.

Ниже предлагается алгоритм работы ПИА при определении влияния неисправностей в сети доступа на показатели производительности качества обслуживания абонентов.

Алгоритм работы ПИА. Алгоритм импакт-анализа для данного случая разработан в работе [153] и может быть использован для оценки неисправностей в сети доступа на качество предоставления услуг, которые можно оценить с помощью показателей производительности.

Шаг 1. Обнаружение и локализация неисправностей в сети доступа. Информация о них поступает в ПИА из СУИ, CRM-системы или СУ отдельными коммуникационными технологиями. После фильтрации и корреляции сообщений о неисправностях выявляется проблемный узел или линия связи.

Шаг 2. Оценивается влияние неисправности на снижение производительности путем определения значений коэффициентов $K_{i,j}$ и $k_{i,j(n)}$ только для проблемного узла и исходящих линий связи. Если значения снижения производительности не превышают порога реагирования, который свидетельствует о существенности неисправности, и в сети есть только одна неисправность, то дальнейшие шаги импакт-анализ не выполняются.

Шаг 3. Если неисправность оказывается существенной или имеет место несколько независимых неисправностей, то выявляются узлы, расположенные на смежных расположенных ниже иерархических уровнях, на которые влияние неисправностей распространяться непосредственно, и узлы, на которые неисправность в рассматриваемом узле влияет косвенно. Для всех этих узлов рассчитываются коэффициенты $K_{i,j}$ и $k_{i,j(n)}$. Если промежуточные коэффициенты не превышают порог реагирования и ошибок в сети не более одной, то считается, что влияние неисправности затухло и дальнейшие шаги импакт-анализа не выполняются.

Шаг 4. Осуществляется проверка, достигнут ли уровень абонентского оборудования. Если не достигнут, то повторяются шаги 2 и 3.

Шаг 5. Выявляются все абоненты, у которых вследствие неисправности снизилась производительность. Абоненты ранжируются по важности и для каждого из них, начиная с наиболее важных, проводится сравнение значений рассчитан-

ных коэффициентов k_m с пороговыми значениями. Если величина снижения производительности не превышает порог реагирования, то никаких действий не выполняется. Если порог реагирования превышен, то ПИА генерирует соответствующее информационное сообщение.

Данный алгоритм позволяет алгебраически рассчитать влияние неисправностей на значения параметров качества, когда показателем уровня услуги является скорость доступа.

Алгебраический подход отличается большим быстродействием и позволяет рассчитать значение ρ_{klm} от места возникновения неисправности через промежуточные узлы до оборудования пользователей. Однако такой поход можно применять только при простой и фиксированной логике взаимодействия узлов в ТКС. Кроме того, этот подход можно использовать только для оценки показателей качества, для которых можно использовать алгебраические методы. Для решения задач импакт-анализа в общем случае следует использовать рассматриваемый ниже подход на основе правил.

6.3. ОБОБЩЕННЫЙ ПОДХОД К ОЦЕНКЕ ВЛИЯНИЯ НЕИСПРАВНОСТЕЙ В ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ НА УРОВЕНЬ УСЛУГ

6.3.1. Особенности задачи анализа распространения влияния неисправности на качество услуг

Сегодня успешность предприятий существенно зависит от непрерывности и качества информационного взаимодействия, осуществляемого различными коммуникационными системами как внутри предприятия, так и между головным офисом и филиалами, предприятием и сторонними организациями, между сотрудниками предприятия и клиентами и т. п. Коммуникационное взаимодействие обеспечивается операторами мобильной связи, телефонными компаниями, интернет-сервис провайдерами и другими поставщиками коммуникационных услуг – ОТС. При этом ОТС, как правило, предоставляют комплекс коммуникационных услуг.

Поскольку потери бизнеса от несвоевременности получения или низкого качества коммуникационных услуг могут быть очень значительными, предприятия и корпорации перекладывают их на поставщиков этих услуг в виде штрафных санкций, размер которых может быть весьма существенным. Большие объемы и стоимость телекоммуникационных услуг при одновременных жестких требованиях к непрерывности и качеству их предоставления, существенные штрафные санкции и опасность потери клиентов вынуждают ОТС непрерывно отслеживать состояние коммуникационной системы.

В свою очередь, острая конкуренция в сфере телекоммуникационных услуг вынуждает ОТС не только непрерывно поддерживать высокий уровень предоставляемого сервиса, но и прилагать постоянные усилия для повышения качества обслуживания своих абонентов. Параметры качества обслуживания заказчиков или абонентов оговариваются в SLA. SLA фиксирует качественное и количественное описания сервисов с указанием условий, параметров и других характеристик услуг, как с точки зрения клиента, так и с точки зрения ОТС, а также взаимные ответственности. В SLA, в частности, определяются требования к параметрам производительности и поддержки – среднему или максимальному

времени реагирования; доступности – проценту времени обслуживания, в течение которого услуга фактически доступна пользователю; надежности – количеству инцидентов за определенный срок в течение согласованного времени обслуживания; устойчивости – времени восстановления услуги (полного или частичного) в случае повреждения инфраструктуры ОТС. При невыполнении требований к качеству обслуживания, оговоренных в SLA, ОТС несет убытки в результате потери клиентов или штрафных санкций. Поэтому при возникновении неисправностей необходимо не только осуществлять их локализацию и максимально быстрое устранение, но и оперативно определять степень влияния неисправностей в коммуникационной системе на качество предоставления сервиса всем клиентам, в первую очередь наиболее важным.

В настоящее время в мире телекоммуникаций превалирует модель взаимодействия с клиентами CRM, ставящая в центр философии бизнеса пользователя и выдвигающая одним из основных направлений деятельности обслуживание абонентов.

В то же время большое количество пользователей и необходимость снижения затрат на обслуживание ИТ-инфраструктуры вынуждают ИТ-подразделение дифференцированно обслуживать пользователей. Для реализации эффективного дифференцированного обслуживания в случае возникновения неисправностей необходимо обеспечить в ИТ УИТИ поддержание жизненного цикла снижения потерь из-за ухудшения качества предоставляемых услуг, состоящего из следующих этапов: выявление неисправностей; определение их влияния на качество услуг; выделение наиболее важных пользователей, пострадавших в результате получения некачественных услуг; проведение дополнительных мероприятий по отношению к этим пользователям или, безотносительно значимости пользователей, первоочередное восстановление качества предоставления наиболее важных услуг.

Для автоматизации процессов модели CRM ОТС используют множество пакетов специального ПО, из которых некоторые выполняют функции определения распространения влияния неисправностей в сети на качество сервиса абонента. Это направление деятельности ОТС получило название импакт-анализа (Impact Analysis – IA) [333].

Перед подсистемами импакт-анализа (ПИА), являющихся частью СУИ, ставятся две задачи:

1) выявить абонентов или пользователей, пострадавших в результате неисправностей в сети связи и ощутивших влияние неисправностей в ТКС на качество услуг;

2) определить, насколько серьезно они пострадали, т. е. насколько снизились показатели уровня услуг у этих абонентов.

На решение задач импакт-анализа и последующую реакцию CRM системы относительно наиболее важных из пострадавших клиентов отводятся считанные минуты. Оперативное решение задач импакт-анализа осложняется большим объемом инвентаризационных баз данных, содержащих информацию не только об абонентах, но и обо всем коммуникационном оборудовании, линиях связи и других элементах ИТ-инфраструктуры. Причем количество сетевых компонентов в сетях доступа крупных ОТС, содержащих десять и более иерархических уровней, может достигать нескольких миллионов единиц. ПИА должны связать неисправности, возникающие в коммуникационной системе, с соответствующим ухудшением качества предоставления услуг абонентам.

Импакт-анализ сопряжен с рядом трудностей, обусловленных большим количеством элементов коммуникационных систем, сложностью и запутанностью структуры сети, сложными зависимостями влияния неисправностей и совокупности неисправностей на качество предоставляемого сервиса, наличием схем резервирования, большим количеством клиентов, часто исчисляемых миллионами, большим количеством услуг и пр. Для разработки ПИА, позволяющих в минимальные сроки выявлять клиентов, ощущающих снижение качества обслуживания вследствие неисправностей в коммуникационной системе, и оценить степень ухудшения качества получаемых телекоммуникационных услуг, необходимо разработать эффективные математические модели и алгоритмы определения степени влияния неисправностей в сетях ОТС на качество предоставления телекоммуникационных сервисов. Эти и другие проблемы превращают импакт-анализ в трудоемкую задачу, решение которой требует больших вычислительных ресурсов. При этом необходимо определить принципы построения ПИА, нацеленной на максимально быстрое решение задач импакт-анализа.

Для разработки ПИА необходимо:

- разработать эффективные математические модели представления ИТ-инфраструктуры,
- разработать алгоритмы определения степени влияния неисправностей в ТКС на качество услуг;
- определить принципы построения ПИА, нацеленной на максимально быстрый импакт-анализ.

Создать универсальную модель или метод для оценки влияния неисправностей на все услуги не представляется возможным. Поэтому следует разработать совокупность моделей, методов и алгоритмов импакт-анализа, учитывающих особенности услуг и технологий их предоставления, которые способны составить основу программного инструментария импакт-анализа в рамках имеющихся временных, технологических и других ограничений.

Обзор существующих решений. Анализу влияния неисправностей в сетях крупных ОТС посвящено множество работ. В работах [88, 91, 187, 192, 252, 333, 340, 342] рассматриваются различные аспекты решения этой актуальной проблемы. В большинстве анализируется влияние отказов в предоставлении сервисов на бизнес-операции компаний, которые предоставляют сервисы или используют ИТ для поддержки своих бизнес-процессов [187, 192]. Авторы этих работ в основном концентрируются на моделях и методах эффективного управления инцидентами и процессами восстановления сервисов. На их основе разрабатываются системы, целью которых является минимизация влияния на бизнес срывов важнейших сервисов. Базовая концепция построения этих систем – интеграция управления выполнением бизнес-процессов с управлением системой в целом и обеспечение их соответствующей ИТ-инфраструктурой.

Так, в работе [123] рассматриваются вопросы управления устранением неисправностей в ИУС без учета влияния неисправностей в сети на качество предоставления сервисов. Анализу качества функционирования элементов ИУС посвящена работа [104], но в ней не предлагаются модели и методы анализа распространения изменения состояния элементов телекоммуникационной сети на параметры функционирования оборудования в точках доступа абонентов.

В работе [88] предложен метод и описана система определения влияния неисправностей в телекоммуникационных сетях на работоспособность сервисов.

На основе анализа собранных данных, характеризующих работоспособность сервисов; предложенная система позволяет предсказывать их последующее состояние. Объединяя фактические и потенциальные неисправности в сети, система снабжает подразделение технической поддержки ОТС информацией об отказах в оборудовании. Система не учитывает важность абонентов, а технология мобильной связи ограничивает возможности службы технической поддержки в выборе методов индивидуальной работы с наиболее важными абонентами.

В работе [91] предложена сервисно-ориентированная модель, рассматривающая различные фазы предоставления сервиса и позволяющая определить, каким образом неисправности, возникающие в отдельных сетевых элементах, влияют на работоспособность сервисов в целом. Для этого вводится понятие индикаторов состояния компонентов сети применительно к отдельным сервисам, а умножение значений индикаторов на соответствующий вес позволяет получить приоритетный список неисправностей, оказывающих наибольшее влияние на качество предоставления сервисов. При этом не предлагаются методы оценки влияния неисправностей на качество предоставления сервисов отдельным абонентам.

Если для транспортных сетей проблемы оценки качества сервисов, которые предоставляют операторы своим клиентам, является предметом многих исследований, например [88, 91, 333], то для сетей доступа проблеме оценки качества обслуживания операторами своих клиентов посвящено сравнительно немного исследований. Причем эти исследования преимущественно фокусируются на вопросах определения влияния отклонения значений показателей производительности сети на качество сервисов, предоставляемых операторами мобильной связи. Отсутствуют работы, в которых бы предлагались целостный подход или концепция решения этой комплексной проблемы.

Некоторые работы посвящены разработке моделей, которые определяют зависимости сервисов от других сервисов и ресурсов, а также параметров качества сервисов в SLA клиентов и текущего уровня использования сервисов. На основе этих моделей разрабатываются процедуры принятия решений для эффективного восстановления сервисов [252]. Значительная часть работ посвящена моделям анализа влияния проблем в определенных типах сетей, например IP-телефонии [342].

Еще одним направлением исследований является разработка моделей влияния на бизнес операционных рисков, которые являются результатом изменений, связанных со срывами в предоставлении сервисов на неопределенное время [340].

В работе [161] предложен подход к быстрому решению данной проблемы на основе комбинирования продукционного и алгоритмического подходов. Там же приводится алгоритм получения нужных результатов в рамках временных требований с соответствующим уровнем точности. Алгоритм реализован так, что представление сети и необходимых для расчетов данных и собственно обработки этих данных осуществляется в ОЗУ.

Если для управления выполнением предложено много моделей общего вида, например [148, 160], которые нашли воплощение в реальных СУИ [120, 154], то информация об общих моделях оценки влияния неполадок в сетях ОТС на качество обслуживания клиентов в литературных источниках отсутствует. В статье [153] предложено решение для задачи определения влияния неполадок в сети до-

ступа на качество сервисов, предоставляемых ОТС клиентам. Это решение базируется на аналитическом расчете влияния неисправностей в узлах сети на снижение показателей продуктивности для каждого абонента.

Однако целостный подход к решению этой проблемы отсутствует. Поэтому возникает потребность в общим подходе к решению задачи импакт-анализа, который позволит определять структурно-функциональную схему взаимодействия различных моделей. Такой подход, построенный на определенных правилах распространения влияния неисправностей, учитывающих особенности структуры сети и услуг, используемые технологии и другие важные факторы, должен помочь разработчикам ПИА определить последовательность действий, построить механизм подключения моделей, уточнить характеристики, влияющие на распространение неблагоприятных последствий неисправностей на качество предоставления услуг пользователям.

Разработчикам ИТ, применяемым в ПИА, важно знать различные потенциально возможные подходы к решению проблемы импакт-анализа и иметь их сравнительные характеристики, прежде всего требуемые время и память. С одной стороны, это важно для принятия решения о целесообразности выполнения самого проекта. С другой, когда решение уже принято, эта информация позволит определиться с архитектурой программной реализации, набором компонентов и основными проектными решениями по каждому из компонентов. Для этого нужно обобщить предложенные идеи, разработать средства формального описания структур сетей доступа и схем предоставления сервисов, представления и обработки продукций, разработать более эффективные алгоритмы для реализации подхода и выполнить глубокие экспериментальные исследования.

Поэтому указанные положения следует развить и довести до уровня реализации системы оценки качества сервисов на основе использования правил. Для реализации такой системы необходимо разработать алгоритм расчета параметров воздействия и экспериментально проверить его работоспособность и возможность получения нужных результатов в приемлемый для реальных сетей доступа время.

Ниже предлагается подход к быстрому решению проблемы импакт-анализа на основе комбинирования продукционного и алгоритмического подходов, приводится алгоритм получения нужных результатов в рамках временных требований с соответствующим уровнем точности.

6.3.2. Продукционный и продукционно-алгоритмический подходы к решению задач импакт-анализа

Неисправности в устройствах или сервисах сети доступа могут снижать показатели качества обслуживания клиентов оператора, угрожая экономическими потерями. Для их предупреждения менеджеры и технические службы оператора должны принимать взвешенные и своевременные решения по устранению влияния неисправностей [347, 348, 349]. Но для этого службам оператора необходимо контролировать показатели уменьшения уровня обслуживания для каждого клиента, а реальные значения показателей должны определяться достаточно ограниченное время после возникновения событий, которые привели к изменению их значений. Расчет этих показателей требует учета многих параметров, причем логика осуществления расчетов должна быть гибкой, поскольку сетям доступа и

политике оператора по построению отношений с клиентами присуща изменчивость.

Поэтому перед разработчиками СУ для ОТС встала сложная научно-практическая проблема разработки общего подхода к решению задачи импакт-анализа в реальных условиях функционирования сетей доступа. В основу подхода положена модель сети, которая включает только необходимые для оценки уровня услуг структурные и функциональные особенности сети и эффективный способ импакт-анализа.

Эффективность импакт-анализа обеспечивается следующими приемами ускорения расчетов:

1) использование структурного представления сети доступа, ее состояния и правил выполнения расчетов;

2) выделение сегмента сети, на который влияет неисправность, с последующим решением задачи импакт-анализа только в этом сегменте;

3) ранжирование пользователей по значению коэффициента важности w_m, $m=\overline{1,L}$, на классы с последующим импакт-анализом для классов;

4) выполнение импакт-анализа в порядке убывания важности w_m, $m=\overline{1,L}$;

5) использование предварительно подготовленных вариантов базовых расчетов, комбинируя которые можно быстро охватить все классы, на которые оказали влияние неполадки;

6) использование такого представления сети, которое позволяло бы хранить все необходимые для проведения расчетов данные в ОЗУ;

7) использование такого представления сети, которое позволяло бы параллельное выполнение расчетов для определения влияния неполадок.

Для каждого m-го пользователя, $m=\overline{1,L}$, где L – количество пользователей, существует матрица $Q_m=\|q_{klm}\|$, элемент q_{klm}, $k=\overline{1,M_l}$, $l=\overline{1,K}$, которой соответствует целевому значению k-го показателя качества l-й услуги, определенному для m-го пользователя. Каждому пользователю A_m, $m=\overline{1,L}$, присваивается коэффициент важности w_m, принимающий значения из отрезка [0, 1].

Для каждого k-го показателя качества l-й услуги m-го пользователя вычисляется коэффициент ухудшения уровня услуги с помощью выражения

$$\rho_{klm}=\frac{q_{klm}-q^{*}_{klm}}{q_{klm}}, \tag{6.24}$$

где q^{*}_{klm}, $k=\overline{1,M_l}$, $l=\overline{1,K}$, $m=\overline{1,L}$, – фактические значения k-го показателя качества l-й услуги.

Для каждого m-го пользователя, $m=\overline{1,L}$, задается пороговое значение $\tilde{\rho}_{klm}$ допустимого снижения k-го показателя качества l-й услуги, которое сравнивается со значением ρ_{klm}. При выполнении условия

$$\rho_{klm}>\tilde{\rho}_{klm}, \tag{6.25}$$

относительно пользователей с заданными значениями w_m предпринимаются дополнительные мероприятия по обслуживанию (информирование, переключение

на резерв, предоставление альтернативных источников услуг и др.), в то время как для остальных пользователей выполняется номинальное обслуживание.

Таким образом, задача импакт-анализа сводится к тому, чтобы при возникновении неисправности в ИТ-инфраструктуре определить значения коэффициента (6.24), при $\rho_{klm} > 0$ провести проверку условия (6.25), и если условие выполняется, то передать в СУ ТКС значения m, w_m и ρ_{klm}.

Постановка проблемы исследования. Пусть известна информация об элементах ТКС и их взаимосвязях. Пусть заданы значения T_c – временной интервал между моментами поступления информации об изменении составляющих ТКС или их взаимосвязи, T_f – временной интервал между поступлением сообщений о неисправностях или запросами пользователей о требовании определения уровня их обслуживания и реакцией ПИА.

Для определения показателей уровня обслуживания клиентов необходимо разработать формальное представление сети на основе понятий теории графов, а также эффективные алгоритмы его построения и модификации при условии предоставления данных об изменениях элементов или связей за время, не превышающее

$$\min\{T_c, T_f\}. \qquad (6.26)$$

Кроме того, на базе представления структуры сети требуется разработать:

- метод расчета показателей уровня обслуживания клиентов при изменении значений показателей функционирования узлов;
- соответствующие алгоритмы расчета показателей уровня обслуживания клиентов в соответствии с заданными изменениями в показателях функционирования узлов;
- обосновать оценки алгоритмов;
- определить время работы программ, реализующих алгоритм расчета показателей уровня обслуживания клиентов, который не должен превышать ограничения T_f.

Суть подхода. Учитывая наличие резервных линий, изменяющиеся политики управления ТКС, динамику сетей, оборудования и технологий, импакт-анализ целесообразно базировать не на основе алгоритмического подхода, а использовать построенные на правилах продукционные системы, когда правила (продукции) формируют базу знаний, а механизм вывода определяет порядок применения правил. Именно по такой схеме работает продукционный подход. Он позволяет: отделить алгоритм расчета коэффициентов узла от правил, по которым взаимодействуют узлы в сети; описать логику взаимодействия узлов в сети в виде продукционных правил.

Продукционные системы построены на символьных преобразованиях. Реализация продукционного подхода в программном коде требует выполнения большого количества строковых сравнений и преобразований, что существенно увеличивает время импакт-анализа и не может обеспечить его выполнение в условиях жестких временных ограничений. Ситуацию усугубляет тот факт, что обработка большого количества правил также требует значительного времени. Поэтому в чистом виде продукционный подход не применяется.

Такого недостатка нет в предлагаемом продукционно-алгоритмическом подходе, объединяющем преимущества продукционного подхода с алгоритмическим, обладающим высоким быстродействием за счет выполнения алгебраических операций над коэффициентами. Продукции используются для построения модели сети с учетом правил влияния, а алгоритмический метод – для определения степени влияния.

После применения продукционных правил выполняются расчеты по модели с использованием алгоритмического подхода. Они сводятся к расчету коэффициентов узлов в сети. Рассчитываются коэффициенты всех узлов модели, а при возникновении неисправностей перерасчет выполняется только для узлов, на которые повлияла неисправность.

Итак, изменчивость сетей, их оборудования и технологий, возможное изменение политики компании относительно базовых принципов оценки воздействия обусловили применение для импакт-анализа алгоритмического и продукционного подходов. Вклад продукционного подхода – не только отделение данных (правил и рабочей области) от процедур (механизма вывода), но и устранение потребности в перепрограммировании СУИ при изменении отдельных правил. Однако устранение недостатков символьных преобразований, применение которых не может обеспечить выполнение расчетов в условиях жестких временных ограничений, требует применения дополнительных структур данных для ускорения поиска как в рабочей области, так и в системе продукций. Это вызывает необходимость дополнительных исследований. Здесь перспективными представляются следующие приемы повышения эффективности работы систем, основанных на представлении и обработке знаний с помощью продукций: индексация элементов рабочей области; структурирование системы правил; сохранение правил в элементах представления сети доступа, соответствующих узлам, для анализа влияния которых на другие элементы созданы эти правила.

Вклад алгоритмического подхода связан с использованием упрощенного представления сети, которое содержит результаты обработки правил продукционной системы в виде, готовом для прямых расчетов. Иными словами, на основе системы продукций строится алгоритм расчета влияния неполадок. При этом правила отрабатываются заранее – проверяются условные части продукций, решаются возможные конфликты и на соответствующем пути в графе сохраняется коэффициент, необходимый для расчета показателя влияния неполадок. Тогда собственно определение влияния неисправностей сводится к пересмотру графа от вершин верхних ярусов к листьям, начиная с вершин с неисправностями, и выполнению последовательности арифметических операций над хранимыми коэффициентами влияния с учетом параметров неполадки.

6.3.3. Модели сети для оценки влияния неисправностей

Представление сети. Представление сети определяет структуру сети и схему предоставления услуг с учетом состава сети, связей между элементами, элементами сети и услугами, особенностей сети, услуг и технологий с точки зрения импакт-анализа.

Главное требование к представлению – достаточность информации относительно состава, связей между элементами сети, элементами сети и сервисами, особенностей сети, сервисов и технологий реализации, что позволить оценить

последствия неполадок в структурных элементах и сервисах для определения качества обслуживания клиентов.

Многоплановость назначения обуславливает разноплановость требований к соответствующим формальным средствам. В наиболее абстрактном виде эти требования можно разбить на две категории: требования к возможностям представления сети с точки зрения реализации функций управления сетью и требования к временным параметрам и параметрам программно-аппаратной реализации, которым может удовлетворить представление. Учитывая такое деление, целесообразно представление сети разбить на два основных уровня:

1) первый уровень – полное представление сети;

2) второй уровень – упрощенное представление сети.

Первый уровень составляет полное представление о составе и структуре сети, пользователях, сервисах, оборудовании и технологиях, характеристиках элементов сети. Это ядро подсистемы инвентаризации СУИ, которое интегрирует представления о сети и составляет центральный компонент СУ оператора. Полное представление используется для визуализации структуры сети, анализа и прогнозирования состояния сети, ее подсетей и элементов, окончательных расчетов и реализации других функций СУ.

Формируя общее представление о структуре, сетевых сервисах, состоянии элементов и компонентов, технологиях и ограничениях на функционирование полное представление составляет очень большой массив информации. Эта информация хранится в большой БД под управлением промышленной СУБД, и структурные связи реализуются за счет адресных ссылок (индексов). Что касается анализа проблем, связанных с нештатными режимами функционирования сетей и оценкой влияния неполадок на качество обслуживания пользователей и последствий невыполнения SLA, такое представление не дает оснований надеяться на построение быстрых алгоритмов и не позволяет провести импакт-анализ за время, меньшее T_f.

Поэтому для импакт-анализа необходимо использовать следующий уровень представления сети – упрощенное представление (модель) сети в виде дерева распространения влияния, которое формирует второй уровень представления. Упрощенное представление используется для быстрого импакт-анализа и должно быть таким, чтобы его можно было полностью записать в ОЗУ при запуске ПИА и все расчеты проводить на основе данных, загруженных в ОЗУ.

Сети доступа имеют преимущественно иерархическое строение, количество уровней которой зависит от количества клиентов и их расположения, причем количество уровней сетей доступа некоторых операторов может достигать 10. Для выполнения требований к надежности обслуживания клиентов им обеспечивается доступ к ресурсам сети несколькими путями.

Упрощенное представление, которое называется «ярусно-параллельная форма распространения влияния неполадок», сокращенно – «дерево распространения», состоит из двух частей:

1) описание структуры сети системой элементарных иерархических подграфов, структуры сервисов и служб;

2) описание особенностей импакт-анализа: задаются элементы сети, влияющие на качество функционирования других элементов, определяются направление и объем необходимых вычислений.

По своей сути эти два описания должны характеризовать сеть доступа с двух взаимодополняющих точек зрения.

В первом случае общая структура сети представляется как ярусно-параллельная форма специального вида – система элементарных иерархических подграфов, объединение которых позволит получить ориентированный граф структуры сети, который в общем случае не является корневым деревом (рис. 6.9). На самом верхнем ярусе сосредоточены ресурсы, а на нижнем, т. е. в листьях – клиенты. Каждая вершина может иметь n родительских вершин из разных ярусов и g дочерних вершин из разных ярусов. Но принципиально важным является тот факт, что этот граф можно изобразить в виде объединения двух и более корневых деревьев, которые могут пересекаться [161]. Пересечение этих графов определяет те устройства и линии связи, которые используются для предоставления основных и резервных путей доступа клиентам. Листья соответствуют пользовательскому оборудованию.

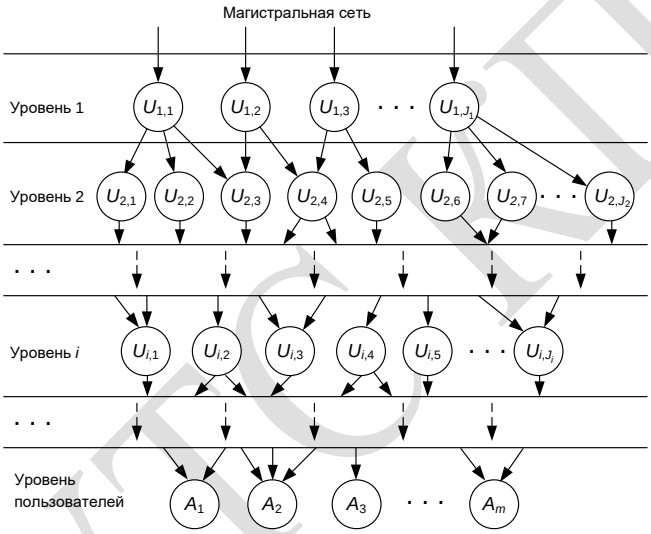


Рис. 6.9. Граф распространения влияния неисправностей

Такое представление позволит применить эффективные в условиях изменения структуры формальные средства и структуры данных. Так, для представления поддеревьев можно использовать разбитую на зоны левую польскую бесскобочную нотацию, а для анализа структуры – алгебраический подход. Также для эффективной обработки данных в ОЗУ можно использовать двоичное представлено структурных, технологических и других особенностей сетей.

С другой стороны, сеть представляется как система точек влияния, соответствующих им сегментов воздействия и коэффициентов расчета влияния. Такое представление позволит задать, во-первых, точки воздействия – элементы сети, в которых сеть разветвляется и которые определенным образом влияют на качество функционирования нескольких следующих элементов, сегментов или сервисов сети. Во-вторых, такое представление позволяет определить направление и объем необходимых вычислений. Для каждой точки воздействия указываются компоненты, сегменты, элементарные подграфы, сервисы и технологии непосредственного воздействия, а также коэффициенты для выполнения непосредственных вычислений.

В каждом узловом коммуникационном устройстве каждая линия связи задействует определенный порт. Таким образом, эти устройства структурированы и неполадки могут касаться всего устройства в целом или определенной подструктуры. При этом речь может идти о полной потере функции устройства (подструктуры), или потере части функций устройства (подструктуры).

Поэтому оператор для каждого устройства вводит специальный параметр оценки уровня соответствия нормам функционирования – severity level, который принимает значения, например, в диапазоне от 0 до 100 или является нормированной величиной из отрезка [0, 1]. Значение каждой позиции этого параметра четко определено и ОТС знает последствия для клиента каждого значения из указанного диапазона.

На основе значений этого показателя для устройств сети, структуры сети, параметров технологий, сервисов и оборудования необходимо рассчитать значение показателя качества обслуживания для каждого клиента. Это можно сделать на основе набора правил, который, однако, достаточно громоздок. Сеть доступа оператора может насчитывать миллионы клиентов, которые могут принадлежать к разным классам важности; для оператора, например, это как минимум клиенты классов platinum, gold, silver, bronze. При этом для расчета показателей для всех клиентов отводится ограниченное время, у некоторых операторов оно составляет не более 15 минут на 1,5 млн клиентов.

Размеры сетей, плотный поток изменений в сети и довольно ограниченное время, которое отводится на расчет, превращает эту рутинную, на первый взгляд, процедуру в сложную научно-практическую проблему разработки моделей и методов представления структуры и определение сети и создания на этой основе информационной технологии оценки влияния неполадок в сетях доступа операторов связи, интегрированную в СУ сетями доступа ОТС на уровне OSS.

Необходимо иметь соответствующее ПО, которое, во-первых, будет отвечать за ведение необходимых БД и выполнение расчетов в ограниченное время, а, во-вторых, будет интегрированным в системы OSS для управления сетями доступа операторов ОТС.

Информация о сети находится в неструктурированном в терминах графов виде. Задаются только узловые устройства и устройства следующего уровня, присоединенные к их соответствующим портам, и информация об устройствах, портов и связей.

Когда в системе случаются неполадки, информация о них поступает в виде новых значений параметров оценки уровня их соответствия нормам функционирования. Тогда и необходимо учитывать новые значения показателей обслуживания клиентов с учетом приведенных выше временных ограничений.

С учетом существенных особенностей логики определения влияния неполадок в сетях доступа, наличия значительного количества факторов влияния, существенной размерности сетей, большого количества клиентов и жестких временных ограничений возникает сложная практическая проблема представления информации о структуре сети и правил пересчета показателей и создания программного инструментария оценки влияния неполадок в сетях доступа ОТС на качество обслуживания клиентов.

На эту проблему накладываются дополнительные ограничения, связанные с тем, что периодически q раз в сутки или апериодически поступает информация от других компонентов систем уровня OSS управления сетями доступа ОТС об изменениях в структуре сети.

Поэтому актуальной становится научная проблема разработки моделей и соответствующих методов и алгоритмов пересчета, способных составить основу программного инструментария, который позволит решить приведенную практическую проблему в рамках указанных временных, технологических и других ограничений.

Очевидно, что эффективно решить поставленную задачу можно только при наличии продуманного представления сети доступа и соответствующего способа определения степени влияния неисправностей на качество обслуживания клиентов оператора.

Организация выполнения расчетов. Обобщенная структура функционирования ПИА приведена на рис. 6.10. Эта схема позволит четко определить роль временных ограничений на выполнение расчетов. Модуль формирования модели сети применяется однократно для создания упрощенного представления из полного. Модуль изменения модели применяется каждый раз, когда в сети происходят изменения. Выполнение этих модулей должно осуществляться за время, соответствующее (6.26). Модуль импакт-анализа предназначен непосредственно для оценки влияния неисправностей и активируется каждый раз, когда в систему поступает сообщение о неполадках или запрос от пользователя, который требует определения уровня его обслуживания. Время работы модуля не должно превышать T_f. Поскольку объемы информации очень большие, а интервалы T_c и T_f достаточно малы, то требования к временным параметрам работы модулей системы очень жесткие.

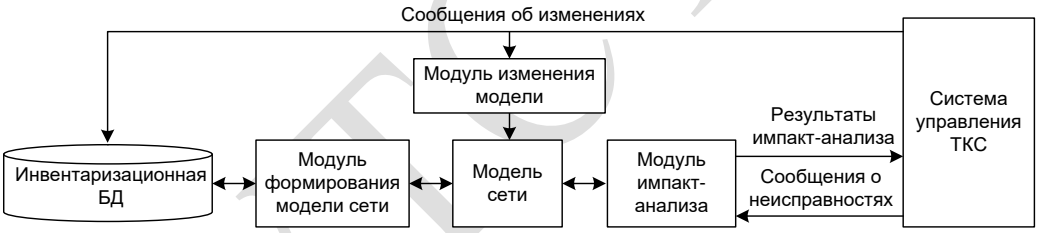


Рис. 6.10. Обобщенная структура функционирования ПИА

Реализация предлагаемого подхода требует разработки моделей и методов оценки влияния неисправностей на уровень обслуживания пользователей. Модели необходимы для формального описания ТКС и учета особенности импакт-анализа. Основное требование к этим моделям: с помощью небольшого базового множества конструктивных элементов и способов их комбинирования однозначно определить многочисленное, а в общем виде содержащее бесконечное количество элементов, отображение из декартового произведения множества элементов ТКС $U = \{U_n, n = \overline{1,R}\}$, где R – количество элементов ТКС, и неисправностей $H = \{H_l, l = \overline{1,N}\}$, где N – количество неисправностей в них, во множество оценок $\rho = \{\rho_{kl}, k = \overline{1,M_l}, l = \overline{1,K}\}$ для импакт-анализа:

$$Q_1 = U \times H \to \rho. \qquad (6.27)$$

Назовем такое отображение базовым отображением импакт-анализа. Методы импакт-анализа должны обеспечить определение отображения Q_1 с учетом важных для оператора критериев и ограничений, особенно на время и необходимую память.

Неисправность может привести к частичной (когда $\rho_{klm} \neq 1$) или полной неработоспособности (при $\rho_{klm} = 1$) услуги s_l, $l = \overline{1,K}$. Причем при $\rho_{klm} = 1$ импакт-анализ существенно упрощается. Действительно, при отказе элемента, приведшего к полной неработоспособности услуги s_l, $l = \overline{1,K}$, из таблицы пользователей l-й услуги считывается перечень пользователей это услугой, после чего проводится их сортировка по значению коэффициента w_m, $m = \overline{1,L}$, что и завершает импакт-анализ.

При $\rho_{klm} \neq 1$ импакт-анализ усложняется. Поэтому дальнейший материал посвящен решению задачи импакт-анализа в таких случаях.

Необходимость интеграции двух подходов к импакт-анализу – алгебраического и продукционного в единый продукционно-алгоритмический обоснована достаточно убедительно [150, 153, 161, 333]. Остается только определить сферы рационального применения каждого из них, выявить особенности применения, проанализировав свойства, преимущества и недостатки.

Определимся с формальным описанием представления сети и системы продукций. Затем покажем, как с их помощью определяется базовое отображение. После этого определимся с возможными критериями и ограничениями на вычисление отображения, что позволит сформулировать требования к методам импакт-анализа.

Модель сети. Представление сети будет состоять из полного и упрощенного представления. Полное представление сети доступа хранится в таблицах больших баз данных под управлением промышленных СУБД. Таблицы составляют *n*-арные нормализованные отношения.

Вся информация о составе, связи и характеристики элементов сети находится в таблицах инвентаризационной БД в неструктурированном в терминах графов виде и используется для полного представления сети при сетевом управлении. Считаем, что имеются такие таблицы с указанием в скобках необходимых для работы алгоритмов полей, используемых импакт-анализом:

- *Objects* (идентификатор объекта, имя объекта, тип объекта);
- *Reference* (идентификатор объекта; ссылки на связанный с ним объект; идентификатор атрибута);
- *Parameters* (идентификатор объекта; идентификатор атрибута, значение атрибута).

Поскольку данные в ОЗУ обрабатываются значительно быстрее, целесообразно разработать формальное представление сети в терминах теории графов, а объемы модели должны позволять хранить необходимые для расчетов данные в ОЗУ вычислительных систем. Для упрощенного представления сети доступа в виде дерева распространения будем использовать граф специального вида с помеченными вершинами и ребрами:

$$G = \langle U, E, f_1, f_2, f_3, f_4, f_5 \rangle, \tag{6.28}$$

где U – множество вершин, соответствующих узлам сети; E – множество ребер, соответствующих линиям связи; $f_1 : E \to U \times U$ – функция, которая каждому ребру ставит в соответствие упорядоченную пару вершин; $f_2 : U \to \{0,1\}$ – функция, которая каждой вершине ставит в соответствие признак необходимости пе-

ерасчета; $f_3 : U \to \rho_{kl}$ – функция, определяющая для каждой вершины соответствие уровню услуг; $f_4 : E \to \rho$ – функция, которая каждому ребру ставит в соответствие признак распространения влияния неисправности в вершине, из которого ребро выходит; $f_5 : E \to T$ – функция, которая каждому ребру ставит в соответствие тип ребра (основная или резервная линия связи).

Множество вершин графа делится на классы, причем каждая вершина может принадлежать одному и только одному классу: U_c – клиенты; U_s – сервисы; U_d – оборудование; U_t – технологии; U_p – порты оборудования и U_r – ресурсы оператора. Множество ребер графа также делится на два подмножества $E = E_m \cup E_r$, где E_m – множество основных связей, E_r – множество резервных связей, причем $E_m \cap E_r = \varnothing$.

Отметим, что функции f_1 и f_5 на множествах вершин и ребер с соответствующими разделениями описывают структуру сети доступа и структуру сервисов и служб, которые ее поддерживают, а функции f_2–f_4 описывают особенности учета влияния неисправностей в сети на уровень качества обслуживания клиентов. Именно указанные наборы функций характеризуют сеть доступа как ярусно-параллельную форму и структуру, которая определяет порядок и параметры расчетов.

Указанная структура организации расчетов строится на основе системы продукций. Концептуально подход основан на том, что правила заранее привязываются к структуре сети и на их основе создается алгоритм, на базе которого осуществляется процесс вычислений уровня качества обслуживания клиентов. Символьная обработка правил осуществляется только модулями формирования и изменения упрощенного представления ПИА. Временные ограничения задаются преимущественно T_c, и только при необходимости более жестких ограничений, связанных со временем T_f, вместо символьной обработки правил применяются алгоритмы обхода деревьев в совокупности с необходимыми арифметическими расчетами.

Базовое отображение анализа влияния неисправностей. Отображение из декартова произведения множеств сетей доступа и неисправностей в них на множество оценок влияния неполадок на уровень обслуживания клиентов задается неявно с помощью возможностей структурирования сети доступа, заложенных в упрощенном представлении и правил продукционной системы. Пусть S – множество возможных структур сети доступа, F – множество возможных комбинаций неисправностей в сети доступа, $\Omega = \{\omega_1,\ldots,\omega_t,\ldots,\omega_T\}$, где $T = |V_c|$, – вектор оценок уровня обслуживания клиентов. Тогда система правил $P = \{P_1,\ldots,P_q,\ldots,P_Q\}$ задает базовое отображение анализа влияния неполадок, определяя для каждой возможной структуры сети доступа и комбинации неполадок значения оценок уровня обслуживания для каждого клиента. Элементами множеств S и F являются текущие структура сети доступа и комбинация неисправностей соответственно. В то же время, система продукций P изменяется достаточно редко.

Таким образом, импакт-анализ при продукционно-алгоритмическом подходе реализуется двумя этапами. Первый этап заключается в отработке продукций и завершается построением модели. Для этого проводится однократный запуск модуля формирования модели. Модуль изменения модели запускается при каждом сообщении об изменениях в сети. На первом этапе импакт-анализа механизм вывода на множестве продукционных правил используется для предварительного отбора правил для структурных элементов сети и описания с помощью функций

f_2–f_4 особенностей импакт-анализа. Второй этап заключается в алгоритмических расчетах по модели при поступлении сообщения о неисправности и завершается определением значений ρ_{klm}, $m = \overline{1, L}$.

Чтобы продемонстрировать возможности неявного определения базового отражения анализа влияния проблем, используем анализ особенностей расчета оценок влияния неполадок на клиентов оператора с помощью продукций, выполненный в работах [161, 333], где определены основные возможные способы соединения узлов сети доступа и особенности соответствующих видов продукционных правил.

Например, иерархическому способу соединения узлов соответствуют правила прямого (6.31) и обратного (6.32) распространения влияния неполадок.

Несколько отличаются правила для других способов соединения узлов. Например, если к узлу идет резервная линия связи и для нее выделен запас производительности, то можно использовать правило (6.38).

Таким образом, решение конфликтных ситуаций касается только первой фазы. Необходимые средства предупреждения конфликтных ситуаций будут рассмотрены ниже с учетом размеров сетей доступа. Что касается особенностей реализации второй фазы, здесь возникает сложная проблема определения условий, при которых символьная обработка системы продукций сводится к пересчету распространения влияния неполадок на основе направленного без циклов обхода графов специального вида.

6.3.4. Особенности реализации подхода к решению задачи импакт-анализа

Реализация подхода построена на идеях эффективной интеграции алгоритмического и продукционного подходов. Правила (продукции), формирующие базу знаний, позволяют механизму вывода определять порядок их применения, используя рабочую область, в которой накапливаются данные о текущем состоянии системы.

Упрощенное представление сети содержит результаты обработки правил продукционной системы в виде, готовом для непосредственных расчетов. Их получают предварительной обработкой системы продукций, решением возможных конфликтов и построением графа, на соответствующих путях которого хранятся необходимые для расчета влияния значения коэффициентов распространения влияния неполадок. Тогда определение влияния сводится к просмотру графа от вершин верхних ярусов к листьям, начиная с вершин с неполадками, и выполнении последовательности арифметических операций над коэффициентами распространения.

Итак, исходя из разноплановости требований, представление сети разделено на полное и упрощенное представления. Первое из них традиционно отвечает за состав и структуру сети, пользователей, сервисы, оборудование и технологии, характеристики элементов сети, составляя ядро подсистемы инвентаризации – центрального компонента системы управления оператора.

Упрощенное представление сети требуется для быстрого решения задачи импакт-анализа. Объем упрощенного представления должен позволять полностью загружать его в ОЗУ для последующего выполнения в этом ОЗУ всех необходимых расчетов. Как указано выше, упрощенное представление в виде дерева распространения влияния неисправностей состоит из двух частей:

1) описание структуры сети доступа, структуры ИТ-сервисов и поддерживающих служб: сеть описывается системой элементарных иерархических подграфов, что позволяет применить эффективные в условиях изменения структуры формальные средства и структуры данных;

2) описание особенностей задачи импакт-анализа: задаются элементы сети, влияющие на качество функционирования других элементов, сегментов или сервисов и определяются направление и объем необходимых вычислений (предыдущим заданием коэффициентов распространения влияния неполадок для выполнение непосредственных вычислений в случае возникновения неполадок).

Таким образом, выполнение расчетов осуществляется в две фазы.

Подготовительная фаза предназначена для формирования необходимых для выполнения расчетов данных. Она осуществляется в ПИА модулем формирования модели сети, который применяется однократно для формирования указанного представления, исходя из полного представления. Специальный модуль изменения модели применяется каждый раз, когда в сети происходят изменения. Поскольку организация расчетов осуществляется на основе системы продукций, на этой фазе правила заранее привязываются к структуре сети в упрощенном представлении.

Правила в нотации Бекуса – Наура [304] имеют следующий вид:

$$
\begin{aligned}
&< \textit{система\_правил} >::=[< \textit{продукция} >]\\
&< \textit{продукция} >::=< \textit{условие} >\rightarrow< \textit{следствие} >\\
&< \textit{условие} >::=[< \textit{простое\_условие} >]\\
&< \textit{простое\_условие} >::=< \textit{объект} >< \textit{атрибут} >< \textit{предикат} >< \textit{значение} >\\
&< \textit{объект} >::= \textit{УП}_g,\ i=\overline{1,n}\\
&< \textit{следствие} >::=[< \textit{указание} >|< \textit{формула} >|< \textit{программа} >]\\
&< \textit{предикат} >::=|\neq|<|>|\leq|\geq\\
&< \textit{атрибут} >::=< \textit{приложение} >|< \textit{приоритет\_приложения} >|< \textit{услуга} >|\\
&|< \textit{бизнес\_процесс} >|< \textit{важность\_процесса} >|< \textit{параметр} >|< \textit{статус\_ИТС} >
\end{aligned}
\qquad (6.29)
$$

Параметрами могут быть любые характеристики сервисов, технологий, клиентских устройств. Значениями могут быть константы и имена. Команды выполняют действия над элементами рабочей области. Синтаксис команд зависит от команды. Применяются следующие три команды: modify; remove; add. Они выполняют действия над элементами рабочей области.

Содержание рабочей области составляют элементы рабочей области, которые задаются кортежами полного представления сети доступа и элементами упрощенного представления:

<элемент>::= (<класс объекта> Name <имя> [<атрибут><значение>]). (6.30)

Семантика продукционных систем традиционная, т. е. левые части продукций рассматриваются как условия, а правые части – как действия. Условия проверяются на истинность, т. е. на выполнимость на текущем состоянии сети доступа, которое задается элементами рабочей области.

Классификация объектов составляет важнейший элемент семантики. Целесообразно воспользоваться преимуществами структурирования. Вообще представление знаний при логическом подходе является неструктурированным, а вопросы семантики возникают лишь при интерпретации формул. В данном случае используется структуризация знаний, в основу которой положено традиционное

представление сети доступа. Но можно применить и средства представления знаний когнитивного направления.

На второй фазе, в процессе непосредственных вычислений подготовленное заранее упрощенное представление отрабатывается алгоритмично после поступления сообщения о неисправности. Для этого можно использовать быстрые алгоритмы обхода графов.

Особенности применения правил для импакт-анализа продемонстрируем на правилах распространения влияния неполадок, когда в качестве показателей качества используются параметры производительности [153]. Система элементарных иерархических подграфов приведена на рис. 6.8. Разработанные для подграфов правила адекватно учитывают структуру сети доступа, основные топологические способы соединения ее узлов, особенности коммуникационных технологий, настройки оборудования в зависимости от величины коэффициентов снижения скорости передачи во входящих линиях, наличия резервирования и других параметров. Сделанные максимально простыми для повышения быстродействия, эти правила формируют основу соответствующих продукций. В приведенных правилах для формального оценки влияния проблем на узлы и абонентов сети доступа введены коэффициенты $k_{i,j(n)}$ (6.11) и $K_{i,j}$ (6.14).

При обнаружении неполадок в сети выполняется их локализация и определяется влияние на абонентов путем подсчета с помощью соответствующих правил значений коэффициентов $K_{i,j}$ (6.14) для соответствующих узлов.

Соответствующие *Способу соединения а* (см. п. 6.2.4 и рис. 6.8,*а*) правила распространения с учетом (6.15) и (6.16) разобьем на две такие группы:

а) правила прямого распространения влияния неполадок:

$$\begin{aligned} &IF\ \text{tt.nod} = 1 \wedge \text{type.nod} = \text{‘…’} \wedge \text{class.nod} = \text{‘…’} \wedge \text{type.child.nod} = \\ &\text{‘…’}\ THEN\ \text{tt.child.nod} = 1 \\ &IF\ \text{tt.nod} = 1 \wedge \text{type.nod} = \text{‘…’} \wedge \text{class.nod} = \text{‘…’} \wedge \text{type.reference.nod} = \\ &\text{‘…’} \wedge \text{type.nod.reference} = \text{‘…’} \wedge \text{attribute.nod.reference} = \\ &\text{‘…’}\ THEN\ \text{tt.reference.nod} = 1 \end{aligned} \qquad (6.31)$$

б) правила обратного распространения влияния неполадок:

$$\begin{aligned} &IF\ \text{tt.nod} = 1 \wedge \text{type.nod} = \text{‘…’} \wedge \text{class.nod} = \text{‘…’} \wedge \text{type.parent.nod.} = \\ &\text{‘…’}\ THEN\ \text{tt.parent.nod} = 1 \\ &IF\ \text{tt.nod} = 1 \wedge \text{type.nod} = \text{‘…’} \wedge \text{class.nod} = \text{‘…’} \wedge \text{type.nod} = \\ &\text{‘…’} \wedge \text{type.nod.reference} = \text{‘…’} \wedge \text{attribute.reference.nod} = \\ &\text{‘…’}\ THEN\ \text{tt.reference.nod} = 1 \end{aligned} \qquad (6.32)$$

Для упрощения в этих правилах не определяются имена классов объектов и значения атрибутов. Здесь простые условия tt.nod = 1, tt.child.nod = 1, tt.parent.nod = 1 и tt.reference.nod = 1 означают, что признак наличия неполадок приписан соответственно узлу, дочерней вершине узла, родительской вершине узла, вершине, на которую в узле имеется ссылка. Другие простые условия понятны из контекста.

АУТСТИ

Правила формирования severity level имеют ту же структуру, но дополнительно устанавливают коэффициент влияния неполадки в соответствующем узле (дочернем или родительском, тому что ссылается, или тому, на который ссылаются).

Для *Способа соединения б* (см. п. 6.2.4 и рис. 6.8,*б*) правила распространения с учетом (6.17) имеют такой же вид, как и для предыдущего случая, а правила формирования severity level будут иметь вид

$$\begin{aligned} &IF\ \text{tt.nod}\_1 = 1 \wedge severity.nod\_1 = k \wedge \text{type.nod}\_1 = \text{'…'} \wedge \text{class.nod}\_1 = \\ &\text{'…'} \wedge \text{type.child.nod} = \text{'…'} \wedge \text{severity.nod}\_2 = m \wedge \text{type.nod}\_2 = \\ &\text{'…'} \wedge \text{class.nod}\_2 = \text{'…'}\ THEN\ \text{tt.child.nod} = 1 \vee \text{severity.child.nod} = f(k+m). \end{aligned} \tag{6.33}$$

Здесь используются те же простые условия, что и для *Способа соединения а*, но nod_1 и nod_2 обозначают двух разных родителей. Переменные severity.nod_1 и severity.nod_2 приобретают значение severity level для двух разных родителей. Здесь и далее с помощью функции $f(x)$ подчеркивается необходимость расчета коэффициентов по формулам (6.15)–(6.17).

Способ соединения в. С учетом коэффициента $R_{i,j(n)}$ (6.19) и условия (6.19) коэффициенты падения скорости в 1-й и 2-й линиях связи на выходе узла $U_{2,4}$ определяют по формуле (6.15) следующим образом:

$$k_{2,4(1)} = R_{2,4(1)} \times K_{2,4} \times k_{1,4(1)}, \tag{6.34}$$

$$k_{2,4(2)} = R_{2,4(2)} \times K_{2,4} \times k_{1,4(1)}. \tag{6.35}$$

Соответствующие этому способу правила распространения имеют такой же вид, как и в предыдущих случаях, а правила формирования severity level имеют вид

$$\begin{aligned} &IF\ \text{tt.nod} = 1 \wedge \text{severity.nod} = k \wedge \text{type.nod}\_1 = \text{'…'} \wedge \text{class.nod}\_1 = \\ &\text{'…'} \wedge \text{type.child.nod} = \text{'…'} \wedge \text{type.child.nod} = \text{'…'} \wedge \text{class.child.nod} = \\ &\text{'…'}\ THEN\ \text{tt.child.nod} = 1 \wedge \text{severity.child.nod} = f(k). \end{aligned} \tag{6.36}$$

Способ соединения г. При использовании резервных линий связи, на рис. 6.8,*г* такими линиями являются линии 2, выходящие из узлов $U_{2,5}$ и $U_{2,6}$. Вычисление значения коэффициента $k_{3,6(1)}$ проводится на основе выражения

$$k_{3,6(1)} = \begin{cases} K_{3,6} \times k_{2,6(1)}, \text{ если } K_{2,6} \text{ равен 1,} \\ K_{3,6} \times k_{2,6(2)}, \text{ если } K_{2,6} \text{ равен 0, а } K_{2,5} \text{ равен 1,} \\ 0, \text{ если } K_{2,6} = K_{2,5} = 0, \end{cases} \tag{6.37}$$

а коэффициента $k_{2,5(2)}$ – выражения (6.21).

При падении скорости в основной линии за счет поддержки резерва оценка коэффициента $k_{2,5(1)}$ проводится на основе выражения (6.22).

Состояние одного из узлов например $U_{2,6}$, может оказывать влияние на коэффициенты падения скорости во входящих линиях других узлов, здесь это – $U_{2,5}$, которые находятся на том же или на высших иерархических уровнях относительно рассматриваемого узла и не связанных с ним физически.

АУТСКИ

Соответствующие этому способу правила должны принадлежать такому множеству:

$$\begin{aligned} &IF\ \text{tt.nod} = 1 \wedge \text{severity.nod} = k \wedge \text{type.nod} = \text{‘…’} \wedge \text{class.nod} = \\ &\text{‘…’} \wedge \text{type.child.nod} = \text{‘…’} \wedge \text{sign.reserve\_parent} = \\ &1 \wedge \text{output.nod.reserve\_parent} = \\ &m \wedge \text{type.reserve\_parent} = \text{‘…’} \wedge \text{class.reserve\_parent} = \\ &\text{‘…’} \wedge \text{type.child.reserve\_parent} = \text{‘…’} \wedge \text{output.child.reserve\_parent} = \\ &n\ THEN\ \text{tt.child.nod} = 1 \vee \text{severity.child.nod} = f(k,m) \end{aligned} \quad (6.38)$$

– если есть резервная линия связи и для нее выделен запас производительности;

$$\begin{aligned} &IF\ \text{tt.nod} = 1 \wedge \text{severity.nod} = k \wedge \text{type.nod} = \text{‘…’} \wedge \text{class.nod} = \\ &\text{‘…’} \wedge \text{type.child.nod ‘…’} \wedge \text{sign.reserve\_parent} = \\ &1 \wedge \text{output.nod.reserve\_parent} = \\ &m \wedge \text{type.reserve\_parent} = \text{‘…’} \wedge \text{class.reserve\_parent} = \\ &\text{‘…’} \wedge \text{type.child.reserve\_parent} = \text{‘…’} \wedge \text{output.child.reserve\_parent} = \\ &n\ THEN\ \text{tt.child.nod} = 1 \wedge \text{severity.child.nod} = \\ &f(k+m) \vee \text{tt.child.reserve\_parent} = \\ &1 \vee \text{severity.child.reserve\_parent} = f(n-m) \end{aligned} \quad (6.39)$$

– если есть резервная линия связи, но для нее не выделен запас производительности и резерв поддерживается за счет дочерних вершин резервного родителя.

Здесь дополнительные простые условия sign.reserve_parent = 1 и tt.child.reserve_parent = 1 соответственно означают, что существует связь с резервным родителем и признак наличия неполадок приписан дочерней вершине резервного отца. Переменные severity.nod, severity.child.nod, severity.child. reserve_parent приобретают значение severity level для узла, дочерней вершины и дочерней вершины резервного родителя. Переменные out-put.nod.reserve_parent и out-put.child.reserve_parent приобретают значение определенных выше коэффициентов снижения скорости передачи.

Параметр оценки уровня сервиса – severity level соответствует ρ_{klm}.

Отображение Q_1 (6.27) задается неявно с помощью возможностей структурирования сети, заложенных в модели, и правил $P=\{P_1,\ldots,P_q,\ldots,P_Q\}$ продукционной системы, определяя для каждой структуры сети и комбинации неисправностей значения ρ_{klm}, $m=\overline{1,L}$.

Привязав правила к структуре сети доступа, т. е. определив задействованные правила для вершин графа и включив соответствующие коэффициенты распространения влияния неисправностей в упрощенное представление, вторую фазу можно свести к пересчету влияния неполадок на основе обхода графов специального вида. При этом дерево распространения влияния составляет, по сути, структуру данных для организации работы алгоритма обхода графов специального

вида. В процессе обхода указанный алгоритм должен также выполнить расчет оценок влияния неполадок на уровень обслуживания клиентов оператора на основе сохранившихся в вершинах упрощенного представления коэффициентов распространения.

Таким образом, возникает потребность в алгоритмах импакт-анализа, которые должны интегрировать такие обобщенные процедуры [142, 150, 161, 333]:

- структуризация полного представления сети и построение каркаса дерева распространения влияния (импакт-дерева) из полного представления сети;
- выбор продукционных правил для узлов дерева распространения влияния;
- определение коэффициентов пересчета для узлов дерева распространения влияния;
- непосредственный пересчет влияния.

К тому же, эти алгоритмы должны осуществлять расчеты в условиях жестких временных ограничений.

Рассмотрим идеи, которые можно положить в реализацию некоторых из перечисленных выше процедур, с целью минимизации времени расчетов и использования как можно меньших объемов памяти.

Структурирование полного представления сети и построение каркаса дерева распространения влияния. Назначение процедуры состоит в формировании на основе полного представления сети доступа графа специального вида – ярусно-параллельной формы. Так будет осуществлен переход от слабо структурированного представления элементов сетей доступа и их бинарных связей к хорошо структурированному упрощенному представлению, в котором формируется представление о сети доступа в терминах деревьев, поддеревьев, путей и т. д. Эти термины традиционно ассоциируются с такими компонентами представлений о сети доступа, как сеть, подсеть, путь доставки и др. Указанное дерево является основой визуализации сетей доступа с точки зрения импакт-анализа.

Таким образом, на вход процедуры подается полное представление в виде системы таблиц реляционной БД, а на выходе получаем структуру данных в виде графа специального вида.

Построение каркаса импакт-дерева завершается формированием графа (рис. 6.11), каждый узел которого отнесен к i-му, $i = \overline{1, I}$ уровню. На каждом пути от i-го к $(i + n)$-му, $n = 2,3,\ldots$ уровню должны быть определены узлы на промежуточных уровнях. Если фактические узлы отсутствуют, то вводятся фиктивные узлы. Необходимость введения дополнительных узлов определяется конкретным алгоритмом, осуществляющим непосредственный пересчет влияния.

Выбор продукционных правил. В основу реализации выбора положен механизм вывода в продукционных системах. В общем функционирование механизма вывода сводится к последовательному выполнению простых операций проверки условий правил, пока не будет достигнуто целевое состояние системы. В данном случае это означает, что завершится построения алгоритма перерасчета для всех клиентов.

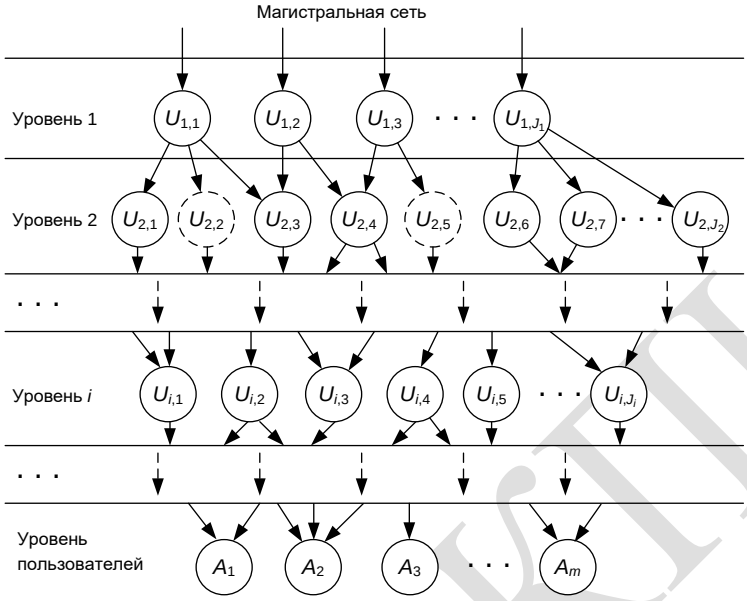


Рис. 6.11. Граф распространения влияния неисправностей с фиктивными узлами

Поскольку в реальных системах условия применения правил могут пересекаться, т. е. одновременно несколько правил, условия которых выполняются, могут претендовать на применение, то возникают конфликтные ситуации. Для принятия решений о применении правил в нужном порядке предлагается использовать отношения предпочтения между правилами и первым применять наиболее приемлемое правило из правил, условная часть которых выполняется.

С целью повышения эффективности ПИА рассмотрим возможные варианты реализации представлений и механизмов, т. е. варианты организации функционирования ПИА. Общая идея обеспечения эффективности пересчета показателей влияния неполадок элементов сети доступа на другие элементы сети и, наконец, на качество обслуживания клиентов заключается в том, чтобы перейти от повторяющегося последовательного просмотра рабочей области и области правил к комбинированному применению таких трех приемов, детально описанных выше в 6.3.2: 1) индексация элементов рабочей области; 2) структурирование системы правил; 3) привязки правил к узлам упрощенного представления сети доступа.

Определение коэффициентов пересчета. Когда дерево распространения влияния построено, тем самым определен общий характер выполнения расчетов. Но существуют альтернативы выполнения расчетов и после построения дерева. Поэтому целесообразно привести общее описание метода расчета влияния неполадок, который будет воплощен в нескольких вариантах алгоритмов реализации, построенных на особенностях соответствующих структур данных.

Целесообразно хранить граф в ОЗУ сервера ПИА в виде списка ребер, содержащего информацию о начальной и конечной вершине ребра, а также дополнительную информацию (тип, коэффициенты и т. п.), необходимую для расчетов. Элементы массива для хранения множества U должны содержать информацию об уровне, которому принадлежит узел, и коэффициенте неисправности.

Выбор продукций для узлов импакт-дерева осуществляется, исходя из используемых коммуникационных технологий, типа и назначения узлов, протоколов маршрутизации и других компонентов сетей доступа.

6.3.5. Быстрые алгоритмы оценки влияния неисправностей

Определившись с общим подходом к построению ПИА, рассмотрим особенности реализации подсистемы, построим для них теоретические оценки и экспериментально проверим работоспособность предложенного подхода. Сравнение реализаций позволит выработать рекомендации по их внедрению в коммерческие продукты.

Стоит отметить, что существует ряд технических аспектов реализации предложенного подхода, которые влияют на алгоритмическое обеспечение ПИА. Во-первых, совместное использование списков, деревьев и упрощенного представления сети доступа потенциально может обеспечить высокую эффективность пересчета показателей влияния неполадок. Во-вторых, еще больше подчеркивает надежду на высокую эффективность то обстоятельство, что упрощенное представление можно расположить в ОЗУ. Таким образом, исчезает необходимость в постоянном обращении к значительно медленной внешней памяти и появляется возможность обеспечить высокую эффективность пересчета. В-третьих, чтобы осуществлять анализ влияния неисправностей на уровень обслуживания клиентов в крупных сетях доступа в ОЗУ, необходимо иметь достаточно эффективную систему кодирования элементов сети доступа.

На основе приведенных концепций для решения проблемы быстрого импакт-анализа с учетом технических аспектов были разработаны алгоритмы импакт-анализа и соответствующие реализации.

В алгоритмах импакт-анализа выделяются два класса подалгоритмов:

- формирования модели (ПФМ);
- выполнения модели (ВМП).

Алгоритмы и подалгоритмы делятся на:

1) алгоритмы типа *A* (их подалгоритмы ПФМ-А и ВМП-А), использующие особенности хранения модели в БД на жестких дисках;

2) алгоритмы типа *B* (их подалгоритмы ПФМ-В и ВМП-В), использующие преимущества хранения упрощенного представления (модели) в ОЗУ.

Для ПФМ входом является полное представление и система продукций, а выходом – упрощенное представление – модель.

Для ВМП входом является модель, а выходом – значения коэффициентов ухудшения уровня услуги ρ_{klm}, $m = \overline{1, L}$.

Алгоритмы типа *A*.

Подалгоритм AS1 формирования упрощенного представления.

Поскольку в алгоритме предполагается использовать поуровневый обход графа, в этой процедуре построения графа каждому узлу необходимо приписать определенный уровень, исходя из количества промежуточных элементов между ним и корнем. Такой граф изображен на рис. 6.11. Одно из ограничений такого подхода заключается в том, что на каждом пути от верхнего уровня к вершинам уровня пользователей должны быть определены узлы на каждом промежуточном уровне. Таким образом, для связей, которые объединяют вершины несмежных уровней, для отображения непосредственной связи между узлами необходимо

вводить фиктивные узлы на каждом промежуточном уровне (на рис. 6.11 фиктивные узлы $U_{2,2}$ и $U_{2,5}$ изображены пунктирными линиями). Расчеты для этих узлов не выполняются и не влияют на показатели сети доступа.

Граф целесообразно хранить в памяти в виде списка ребер, каждый элемент которого содержит информацию о начальной и конечной вершинах соответствующего ребра, а также дополнительную информацию (тип, коэффициенты и т. п.), необходимую для расчетов. Для узлов будет использоваться отдельный массив, каждый элемент которого содержит информацию об уровне, которому принадлежит узел, а также коэффициент неисправности и место для промежуточных данных, необходимых для выполнения расчетов в узле.

Как отмечалось выше, в основу осуществления выбора правил положим механизм вывода в продукционных системах. Функционирование механизма вывода сводится к последовательному выполнению операций проверки условий правил, пока не будет определены задействованные правила для каждой вершины графа. Правила разбиты на простые условия, каждому из которых приписан уникальный идентификатор. Сами же правила представляют собой коллекции простых условий, соединенных логическим оператором конъюнкции. Таким образом, при считывании очередного атрибута узла значение атрибута сравнивается с простым условием соответствующего типа. При выполнении условия соответствующий идентификатор добавляется в коллекцию узла. При этом проверяется, не сработало ли какое-то правило. В случае срабатывания правила коллекция объекта очищается, а на выходных ребрах узла проставляются коэффициенты согласно данному правилу. Такая отработка правил позволяет выполнить необходимые расчеты и определить коэффициенты за один просмотр таблицы атрибутов, а также минимизировать использование памяти. Для разрешения конфликтных ситуаций применяется механизм приоритета правил. Правила пересматриваются в определенном порядке, применяется первое из них, условие которого выполняется.

Определение коэффициентов пересчета для узлов дерева распространения влияния осуществляется путем формирования необходимых для определения влияния коэффициентов распространения по правым частям задействованных для вершины правил.

Расчет влияния неполадок осуществляется поуровнево. Таким образом, при выполнении расчетов для текущего узла имеется уверенность, что уже выполнены расчеты для всех узлов предыдущего уровня, которые влияют на текущий узел.

Итак, общая идея обеспечения эффективности пересчета показателей для задачи импакт-анализа заключается в том, чтобы перейти от повторяющегося последовательного просмотра рабочей области и области правил к комбинированному применению таких трех приемов:

1) индексация элементов рабочей области: это позволит подключить ко всем простым условиям элементы рабочей области, после чего просматривать только эти списки. Для каждого правила создается счетчик количества простых условий, начальным значением которого является 0. При изменении рабочей области в его значение добавляется или отнимается 1 в зависимости от того, добавляется или удаляется элемент рабочей области, удовлетворяющей какому-то простому условию соответствующего правила. Таким образом, сравнение счетчика с количеством простых условий правил позволяет быстро отобрать правила, условия которых выполняются. С каждым простым условием хранится список элементов

рабочей области, которым она удовлетворяет. При изменении элементов рабочей области списки корректируются. При интерпретации пересматриваются только списки и готовится перечень конфликтных правил;

2) структурирование системы правил: в памяти строится дерево описания простых условий, вершины которого соответствуют признакам простых условий, а листья – продукциям, простые условия которых выполняются на пути от корня. ПИА содержит информацию об узлах, соответствующих текущему состоянию рабочей области, и модифицируется при появлении новых элементов рабочей области;

3) привязка правил к узлам упрощенного представления сети доступа: ссылки на правила и счетчики хранятся в каждом узле упрощенного представления сети доступа.

Построение дерева распространения влияния определяет общий характер выполнения расчетов, однако эффективность расчета показателей зависит от его представления и алгоритмов обхода. Как уже отмечалось, позитивному влиянию первого фактора содействует использование специальных структур представления деревьев в ОЗУ. Позитивное влияние второго фактора базируется на построении алгоритмов обходов деревьев распространения неполадок, использующих особенности этого представления.

Алгоритмы типа *A* ориентированы на традиционное размещение упрощенного представления. Именно для формирования такого представления используется ***подалгоритм AS1***.

Вход: Полное представление (таблицы инвентарной базы) и система продукций.

Выход: Упрощенное представление.

Шаг 1. Последовательный просмотр кортежей таблицы Objects и формирование соответствующих вершин и ребер упрощенного представления.

Шаг 2. Последовательный просмотр кортежей таблицы Reference и формирование соответствующих вершин и ребер упрощенного представления.

Шаг 3. Выбирается очередной кортеж таблицы Parameters.

Шаг 4. Находится и отмечается в упрощенном представлении в соответствующей вершине простое условие, которое содержится в кортеже таблицы Parameters.

Шаг 5. Находится и отмечается в таблице продукций в соответствующих правилах простое условие, которое содержится в выбранном кортеже таблицы Parameters.

Шаг 6. Проверяется условие: имеется ли продукция, все простые условия которой выполнены.

Если да, то реализуется правило в упрощенном представлении с формированием соответствующих полей в соответствующих ребрах и вершинах со значениями коэффициентов для расчета влияния неполадок. Удаляются отметки простых условий в соответствующей вершине упрощенного представления. Удаляются отметки простых условий в соответствующем правиле.

Если нет, то переходим на шаг 5.

Шаг 7. Если все кортежи таблицы Parameters рассмотрены, то выдаются показатели импакт-анализа и конец работы, иначе – переход на шаг 3.

При изменении конфигурации сети из инвентарной базы поступает соответствующий сигнал. В этом случае будут выполнены соответствующие записи в

списке ребер и массиве вершин, осуществлен перерасчет подчиненных узлов. При добавлении нового пользователя расчет проводится только для него.

Подалгоритм АЕ1 выполнения упрощенного представления. Идея алгоритма базируется на использовании традиционной идеи обхода дерева распространения, которое хранится в базе данных на внешних дисках и загружается блоками в буфер в ОЗУ. В основу алгоритма положен «поиск в ширину» (BFS) для графов с учетом специфики дерева распространения.

Вход: Упрощенное представление (таблицы: *device_type* – для описания типов узлов сети; *device* – для описания узлов сети; *parent_child* – для описания направленных связей узлов *id_parent* (родительская вершина) с *id_child* (дочерняя вершина). В двух последних таблицах хранятся: *koef* – степень снижения скорости передачи на выходах узла; *kin_sum* – сумма всех коэффициентов линий связи, входящих в узел; $k_{ij} \in [0, 1]$ – коэффициент снижения скорости передачи на линии от *i*-го узла к *j*-му; K_i – степень снижения скорости передачи на выходах узла U_i относительно нормативного; $R_{ij} \in [0, 1]$ – коэффициент распределения полосы пропускания между линиями связи, которые выходят из *i*-го узла.

Выход: Перечень показателей воздействия для каждого клиента.

Шаг 1. Подается идентификатор узла (FIRST_ID) и его новый коэффициент (KO_).

Шаг 2. С помощью SELECT-запросов создаются индексированные массивы: masD_KO(id) – коэффициент узла с идентификатором id; masD_KIN(id) – kin_sum узла с идентификатором id; masPC_pid(i), masPC_cid(i) (находятся из таблицы PARENT_CHILD, как идентификаторы родительского и дочернего узлов в отсортированном порядке по идентификатору родителя) masPC_k(i), masPC_r(i) (находятся из таблицы PARENT_CHILD, как коэффициенты k_{ij} и R_{ij} иерархической связи между родительским и дочерними узлами в отсортированном порядке по идентификатору родителя); masD_begin(i) – первое вхождение родителя с заданным идентификатором в массив masPC_pid; masD_NChild(i) – количество детей у родителя с заданным идентификатором.

Шаг 3. Осуществляется поиск «в глубину» в графе с вершины FIRST_ID: находим циклы (в массив mas_cycle (i) записывается количество родителей для каждого узла *i*).

Шаг 4. Пока masParent не пустой, осуществляется поиск в ширину, начиная с вершины FIRST_ID (массив masParent содержит родительские вершины, из которых осуществляется переход):

1) определяются все дети (узлы, в которые можно попасть из узлов masParent);

2) перечисляются для связи между родительским и дочерними узлами новые коэффициенты k_{ij} линий (записываются в masPC_k) и коэффициенты kin_sum для дочерних узлов (записываются в masD_KIN) по формулам:

$$kin\_sum_j = \sum_{i \in P} k_{ij}; \tag{6.40}$$

$$k_{ij} = K_i \cdot \min\{kin\_sum_i, 1\} \cdot R_{ij}. \tag{6.41}$$

В формуле (6.40) сумма берется по всем родительским узлами для *j*-го узла;

3) для всех дочерних узлов значение mas_cycle уменьшается на 1. Если для какого-то дочернего узла это значение меньше 1, то он записывается в masChild;

4) принимается masParent = masChild.

Алгоритм осуществляет перерасчет коэффициентов пропускной способности k_{ij} линий связи при изменении скорости передачи на выходах K_i для i-го элемента. С целью упрощения расчетные формулы предложены для линий связи с одинаковой скоростью передачи. Стоит также отметить, что шаг 3 (поиск «в глубину») требуется для того, чтобы базовый алгоритм «поиска в ширину» (шаг 4) не проходил вершины по несколько раз, что может обуславливать сложность $O(a^n)$, где n – количество вершин, $a > 1$.

Алгоритмы типа В.

Подалгоритм BS2 формирования упрощенного представления.

Подалгоритм *BS2*, как и алгоритм *BS1*, предназначен для построения дерева на основе полного представления, которое хранится на дисках, и системы правил, которая хранится в ОЗУ. Однако он ориентирован на хранение дерева в ОЗУ. От алгоритма *AS1*, приведенного выше и в работе [163], он отличается структурой представления данных в ОЗУ и особенностями обработки продукционных правил.

Основная идея построения подалгоритма *BS2* заключается в выборе таких структур данных и последовательности шагов их обработки, чтобы по данным инвентарной БД за одно последовательное считывание ее таблиц выстроить все уровни дерева распространения влияния неполадок в сети. Загрузка данных из базы данных осуществляется с помощью драйвера JDBC, программная реализация работает с базой данных напрямую.

Вход: Полное представление (таблицы) и система продукций.

Выход: Упрощенное представление (структуры для дерева распространения).

Состав полей таблиц инвентарной БД приведен выше. Структуры для представления дерева распространения концентрируются вокруг класса Unit, объекты которого для узлов сети хранят списки наследников и предков по основной и резервной линиях связи, а также информацию о состоянии вычислений, которая поступила в узел. Правила являются объектами класса Rule, в списках которых содержатся простые условия {атрибут (левая часть), значения (правая часть), предикат (знак сравнения)} и действие, которое необходимо выполнять, если выполняются все простые условия.

Алгоритм расчета состоит из трех частей: обработка таблицы «object»; обработки таблицы «reference» и работа с правилами. Поскольку все алгоритмы реализуют идею одноразового прохода таблиц инвентарной БД, то остановимся лишь на работе с правилами, для которых выполнены все простые условия.

Если для правила все наследники равны, то показатель влияния неполадки на узел вычисляется по формуле $p_u = \sum_{s \in S} p_s \times K_s$ (сумма берется по всем родителям $s \in S$ узла u).

Если для правила все наследники равны с коэффициентом r, то показатель влияния неполадки на узел вычисляется по формуле $p_u = \sum_{s:\textit{родители}} r_s(u) \times p_s \times K_s$ (сумма берется по всем родителям $s \in S$ узла u; $r_s(u)$ – коэффициент передачи линии между узлами u и s).

Если правило связано с перераспределением коэффициентов передачи линий связи, выходящих из узла u, то соответствующим образом выполняется перераспределение. Например, правила с выполняемыми действиями «Удалить ресурс всех прямых наследников» и «Обеспечить полностью ресурсом всех прямых

наследников» приводят к установлению коэффициентов передачи всех соответствующих линий из узла *u,* равными 0 и 1 соответственно.

Из всех правил для узла *u* класс Unit будет включать только те правила, которые содержат вхождения простых условий с типами, названиями и другими параметрами, которые соответствуют аналогичным характеристикам узла. Действие, ассоциированное с соответствующим правилом, выполняется только для включенных правил.

Легко понять, что формирование графа в худшем случае выполняется за время O(n+m), где n – количество вершин, а m – количество ребер. Время вычисления полосы пропускания в данном узле на основе правил определим, предположив худший вариант – все правила прошли проверку, т. е. должно выполняться каждое из r правил в каждом узле. Легко проверить, что время выполнения каждого правила пересчета мощности не превышает O(n), поскольку в приведенных выше формулах пересчета функция вычисления полосы пропускания вызывается рекурсивно для каждого предка узла, поэтому вызовов будет не больше, чем общее количество вершин. Время выполнения правила распределения пропускных способностей линий, выходящих из узла, также очевидно составляет O(количество наследников) и не превышает O(n).

Подалгоритм BE1 выполнения упрощенного представления. Рассмотрим алгоритм на примере, приведенном на рис. 6.9. Пусть узел $U_{1,3}$ вышел из строя на 50%. Алгоритм работает следующим образом:

Шаг 1. Рассчитывается входной коэффициент для узла $U_{1,3}$. Поскольку узел находится на уровне 1, то входной коэффициент всегда будет равен 1.

Шаг 2. Этому узлу устанавливается новый коэффициент неполадки, равный 0,5.

Шаг 3. Перебираются выходные ребра текущего узла и рассчитываются коэффициенты пропускной способности по формуле: $k = P \times K$, где P – пропускная способность на входе узла, K – коэффициент неполадки узла. Также, при обходе ребра, на следующем (втором) уровне помечаются узлы, на которые влияет эта неполадка. В нашем случае это будет узел $U_{2,4}$ и фиктивный узел $U_{2,5}$.

Шаг 4. Расчет первого уровня закончен. Второй уровень делается текущим.

Шаг 5. В цикле перебираются узлы, помеченные в шаге 3. Поскольку узел $U_{2,5}$ – фиктивный, то расчеты в нем не выполняются, а только помечаются узлы следующего уровня, на которые может повлиять эта неполадка.

Шаг 6. Рассчитываем неполадку в узле $U_{2,4}$. Перебираем массив входных ребер и суммируем коэффициенты пропускной способности (S). Входной коэффициент в узле рассчитывается по формуле: $u = \min(S, 1)$. Если для всех пострадавших узлов текущего уровня $u = 1$, то расчеты прекращаются, поскольку нижние уровни будут находиться в полностью функционирующем состоянии и действие неполадки «погасло».

Шаг 7. Переходим на шаг 1, но на шаге 2 устанавливаем коэффициент неполадки уже текущего узла. В данном случае для $U_{2,5}$ он равен 1 (узел полностью работоспособен).

Шаг 8. Действия повторяем до тех пор, пока текущим не станет уровень пользователей.

Шаг 9. Для каждого пользователя перебираем массив входных ребер и выполняем для него действия шага 6.

Шаг 10. Формируем массив пострадавших пользователей с коэффициентом неполадки.

Резервные узлы учитываем с помощью правил, приведенных выше для *Способа г* соединения узлов сети доступа. Если у пострадавшего узла имеются входные резервные ребра, то влияние неполадки может быть уменьшено за счет резервного ребра. При этом, в некоторых случаях влияние неполадки может распространиться на дочерние вершины резервного родителя.

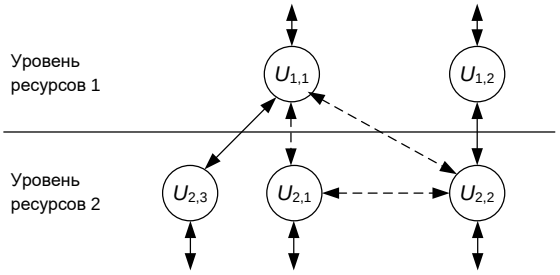


Рис. 6.12. Пример петли

Поскольку направление ребра указывает на увеличение уровня, а трафик может передаваться и в обратном иерархии направлении, то может возникнуть проблема расчета петель. Пример петли приведен на рис. 6.12.

Здесь петля обозначена пунктиром. Связь на одном уровне (между $U_{2,1}$ и $U_{2,2}$) будет реализована с помощью двух ребер. Связь вида $U_{1,1} \leftrightarrow U_{2,2}$ реализовать практически невозможно. Поэтому необходимо заменить ее на эквивалентную. Эквивалент фрагмента сети с петлей приведен на рис. 6.13.

Таким образом, в предлагаемой модели будут существовать связи только вниз или на одном уровне, что позволит успешно применять описанный выше алгоритм работы. Для эквивалентной замены для пересчета понадобится коэффициент R для ребра между узлами $U_{1,1}$ и $U'_{2,2}$, указывающий на долю общей выходной полосы узла $U_{2,2}$, которая будет переключена на это ребро. Также понадобится специальный эквивалентный узел $U'_{2,2}$, который будет копией узла $U_{2,2}$, но только с одним выходом. Отметим, что одноуровневые связи в графе могут быть только резервными.

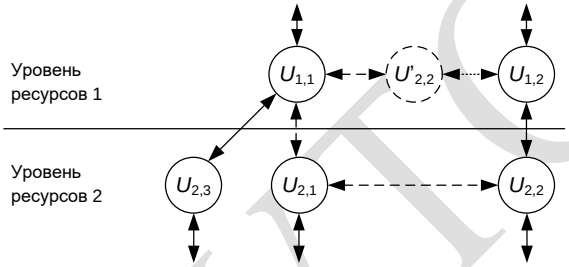


Рис. 6.13. Эквивалентное преобразование петли

Подалгоритм BE2 выполнения упрощенного представления. Алгоритм работает по схеме обхода построенного с помощью алгоритма *BS2* дерева распространения, которое хранится в ОЗУ. От алгоритма *AE1* отличается структурой представления дерева распространения и особенностями обхода его вершин и ребер.

Используется идея поуровневого обхода дерева распространения. Объектно-ориентированная реализация предоставляет базовые классы, которые представляют информацию о влиянии неполадки на элементы сети и пользователей за приемлемое время.

Структура хранения данных базируется на классе Unit для описания узлов сети, особенности которого приведены при определении алгоритма *BS2*. Класс Clients расширяет класс Unit для описания узлов-клиентов. Когда объекту, который соответствует узлу, поступила информация о состоянии вычислений в родительских узлах, вычисляются его коэффициенты и готовится информация для расчета в дочерних узлах K_{ij} по формулам, аналогичным (6.40) и (6.41), с учетом основных и альтернативных линий.

Вход: Упрощенное представление.

Выход: Перечень показателей воздействия для каждого клиента.

Шаг 1. Определение начальных параметров расчетов.

Шаг 2. Фиксация первого уровня дерева распространения.

Шаг 3. Фиксация следующего узла на выбранном уровне дерева распространения.

Шаг 4. Расчет показателей влияния неполадок выбранного узла.

Шаг 5. Если не все узлы уровня просмотрены, то переход на шаг 3, иначе – на шаг 6.

Шаг 6. Если не все уровни просмотрены, то фиксируется следующий уровень и осуществляется переход на шаг 3, иначе – на шаг 7.

Шаг 7. Сведение и вывод результатов.

Более подробно рассмотрим важные шаги алгоритма.

Поскольку значение показателя влияния в узле можно получить, исходя только из показателей его родительских узлов, которых может быть несколько на разных уровнях, то возникает проблема циклов. Если не решить эту проблему, сложность алгоритма будет описываться экспоненциальной функцией, что делает невозможным выполнение расчетов за приемлемое время для нескольких миллионов узлов. Чтобы избежать этого, в класс Unit добавлена булева переменная wasInit, значение которой определяет необходимость нового вычисления значения показателя влияния неисправности K_{ij} для соответствующего узла (предыдущее значение показателя воздействия также хранится в узле).

Очевидно, что изменение значения показателя влияния K_{ij} в узле U_i влияет только на всех его прямых и косвенных наследников. Это приводит к изменению k_{ij} коэффициентов на линиях, исходящих из U_i. Изменение значения показателя влияния K_{ij} в узле U_i может повлиять на какой-то его родительский узел V, если в U_i данные поступают только от резервных линий и V является именно тем родительским узлом, от которого по резервной линии в U_i поступают данные.

При выполнении приведенных выше условий сам показатель влияния в узле V не изменяется, однако изменяются коэффициенты на линиях, которые выходят из V. Поэтому при изменении значения показателя влияния в узле необходимо, чтобы в нем и во всех его преемниках wasInit = false. При необходимости также нужно установить wasInit = false в его родительском узле V, о котором шла речь выше, а также во всех наследниках узла V. В дальнейшем считаем, что показатель влияния в классе содержит «правильное» значение полосы пропускания, если wasInit = true. Это позволит при повторном просмотре узла просто возвращать вычисленное ранее значение показателя, однако порождает проблему быстрой установки в узлах и всех их наследниках (прямых и косвенных) значения wasInit = false. Для ее решения нам понадобится следующее простое утверждение.

Утверждение 1. Если в узле wasInit = false, то во всех его преемниках (прямых и косвенных) также wasInit = false.

Доказательство. Для доказательства достаточно показать, что ситуация, когда на направленном пути из узла верхнего уровня к узлу нижнего уровня после отметок true может стоять пометка false, невозможна. В дальнейшем будем полагать, что узел обозначен как *f*, если для него wasInit = false. Если для узла wasInit = true, то будем считать, что узел обозначен как *t*.

Пусть некоторый наследник обозначен как *t*. Тогда он был обозначен как *t* уже после того, как самый верхний узел был обозначен как *f* (иначе бы все

наследники верхнего узла были бы обозначены как *f* и самый нижний также был бы обозначен как *f*). Нижний узел мог быть обозначенным как *t* только в результате пересчета показателя влияния. Однако при пересчете показателя влияния узла также выполняется перерасчет для всех его родительских узлов (прямых и косвенных). Поэтому самый верхний узел был бы обозначен как *t*. Таким образом, указанная ситуация невозможна. *Утверждение доказано*.

Доказанное утверждение позволяет предложить такой алгоритм обозначения узлов:

Шаг 1. Если узел уже обозначен как *f*, то переходим к следующему.

Шаг 2. Иначе обозначаем узел как *f* и выполняем те же шаги (1–2) для каждого наследника узла.

Отметим, что при добавлении узла *u* к узлу *V* (т. е. узел *u* становится наследником узла *V*) необходимо установить для узла *u* значение wasInit = false, поскольку у узла *u* изменились родители и, как следствие, текущее значение показателя влияния недействительно.

Легко установить, что для оценки времени выполнения можно использовать сложность алгоритма обозначения вершины *u* и ее наследников (прямых и косвенных). Она в среднем составляет $O(n_u(t))$, где $n_u(t)$ – количество наследников *u*, которые обозначены как *t*, а в худшем случае – $O(m_u(t))$, где $m_u(t)$ – количество ребер, которые выходят из наследников *u*.

Оценка памяти, необходимой для выполнения: $O(n + m)$.

Анализ алгоритмов и результаты экспериментальных исследований. Подалгоритмы класса ПФМ [142, 150, 161, 333] AS-1, BS-1 и BS-2 не отличаются разнообразием и сводятся к последовательному просмотру кортежей таблиц *Objects* и *Reference* и формированию вершин и ребер модели. По кортежам таблицы *Parameters* в модели в соответствующей вершине находятся и отмечаются простые условия, содержащиеся в кортеже. Простые условия находятся и отмечаются в таблице продукций, после чего проверяется условие: имеется ли продукция, все простые условия которой выполнены. Если да, то правило реализуется в модели с формированием соответствующих полей в соответствующих ребрах и вершинах со значениями коэффициентов для импакт-анализа. Если нет, то осуществляется переход на предыдущий шаг. После просмотра всех кортежей таблицы *Parameters* выдаются показатели импакт-анализа.

При этом подалгоритмы ПФМ-В различаются структурой представления данных в ОЗУ и особенностями обработки продукционных правил. В худшем случае время выполнения не будет превышать $C + r \cdot D$, где D – максимальное время из множества $\{D_i\}$, $i = \overline{1, r}$ (D_i – время выполнения i-го правила). Поскольку $D = O(n)$ и r – достаточно малое число, то $C + r \cdot D = O$(количество вершин). Оценка памяти, необходимой для выполнения: $O(n^2)$.

В условиях жестких временных ограничений представляет интерес подалгоритм класса ПФМ-В BS-2, предложенный в работе [142]. Для его реализации выбраны такие структуры данных и последовательности шагов их обработки, чтобы по данным инвентарной БД за одно последовательное считывание ее таблиц выстроить все уровни импакт-дерева. Формирование графа в худшем случае выполняется за время $O(n + m)$, где n – количество вершин, а m – количество ребер.

Экспериментальные исследования ПФМ-В BS-1 [161] проводились при различных вариантах структуры сети, характеристиках оборудования и технологий,

для различного количества пользователей. При тестировании BS-1 случайно генерировались деревья с разным количеством элементов и связей. Во время эксперимента случайным образом эмулировался выход из строя верхнего узла. Результаты исследования BS-1 приведены на рис. 6.14.

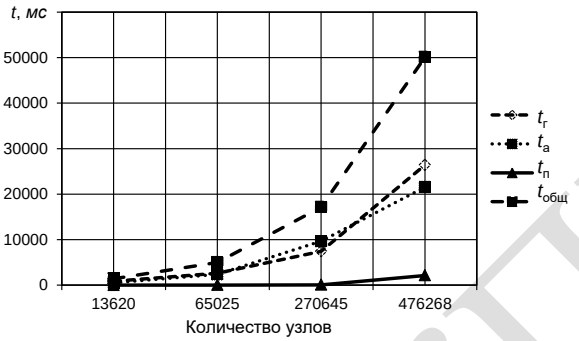


Рис. 6.14. График зависимости времени построения модели от количества узлов ($t_г$ – время генерации дерева, $t_а$ – время чтения аргументов; $t_п$ – время обработки правил; $t_{общ}$ – суммарное время)

Бóльшим разнообразием характеризуются подалгоритмы класса ВМП. Так, в работе [161] разработан базовый подалгоритм класса ВМП – BE-1, приведенный выше, основанный на поуровневом расчете коэффициентов узлов, на которые повлияла неисправность. Предварительно помечаются узлы нижерасположенного уровня, представляющие интерес для импакт-анализа. После чего узлы нижерасположенного уровня становятся текущими. Перебираются выходные ребра текущего узла и рассчитываются соответствующие коэффициенты, учитывающие влияние неисправностей. При обходе ребер помечаются узлы, на которые влияет неисправность. Действия повторяются до тех пор, пока текущим не станет уровень пользователей. Для каждого пользователя перебирается массив входных ребер, формируется массив пострадавших пользователей и значений ρ_{klm}, $m = \overline{1, L}$.

В модели существуют связи вниз или на одном уровне, а поскольку трафик может передаваться в обоих направлениях, то может возникнуть проблема расчета петель. Считается, что одноуровневые связи могут быть только резервными, а влияние неисправностей передается только сверху вниз. В BE-1 одноуровневые связи реализуются с помощью двух ребер, что позволяет избежать зацикливания. При эквивалентной замене для пересчета вводятся соответствующие коэффициенты для ребер.

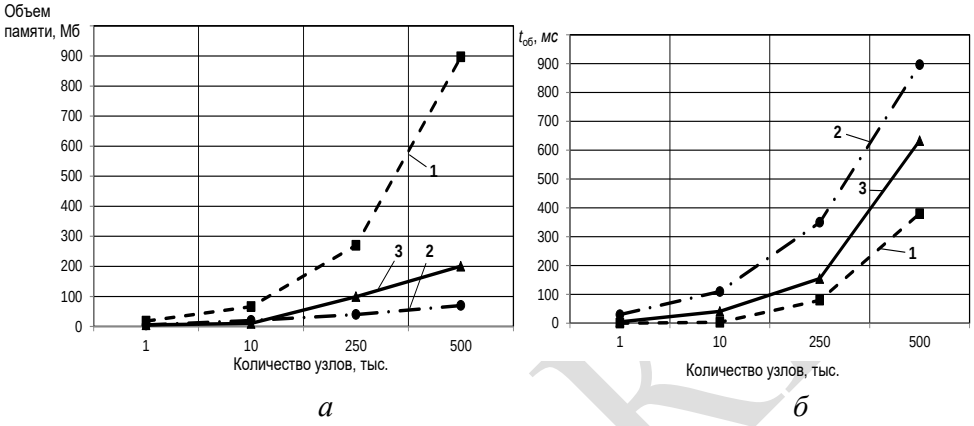
В работе [142] предложен подалгоритм класса ВМП-А – AE-1, приведенный выше, базирующийся на использовании алгоритмов обхода импакт-дерева, которое хранится в БД на внешних дисках и загружается блоками в буфер ОЗУ.

Сложность AE-1 составляет: O($n + m$), где n – количество вершин, m – количество ребер, на создание массивов требуется время O(m·ln(m)), что связано с необходимостью сортировки. Оценка памяти: O(d), где d – объем буфера в ОЗУ, необходимого для сохранения порции данных из БД.

Подалгоритм класса ВМП-В – BE-2, предложенный в работе [142], отличается от подалгоритма BE-1 способом представления импакт-дерева и особенностями обхода его вершин и ребер. Его объектно-ориентированная реализация, использующая поуровневый обход импакт-дерева, предоставляет базовые

классы, которые позволяют осуществлять импакт-анализ за время, удовлетворяющее ограничению (6.26).

Результаты исследования предложенных подалгоритмов по объему используемой памяти ОЗУ и времени вычислений приведены на графиках рис. 6.15. Кривые 1 на обоих графиках получены в результате применения подалгоритма BE-2, 2 – BS-2, 3 – выполнение по алгоритму BE-1.



а *б*

Рис. 6.15. Результаты сравнения алгоритмов: *а* – по объему используемой памяти; *б* – по времени вычислений

Подалгоритм BE-2 быстрее работает, но требует большего объема ОЗУ по сравнению с подалгоритмом BE-1. Разница в показателях эффективности двух алгоритмов, базирующихся на хранении представления в ОЗУ, свидетельствует о том, что алгоритмы импакт-анализа можно усовершенствовать.

Расширение области применения эффективной комбинации продукционного и алгоритмического подходов на широкий класс ТКС, построенных с использованием разных сетевых протоколов и предлагающих различные услуги, требует разработки соответствующих наборов правил и подтверждения работоспособности предложенного метода с выполнением ограничений на время анализа. Такие исследования проведены в работе [150] и параграфе 6.4, где предложен метод импакт-анализа в IP-сетях с протоколами RIP и OSPF, использующий продукционно-алгоритмический подход, и разработаны алгоритмы класса – ПФМ BS-3 и класса – ВМП BE-3.

На основе анализа существующих решений, структурно-технологических и сервисно-функциональных особенностей сетей для общего подхода к решению задач импакт-анализа можно привести следующие рекомендации по применению методов и алгоритмов:

1. При жестко заданной структуре ТКС, предопределенных условиях переключения на резерв и отсутствии логики взаимодействия узлов целесообразно использовать алгебраический подход к импакт-анализу.

2. В противном случае следует использовать методы на основе продукционного или продукционно-алгоритмического подходов, которые на графах специального вида формулируют модели для импакт-анализа. Для решения задач можно использовать несколько вариантов алгоритмов класса ПФМ (AS-1, BS-1, BS-2, BS-3) и ВМП (AE-1, BE-1, BE-2, BE-3), построенных на приемах работы с графами и алгебраическом подходе.

3. Для сетей больших размеров хорошо зарекомендовали себя алгоритмы выполнения расчетов в ОЗУ, такие как BS-1, BS-2, BE-1 и BE-2, у которых граф хранится в ОЗУ в виде списка ребер и вершин.

4. Для графов, показатель соотношения ребер к вершинам, которых Ψ не больше пяти, где $\Psi = \Sigma E / \Sigma U$, следует использовать алгоритмы BS-1 и BE-1, ориентированные на представлении ребрами основной информации об импакт-дереве. В ребрах прописана информация о выше- и нижерасположенных узлах, типах связей, коэффициентах влияния и других параметрах.

5. Для импакт-деревьев, показатель соотношения ребер к вершинам которых Ψ больше пяти, следует использовать алгоритмы BS-2 и BE-2, ориентированные на представление основной информации об импакт-дереве в вершинах (класс Unit). Объекты этого класса для каждой вершины содержат уровень, коэффициент распространения влияния неисправностей по исходящим ветвям, полученный на основании обработки соответствующих правил, связи с зависимыми вершинами и другую нужную информацию.

6. Для импакт-анализа в сетях с количеством элементов N до 500 тыс. можно использовать алгоритмы AS-1 и AE-1. При количестве элементов N больше 500 тыс. следует использовать алгоритмы BS-1, BS-2, BS-3, BE-1, BE-2, BE-3.

7. При жестких временны́х ограничениях, накладываемых на время T_f, следует использовать алгоритмы BS-2 или BE-2.

Таким образом, предложен продукционно-алгоритмический подход к решению задачи импакт-анализа на основе комбинации продукционного и алгоритмического подходов. Реализация подхода базируется на графо-ориентированной модели сети, системе правил и соответствующем механизме вывода, а также применении алгоритмов импакт-анализа на основе схем обхода графов. Интеграция продукционного и алгоритмического подходов позволила повысить гибкость решения и существенно сократить время импакт-анализа.

Кроме того, предложенный подход к решению задачи импакт-анализа позволил при разработке ПИА определить структурно-функциональную схему взаимодействия различных моделей, последовательность действий, построить механизм подключения моделей, уточнить особенности структуры сети и услуг, используемых технологий и другие характеристики, влияющие на импакт-анализ.

Проведенные экспериментальные исследования продемонстрировали работоспособность предложенных моделей и вариантов реализации алгоритмов. Затраты времени и памяти, необходимые для получения нужных операторам результатов, не превосходят реальных ограничений, которые могут не превышать 5 минут на 1,5 млн клиентов.

6.4. ИМПАКТ-АНАЛИЗ В IP-СЕТЯХ С ПРОТОКОЛАМИ RIP И OSPF

Расширение области применения эффективной комбинации продукционного и алгоритмического подходов к решению задачи импакт-анализа на широкий класс сетей доступа, построенных с использованием разных сетевых протоколов и предлагающих различные сервисы, требует разработки соответствующих наборов правил, учитывающих особенности сетей доступа, используемых телекоммуникационных технологий и оборудования. С позиций практического примене-

ния разработка новых моделей и алгоритмов также должна сопровождаться подтверждением их работоспособности и проверкой выполнимости временных ограничений.

Ниже в качестве примера применения предложенного продукционно-алгоритмического подхода к решению задачи импакт-анализа разработаны и реализованы модели и алгоритмы оценки снижения уровня обслуживания клиентов при использовании в IP-сети протоколов маршрутизации RIP и OSPF.

6.4.1. Постановка задачи импакт-анализа в IP-сетях с протоколами RIP и OSPF

Необходимость повышения качества обслуживания пользователей заставляет ОТС непрерывно усовершенствовать методы и средств импакт-анализа [153, 161]. Возникновение неполадок в сети доступа ОТС приводит к снижению качества предоставляемых абонентам телекоммуникационных услуг. В тех случаях, когда происходит полный отказ в предоставлении определенной услуги, можно быстро определить пострадавших абонентов путем чтения списка абонентов, пользующихся этой услугой. Однако в случаях частичного или функционального отказа в сети доступа ОТС определить абонентов, качество обслуживания которых заметно ухудшилось, существенно сложнее.

Причиной этому является то обстоятельство, что предоставление услуги продолжается, но с худшим качеством. Кроме того, механизмы автоматической компенсации неисправностей в виде переключения на резерв могут нивелировать негативное воздействие неисправности частично или полностью. В таких случаях необходимо провести аналитический расчет значений параметров, определяющих качество услуги, предоставляемой абоненту, и сравнить полученные значения с номинальными. Степень расхождения этих значений, а также категория абонента определяют мероприятия, проводимые ОТС по работе с абонентами.

Одной из наиболее распространенных услуг, предоставляемых ОТС, является доступ к сети интернет. Поверх этой услуги реализовано множество сервисов, таких как VoIP, IPTV и др. Одним из важнейших параметров интернет-услуг является скорость доступа. При снижении скорости доступа в результате неполадок услуга предоставляется, но ее качество перестает соответствовать оговоренным требованиям.

Среди сетей доступа к сервисам интернета наиболее распространенными являются IP-сети с протоколами маршрутизации RIP и OSPF [283, 303]. Используя эти протоколы, ОТС недостаточно внимания уделяют задаче импакт-анализа. В работе [153] задача импакт-анализа решается для ОТС, определяющих качество обслуживания клиентов на основе такого традиционного параметра как скорость доступа к сети интернет. Предложенный подход базируется на аналитическом расчете коэффициентов снижения скорости доступа. Но он не позволяет учесть влияние многих неполадок в тех случаях, когда задействуются компенсаторные функции путем переключения на резерв.

Возникает потребность в модификации этого метода. Итак, известна информация о структуре, составных частях и взаимосвязях между элементами IP-сети доступа ОТС, настройках каждого узла. Вся информация о сети доступа, использующей протоколы маршрутизации RIP или OSPF, хранится в инвентарной БД ОТС.

Необходимо разработать метод решения задачи импакт-анализа, позволяющий выявлять абонентов, у которых качество сервиса снизилось в результате неполадок в сетях доступа, и использующий механизмы компенсации. При этом требуется определить степень снижения качества получаемых сетевых услуг.

Решение этой задачи обусловливает необходимость разработать типовые правила и алгоритмы расчетов, учитывающие особенности используемых протоколов маршрутизации, и подтвердить их работоспособность.

6.4.2. Структура подсистемы импакт-анализа в IP-сетях с протоколами RIP и OSPF

Структура ПИА определения влияния неполадок в IP-сетях с протоколами RIP и OSPF на качество обслуживания абонентов ОТС приведена на рис. 6.16.

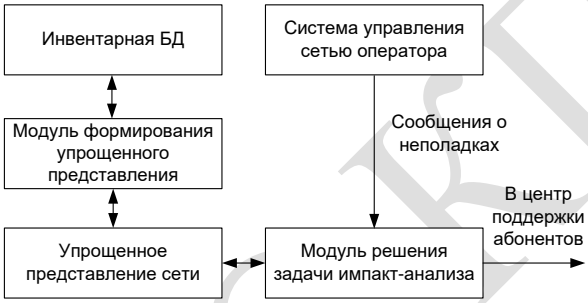


Рис. 6.16. Структура подсистемы импакт-анализа

Инвентарная БД хранит информацию обо всем оборудовании и его настройках, линиях связи, их пропускной способности и скорости передачи данных, предоставляемых сервисах, пользователях, которыми являются частные лица и организации, и другие полезные сведения.

Поиск в инвентарной БД позволяет найти все сведения о структуре и свойствах сети, абонентах, необходимые для решения целого комплекса задач управления. Но в условиях жестких временных ограничений решить задачу импакт-анализа на основе информации из инвентарной базы не представляется возможным.

Поэтому для решения задачи импакт-анализа вместо инвентарной БД используется упрощенное представление, которое отображает только те аспекты сетевой инфраструктуры, которые имеют отношение к решению задачи импакт-анализа [153]. Упрощенное представление сети содержит топологическую схему сети, модель узла, а также систему правил, описывающих особенности взаимодействия узлов сети.

Топологическую схему удобно представить в виде графа, вершины которого соответствуют сетевым узлам, а ветви – линиям связи, соединяющим узлы. Модель узла должна описывать прохождение информационных потоков через сетевой узел и учитывать реакцию сетевого оборудования на возникающие неисправности. Система правил должна учитывать особенности сетевых протоколов, функционирования сетевого узла и реакции узлов на неполадки.

Сообщения о неполадках поступают из СУ сетью ОТС (см. рис. 6.16). Результаты решения задачи импакт-анализа – оценка влияния неполадок в сети до-

ступа – позволяет определить абонентов, которые должны ощутить снижение качества получаемых сетевых сервисов. Перечень абонентов и степени снижения показателей качества сервисов передаются в центр поддержки абонентов ОТС для последующей работы с абонентами.

6.4.3. Модель сети доступа с протоколами RIP и OSPF

Модель сети доступа, используемая для решения задачи импакт-анализа, представляется в виде графа, вершинам которого сопоставляются узлы сети, а ребрам – линии связи [161].

Каждый сетевой узел $U_{i,j}$, $i=\overline{1,I}$, $j=\overline{1,J_i}$, I – количество уровней иерархии узлов, J_i – количество узлов на i-м иерархическом уровне, соединяется с другими узлами входящими и исходящими связями. Для решения задачи импакт-анализа предлагается использовать модель узла, представленную на рис. 6.17.

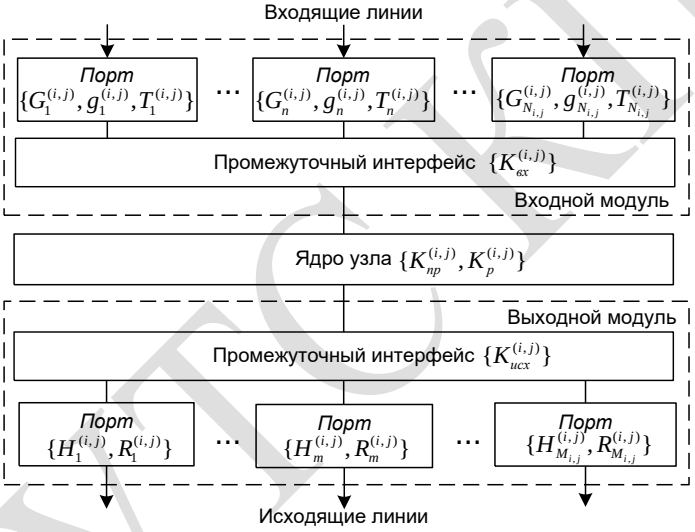


Рис. 6.17. Модель узла сети

Каждый узел сети $U_{i,j}$, $i=\overline{1,I}$, $j=\overline{1,J_i}$ представляется в виде совокупности входного модуля, ядра узла и выходного модуля.

Входной модуль состоит из портов и промежуточного входного интерфейса.

Оценка состояния входной связи проводится посредством нормированного коэффициента снижения скорости доступа на входе порта входного модуля $G_n^{(i,j)}$, $n=\overline{1,N_{i,j}}$, где n – номер порта входного модуля, $N_{i,j}$ – количество портов входного модуля узла $U_{i,j}$, $i=\overline{1,I}$, $j=\overline{1,J_i}$. Коэффициент $G_n^{(i,j)} \in [0,1]$ вычисляется следующим образом:

$$G_n^{(i,j)} = \begin{cases} \dfrac{s_n^{(i,j)}}{v_n^{(i,j)}} \text{ при } v_n^{(i,j)} > 0; \\ 0 \text{ при } v_n^{(i,j)} = 0, \end{cases} \tag{6.42}$$

где $s_n^{(i,j)}$ и $v_n^{(i,j)}$, $n=\overline{1,N_{i,j}}$ – соответственно фактическая и номинальная скорости на входе n-го порта узла $U_{i,j}$.

На вход модели узла (см. рис. 6.17) могут поступать потоки с различающейся скоростью передачи данных. Вклад этих потоков в суммарный выходной поток на выходе промежуточного интерфейса входного модуля будет различным. Также будет различным влияние отказа одного из потоков на общую картину трафика, проходящего через узел. В данном случае задача импакт-анализа рассматривается относительно скорости доступа, поэтому отказ потока с большей номинальной скоростью приведет к большему снижению скорости доступа на выходе промежуточного интерфейса входного модуля. Для учета степени влияния отказа потока введем коэффициент $g_n^{(i,j)}$, который для всех $n=\overline{1,N_{i,j}}$ определяется следующим образом:

$$g_n^{(i,j)} = \begin{cases} \dfrac{v_n^{(i,j)}}{V_{i,j}} \text{ при } V_{i,j} > 0; \\ 0 \text{ при } V_{i,j} = 0, \end{cases} \tag{6.43}$$

где $v_n^{(i,j)}$, $n=\overline{1,N_{i,j}}$ – номинальная скорость n-го порта узла $U_{i,j}$, а $V_{i,j}=\max_n(v_n^{(i,j)})$, $n=\overline{1,N_{i,j}}$ – максимальная номинальная скорость среди всех входных портов узла $U_{i,j}$, $g_n^{(i,j)} \in [0.1]$.

Связи узлов могут быть основными или резервными. Основные связи используются для приема и передачи данных при штатной работе узла. Резервные связи в нормальных условиях не используются, а задействуются только при возникновении неисправностей в сети.

Для учета типа входных связей узла введем вектор $\bar{D}_{i,j}=(d_1^{(i,j)},d_2^{(i,j)},...,d_{N_{i,j}}^{(i,j)})$ размерностью $N_{i,j}$, значения элементов которого определяются следующим образом

$$d_n^{(i,j)} = \begin{cases} 1, \text{ если связь основная или задействована резервная;} \\ 0, \text{ если связь резервная и не задействована,} \end{cases} \tag{6.44}$$

для всех $n=\overline{1,N_{i,j}}$.

Работоспособность порта входного модуля оценивается коэффициентом $T_n^{(i,j)} \in [0,1]$, $n=\overline{1,N_{i,j}}$, значения которого определяются с помощью следующего выражения:

$$T_n^{(i,j)} = \begin{cases} \dfrac{w_n^{(i,j)}}{s_n^{(i,j)}} \text{ при } s_n^{(i,j)} > 0; \\ 0 \text{ при } s_n^{(i,j)} = 0, \end{cases} \tag{6.45}$$

где $s_n^{(i,j)}$ и $w_n^{(i,j)}$, $n=\overline{1,N_{i,j}}$ – соответственно фактическая скорость на входе и выходе n-го порта узла $U_{i,j}$.

АУПСКП

Для оценки скорости доступа входного модуля введем коэффициент $K_{вх}^{(i,j)}$, значения которого определяются следующим образом:

$$K_{вх}^{(i.j)} = \begin{cases} \dfrac{\sum_{n=1}^{N_{i,j}} (G_n^{(i,j)} \cdot T_n^{(i,j)} \cdot g_n^{(i,j)} \cdot d_n^{(i,j)})}{\sum_{n=1}^{N_{i,j}} (g_n^{(i,j)} \cdot d_n^{(i,j)})} \text{ при } \sum_{n=1}^{N_{i,j}} (g_n^{(i,j)} \cdot d_n^{(i,j)}) > 0; \\ 0 \text{ при } \sum_{n=1}^{N_{i,j}} (g_n^{(i,j)} \cdot d_n^{(i,j)}) = 0. \end{cases} \quad (6.46)$$

В ядре узла принимается решение о переключении на резервные линии связи. Введем коэффициент перехода на резерв $K_{пр}^{(i,j)}$, $i = \overline{1,I}$, $j = \overline{1,J_i}$, значение которого определяется настройками узла $U_{i,j}$, хранящимися в инвентарной БД, и учитывается продукционными правилами для конкретных узлов и протоколов. Ядру узла соответствуют шасси и операционная система реального коммутационного оборудования.

Когда выполняется условие $K_{вх}^{(i,j)} \le K_{пр}^{(i,j)}$, ядро переводит резервные связи в активное состояние и входной модуль учитывает это при расчете коэффициента $K_{вх}^{(i,j)}$. При возвращении скорости доступа по основным связям к номинальному значению ядро переводит резервные связи в неактивное состояние, и они перестают учитываться при расчете коэффициента $K_{вх}^{(i,j)}$.

Для оценки работоспособности ядра введем коэффициент $K_p^{(i,j)} \in [0,1]$, $i = \overline{1,I}$, $j = \overline{1,J_i}$, который зависит от состояния узла и определяется следующим образом:

$$K_p^{(i,j)} = \begin{cases} \dfrac{S_{вых}^{(i,j)}}{S_{вх}^{(i,j)}} \text{ при } S_{вх}^{(i,j)} > 0; \\ 0 \text{ при } S_{вх}^{(i,j)} = 0, \end{cases} \quad (6.47)$$

где $S_{вх}^{(i,j)}$ и $S_{вых}^{(i,j)}$ – соответственно скорость доступа на входе и выходе ядра узла $U_{i,j}$.

Выходной модуль состоит из промежуточного интерфейса и портов. Состояние промежуточного интерфейса выходного модуля оценивается коэффициентом $K_{исх}^{(i,j)}$, который рассчитывается с помощью следующего выражения:

$$K_{исх}^{(i,j)} = K_{вх}^{(i,j)} \cdot K_p^{(i,j)}. \quad (6.48)$$

Для оценки работоспособности портов выходного модуля введем коэффициент $R_m^{(i,j)}$, $m = \overline{1,M_{i,j}}$, где $M_{i,j}$ – количество портов выходного модуля узла $U_{i,j}$, $i = \overline{1,I}$, $j = \overline{1,J_i}$. Коэффициент определяется аналогично коэффициенту $T_n^{(i,j)}$.

АУТСКЦ

Оценка состояния исходящей связи осуществляется коэффициентом $H_m^{(i,j)}$, который рассчитывается с помощью выражения

$$H_m^{(i,j)} = K_{исх}^{(i,j)} \cdot R_m^{(i,j)}. \quad (6.49)$$

Посредством описанных выше коэффициентов можно аналитически рассчитать скорость на исходящих линиях узла сети при известных скоростях входящих связей.

6.4.4. Продукционные правила для сетей с протоколами RIP и OSPF

В зависимости от протоколов маршрутизации, используемых в сетях доступа, влияние неполадок на распределение и перераспределение сетевого трафика будет различным. Протоколы маршрутизации типа OSPF задействуют резервные линии связи или строят сложные пути маршрутизации трафика. Протоколы, подобные RIP, не обладают подобными свойствами.

Предложенная модель узла позволяет рассчитывать степень снижения скорости доступа на выходе узла, но не учитывает аспекты взаимодействия узлов и особенности протоколов маршрутизации. Учет взаимодействия узлов при решении задачи импакт-анализа можно реализовать путем использования продукционных правил, описывающих логику взаимодействия узлов сети и свойства самих узлов, необходимые для расчета по упрощенному представлению.

Реализация продукционного подхода в программном коде требует выполнения большого количества строковых сравнений и преобразований, что существенно увеличивает время расчета влияния неисправностей в сети на качество сервисов. Поэтому целесообразно объединить продукционный подход с алгоритмическим, обладающим высоким быстродействием и позволяющим быстро пересчитывать все необходимые коэффициенты узлов сети. Высокая скорость работы программной реализации алгоритмического подхода достигается за счет выполнения только алгебраических операций над коэффициентами.

Продукционный подход использует систему правил, запись которых в нотации Бекуса – Наура имеют вид (6.29) [161].

Для узлов $U_{i,j}$, $i = \overline{1,I}$, $j = \overline{1,J_i}$, использующих протокол маршрутизации RIP, необходимо использовать следующие правила:

IF technology.parent.port.node = RIP $\wedge$ class.parent.port.node = Router $\wedge$ technology.node = RIP $\wedge$ class.node = Router THEN IsActive.port.node = 1;

IF technology.parent.port.node $\neq$ RIP $\wedge$ class.parent.port.node = Router $\wedge$ technology.node = RIP $\wedge$ class.node = Router THEN g.port.node = 0,

где node – узел сети, для которого применяется продукционное правило (далее – текущий узел), port.node – входной порт текущего узла, technology.node – протокол маршрутизации текущего узла, parent.port.node – узел, который соединен с текущим узлом через порт port.node (далее – родительский узел), technology.parent.port.node – протокол маршрутизации родительского узла, IsActive.port.node определяет, является ли связь порта основной, g.port.node – коэффициент $g_n^{(i,j)}$ порта.

Если используется маршрутизация RIP, а узлы $U_{i,j}$, $i = \overline{1,I}$, $j = \overline{1,J_i}$ являются маршрутизаторами, то нужно указать, что входная связь является основной. Если

родительский узел не использует протокол RIP, то с помощью изменения коэффициента $g_n^{(i,j)}$ поток порта перестает использоваться в расчете $K_{вх}^{(i.j)}$.

Для узлов $U_{i,j}$, $i=\overline{1,I}$, $j=\overline{1,J_i}$, использующих протокол OSPF, необходимо применять следующие правила:

IF technology.parent.port.node = OSPF $\wedge$ class.parent.port.node = Router $\wedge$ technology.node = OSPF $\wedge$ class.node = Router THEN IsActive.port.node = 1;

IF technology.parent.port.node $\neq$ OSPF $\wedge$ class.parent.port.node = Router $\wedge$ technology.node = OSPF $\wedge$ class.node = Router THEN g.port.node = 0;

IF UseAsReserve.port.parent.port.node = 1 $\wedge$ technology.parent.port.node = OSPF $\wedge$ class.parent.port.node = Router $\wedge$ technology.node = OSPF $\wedge$ class.node = Router THEN IsReserve.port.node = 1;

IF UseAsReserve.port.parent.port.node = 1 $\wedge$ technology.parent.port.node = OSPF $\wedge$ class.parent.port.node = Router $\wedge$ technology.node = OSPF $\wedge$ class.node = Router THEN IsActive.port.node = 0,

где UseAsReserve.port.parent.port.node определяет, используется ли связь как резервная в текущей конфигурации сети.

Для учета общих особенностей взаимодействия всех узлов сети необходимо описать общие продукционные правила, учитывающие общесетевые аспекты и аспекты взаимодействия с внешними сетями. Например, необходимо отдельно описать правило для узла верхнего уровня, скорость доступа которого зависит не от неполадок в сети провайдера, а от неполадок в магистральных сетях.

Такими правилами могут быть, например:

IF MainNode.node = 1 $\wedge$ InPortCount.node = 0 THEN kIn.node = InputValue.MainNode.node,

где MainNode.node – признак того, что скорость доступа узла зависит от внешних сетей, InPortCount.node – количество портов входного модуля узла, kIn.node – коэффициент $K_{вх}^{(i.j)}$ узла, InputValue.MainNode.node – отношение реальной скорости доступа к заявленной.

6.4.5. Алгоритмы решения задачи импакт-анализа в сетях с протоколами RIP и OSPF

Алгоритм расчета упрощенного представления сети доступа BS-3. Упрощенное представление строится на основании информации из инвентарной БД и продукционных правил. При возникновении неисправностей необходимо быстро пересчитать коэффициенты узлов с целью определения влияния неисправностей на скорость доступа абонентов. Расчет коэффициентов происходит с помощью модифицированного алгоритма расчета по упрощенному представлению [161]. Алгоритм заключается в последовательном обходе графа в глубину и вычислении коэффициентов для каждого узла сети. Отличительной особенностью предлагаемого алгоритма является дополнительная логика при повторном расчете коэффициентов узла и исключение узлов из расчета. Когда коэффициенты узла невозможно рассчитать по причине, например, циклических связей в модели, узел добавляется в список узлов, требующих повторного расчета. Если узел нельзя рассчитать несколько раз, он добавляется в список узлов, коэффициенты которых рассчитать невозможно, а информация о таких узлах поступает к

администратору ПИА. Для повышения скорости расчета при возникновении неисправности обход графа начинается с вершины, в которой возникла неисправность. Определяются все зависимые узлы, для которых проводится перерасчет коэффициентов. После перерасчета коэффициентов в узлах нижнего уровня (уровня абонентов) можно определить степень снижения качества предоставляемого сервиса путем определения фактической скорости доступа.

Работу алгоритма BS-3 при первоначальном расчете, когда необходимо рассчитать коэффициенты для всех узлов в сети, можно описать следующим образом:

Шаг 1. Начиная с верхнего иерархического уровня, для всех узлов рассчитать коэффициенты $K_{вх}^{(i,j)}$ и $K_{исх}^{(i,j)}$.

Шаг 2. Если коэффициенты узла нельзя пересчитать, то добавить узел в список узлов, требующих повторного расчета.

Шаг 3. Если расчет коэффициентов для всех узлов закончен, то пересчитать коэффициенты узлов из списка узлов, требующих повторного расчета.

Шаг 4. Если коэффициенты узла нельзя рассчитать несколько раз (количество попыток задается администратором системы импакт-анализа), то добавить узел в список узлов, коэффициенты которых рассчитать невозможно.

В случае неполадки в узле работу алгоритма можно описать в таком виде:

Шаг 1. Пересчитать коэффициенты проблемного узла.

Шаг 2. Пересчитать коэффициенты узлов, имеющих входную связь с проблемным узлом.

Шаг 3. В случае снижения скорости доступа для одного из узлов – считать его проблемным и повторить алгоритм, начиная с шага 1.

Алгоритм решения задачи импакт-анализа BE-3. Предлагается следующий алгоритм BE-3 оценки влияния неисправностей на качество сервиса в сетях доступа с протоколами RIP и OSPF.

Шаг 1. Построить граф модели. На основании данных инвентарной БД ОТС строится граф модели сети.

Шаг 2. Применить продукционные правила к графу с целью получения упрощенного представления. Для учета логики взаимодействия узлов сети к узлам графа применяются продукционные правила, состоящие из условия и следствия. Если условие выполняется, то применяется следствие правила. Продукционный подход позволяет отделить алгоритм расчета коэффициентов узла от правил, по которым взаимодействуют узлы в сети; описать логику взаимодействия узлов в сети в виде продукционных правил. Недостаток продукционного подхода – медленное выполнение преобразований при применении правила к узлу.

Шаг 3. Выполнить расчет коэффициентов по упрощенному представлению. После применения продукционных правил выполняются расчеты по упрощенному представлению с использованием алгоритмического подхода, позволяющего быстро проводить все преобразования за счет выполнения только математических операций над коэффициентами. Расчеты по упрощенному представлению сводятся к расчету коэффициентов узлов в сети. Сначала рассчитываются коэффициенты всех узлов модели. При возникновении неполадок перерасчет выполняется только для узлов, на которые повлияла неполадка.

Шаг 4. Определить степень влияния неполадки на качество предоставляемых сервисов путем сравнения полученных значений коэффициентов с номинальными значениями. Информация о степени снижения значений показателей качества сервиса передается в центр поддержки абонентов провайдера.

Тестовые примеры модели сети и результаты моделирования. Подтверждение работоспособности предлагаемого метода получено на тестовых примерах модели сети.

1. Модель без резервных связей.

Все узлы являются маршрутизаторами, использующими протокол RIP. Резервные линии связи не используются (рис. 6.18).

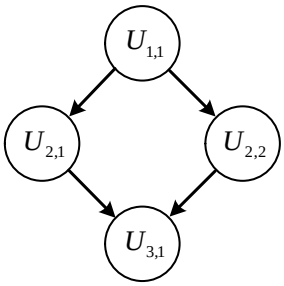


Рис. 6.18. Модель без резервных связей

На основании модели узлов и продукционных правил опишем модель следующим образом. В начальный момент коэффициенты узлов в сети имеют такие значения:

$$\begin{aligned} &K_p^{(1,1)} = K_p^{(2,1)} = K_p^{(2,2)} = K_p^{(3,1)} = 1, \\ &T_1^{(2,1)} = T_1^{(2,2)} = T_1^{(3,1)} = T_2^{(3,1)} = 1, \\ &g_1^{(2,1)} = g_1^{(2,2)} = g_1^{(3,1)} = g_2^{(3,1)} = 1, \\ &R_1^{(1,1)} = R_2^{(1,1)} = R_1^{(2,1)} = R_1^{(2,2)} = 1. \end{aligned} \tag{6.50}$$

Все связи являются основными $d_n^{(i,j)} = 1$, $i = \overline{1, I}$, $j = \overline{1, J_i}$, $n = \overline{1, N_{i,j}}$.

Значения параметров узла $U_{1,1}$: MainNode.node = 1, InPortCount.node = 0, InputValue.MainNode.node = 1.

Применив общие правила и правила для протокола RIP, задав $K_{вх}^{(1,1)} = 1$ для сети без неполадок, получим

$$K_{вх}^{(3,1)} = \frac{G_1^{(3,1)} \cdot T_1^{(3,1)} \cdot g_1^{(3,1)} \cdot d_1^{(3,1)}}{g_1^{(3,1)} \cdot d_1^{(3,1)} + g_2^{(3,1)} \cdot d_2^{(3,1)}} + \frac{G_2^{(3,1)} \cdot T_2^{(3,1)} \cdot g_2^{(3,1)} \cdot d_2^{(3,1)}}{g_1^{(3,1)} \cdot d_1^{(3,1)} + g_2^{(3,1)} \cdot d_2^{(3,1)}} = 1. \tag{6.51}$$

Таким образом, если в сети нет неполадок, то скорость доступа на выходе узла $U_{3,1}$ на основании (6.50) и (6.51) соответствует заявленной.

В случае отказа узла $U_{2,1}$ коэффициент $K_p^{(2,1)} = 0$, а $K_{вх}^{(3,1)} = 0{,}5$ и происходит очевидное снижение скорости в два раза.

2. Модель с одной резервной связью.

Все узлы являются маршрутизаторами, направляющими трафик по нескольким линиям. Резервная линия связи обозначена R (рис. 6.19).

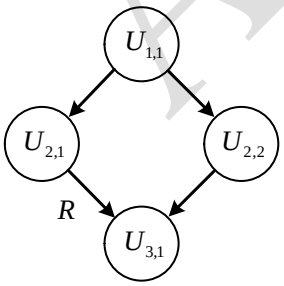


Рис. 6.19. Модель с одной резервной связью

При отказе узла $U_{2,2}$ или при обрыве связи между узлами $U_{3,1}$ и $U_{2,2}$ задействуется резервная связь, соединяющая узлы $U_{3,1}$ и $U_{2,1}$. В случае снижения скорости передачи от узла $U_{2,2}$ на узел $U_{3,1}$ резервная связь задействоваться не будет, т. е. считается, что коэффициент перехода на резерв $K_{пр}^{(3,1)} = 0$.

На основании модели узлов и продукционных правил при отсутствии неисправностей коэффициент $K_{вх}^{(3,1)} = 1$.

При отказе узла $U_{2,2}$ скорость на выходе узла $U_{3,1}$ упадет до нуля, но тогда будет выполнено условие

АИС КШ

$K_{вх}^{(3,1)} \le K_{пр}^{(3,1)}$ и, в соответствии с продукционным правилом для OSPF, $d_1^{(3,1)} = 1$ и за счет маршрутизации

$$K_{вх}^{(3,1)} = \frac{G_1^{(3,1)} \cdot T_1^{(3,1)} \cdot g_1^{(3,1)} \cdot d_1^{(3,1)}}{g_1^{(3,1)} \cdot d_1^{(3,1)} + g_2^{(3,1)} \cdot d_2^{(3,1)}} + \frac{G_2^{(3,1)} \cdot T_2^{(3,1)} \cdot g_2^{(3,1)} \cdot d_2^{(3,1)}}{g_1^{(3,1)} \cdot d_1^{(3,1)} + g_2^{(3,1)} \cdot d_2^{(3,1)}} = 1. \qquad (6.52)$$

Таким образом, после перехода на резервную линию скорость доступа узла $U_{3,1}$ восстановилась до нормативного значения.

3. Модель фрагмента сети доступа ОТС.

Приведем фрагмент сети ОТС с двумя корневыми узлами $U_{1,1}$ и $U_{1,2}$ (рис. 6.20).

При работоспособной сети

$$K_p^{(i,j)} = T_n^{(i,j)} = g_n^{(i,j)} = R_n^{(i,j)} = 1, \qquad (6.53)$$

для всех $i = \overline{1,3}, j = \overline{1,6}$.

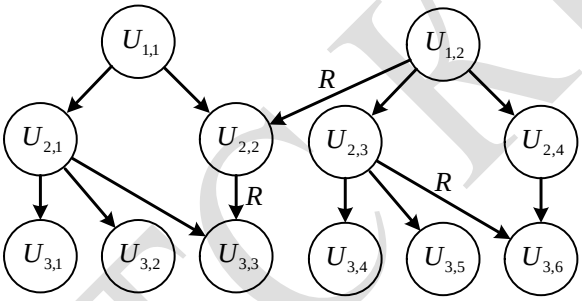


Рис. 6.20. Модель фрагмента сети ОТС

При отказе узла $U_{1,1}$ скорость доступа на узлах $U_{3,1}$ и $U_{3,2}$ упадет до нуля.

При $K_{вх}^{(2,2)} \le K_{пр}^{(2,2)}$ в соответствии с продукционным правилом для OSPF $d_2^{(3,3)} = 1$ и за счет маршрутизации

$$K_{вх}^{(3,3)} = \frac{G_1^{(3,3)} \times T_1^{(3,3)} \times g_1^{(3,3)} \times d_1^{(3,3)}}{g_1^{(3,3)} \times d_1^{(3,3)} + g_2^{(3,3)} \times d_2^{(3,3)}} + \frac{G_2^{(3,3)} \times T_2^{(3,3)} \times g_2^{(3,3)} \times d_2^{(3,3)}}{g_1^{(3,3)} \times d_1^{(3,3)} + g_2^{(3,3)} \times d_2^{(3,3)}} = 1, \qquad (6.54)$$

скорость доступа на узле $U_{3,3}$ за счет маршрутизаторов не изменится.

Результаты моделирования. Моделирование проводилось на сервере с ОС Windows Server 2008 R2 Enterprise, процессором Intel Xeon X5675 (частота 3 ГГц), 96 Гб ОЗУ, режим работы 64-бит, один поток. Граф тестовой модели состоял из десяти иерархических уровней, выходной модуль каждого из узлов содержал пять портов. Результаты моделирования усреднялись по тридцати экспериментам. Для демонстрации преимуществ предлагаемого метода проводились дополнительные исследования с использованием только продукционного подхода, когда задача импакт-анализа решалась только применением правил.

Результаты моделирования BS-3 и BE-3 приведены в табл. 6.1.

АУЦКП

Таблица 6.1. Результаты моделирования

	Предлагаемый подход		Продукционный подход	
Кол-во узлов, млн штук	Среднее время построения упрощенного представления, с	Среднее время импакт-анализа, с	Среднее время построения упрощенного представления, с	Среднее время импакт-анализа, с
0,125	5,46	0,9	7,05	4,5
0,25	10,78	1,5	13,8	9,75
0,5	15,96	3,2	23,4	18
1	29,4	6,4	38,85	28
2	81,2	13,5	105,6	63,5
4	143,92	30	181,8	101
8	313,6	49,5	418,5	275
16	634,76	73,5	793,2	493
32	1418,9	156	1805,25	1135,7

Результаты моделирования BS-3 и BE-3 приведены на графике рис. 6.21, где изображена зависимость среднего времени построения модели (кривая 1 – продукционным методом; кривая 2 – BS-3) и среднего времени импакт-анализа (кривая 3 – продукционным методом; кривая 4 – BE-3) от количества элементов сети.

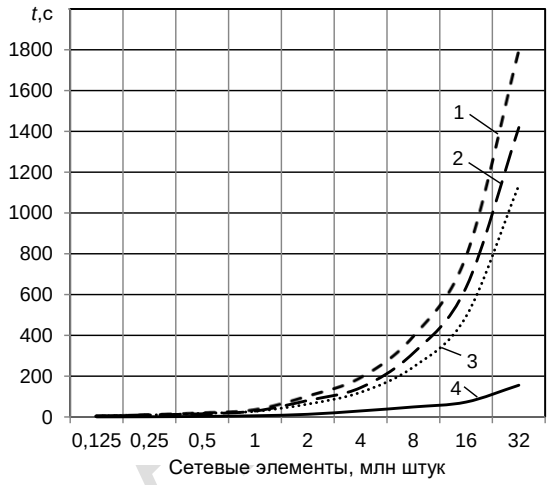


Рис. 6.21. Результаты моделирования: 1 – среднее время построения упрощенного представления продукционным методом; 2 – среднее время построения упрощенного представления предлагаемым методом; 3 – среднее время выполнения анализа продукционным методом; 4 – среднее время выполнения анализа предлагаемым методом

Анализ результатов эксперимента позволяет сделать вывод, что метод, использующий интеграцию продукционного и алгоритмического подходов, значительно быстрее выполняет импакт-анализ, чем продукционный подход. Причем выигрыш тем больше, чем больше узлов в сети.

Таким образом, продукционно-алгоритмический подход позволяет объединить гибкость продукционных правил и высокую скорость арифметического расчета по модели сети. Результаты проведенных исследований подтвердили высокое быстродействие, благодаря чему время решения задачи импакт-анализа для сети с количеством оборудования 4 млн единиц составило не более 30 с.

ВЫВОДЫ К ГЛАВЕ 6

1. Для обнаружения неисправностей в ИТ-инфраструктуре разработан подход к анализу элементов функциональной группы на основе глубинно-упорядоченных регионов. Представление элементов ИТ-инфраструктуры в евклидовом пространстве позволяет, с одной стороны, использовать глубинное упорядочивание точек, соответствующих этим элементам в параметрическом пространстве, для определения нормы и оценки отклонений состояний от нормы, с другой стороны, применять для этого эффективные методы классификации.

2. Для реализации предложенного подхода разработан комплекс моделей, методов и инструментов определения состояния и анализа динамики изменения свойств ОМУ методами статистической геометрии с выделением глубинно-упорядоченных регионов. Это позволяет осуществлять обобщение оценки состояния до уровня подсистем и ИТ-инфраструктуры в целом после преобразования параметрических пространств с учетом иерархии.

3. Исследованы преимущества и недостатки алгебраического, продукционного и продукционно-алгоритмического подходов к решению задачи импакт-анализа и определены области их применения.

4. Разработан продукционно-алгоритмический подход к решению задачи оценки влияния неисправностей на качество предоставляемых услуг. Предложенные графо-ориентированная модель представления сети и инфокоммуникационных сервисов, система продукций и соответствующий механизм вывода, алгоритмы анализа влияния неисправностей на основе схем обхода графов, позволили сочетать гибкость систем продукций и эффективность алгоритмов. По существу алгоритм оценки влияния неисправностей предварительно выводится с помощью системы продукций, а потом быстро выполняется по мере необходимости.

5. Выполнена реализация предложенного подхода к решению задачи оценки влияния неисправностей, которая отличается большой гибкостью, позволяет учесть наличие резервных линий, изменяющиеся политики управления ТКС, динамику сетей, оборудования и технологий, а также имеет высокое быстродействие, что позволяет получать нужные результаты в рамках жестких временных ограничений.

6. Предложен общий подход к импакт-анализу, позволяющий определить структурно-функциональную схему взаимодействия различных моделей импакт-анализа. Подход, основанный на правилах распространения влияния неисправностей, учитывающих особенности структуры сети и услуг, используемые технологии и другие важные факторы, позволяет разработчикам ПИА определять последовательность действий, строить механизм подключения моделей, уточнять характеристики, влияющие на распространение неблагоприятных последствий неисправностей на качество предоставления услуг пользователям.

АУТСКИЙ

ГЛАВА 7

УПРАВЛЕНИЕ РЕСУРСАМИ И НАГРУЗКОЙ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРЫ

Повышения качества предоставления услуг в корпоративной ИТ-инфраструктуре можно достичь за счет выделения дополнительных ресурсов приложениям, предоставляющим услуги, или обеспечения приоритетного прохождения трафика, генерируемого критичными приложениями. В первом случае возникают задачи управления нагрузкой и ресурсами. Эти задачи делятся на два больших класса – управление ресурсами в условиях их дефицита, что характерно для большинства ИТ-инфраструктур корпоративного уровня с распределенными ресурсами, и рационального использования ресурсов в условиях их избытка, что характерно для ИТ-инфраструктур на основе ЦОД. Во втором случае возникают задачи теории расписаний, определения приоритетов и диспетчирования. Глава посвящена разработке моделей и методов рационального использования ресурсов ИТ-инфраструктуры в условиях дефицита ресурсов. Задачи управления ресурсами ЦОД рассматриваются в следующей главе.

7.1. ПРОЦЕССНО-РЕСУРСНЫЙ ПОДХОД К УПРАВЛЕНИЮ РАСПРЕДЕЛЕНИЕМ РЕСУРСОВ

7.1.1. Актуальность проблемы управления ресурсами в условиях их дефицита

В процессе функционирования ИУС возникают ситуации, когда ресурсов становится недостаточно и бизнес-процессы, лишенные или сильно ограниченные поддержкой со стороны ИТ-инфраструктуры, начинают работать неэффективно. К дефициту ресурсов приводят неисправности в ИТ-инфраструктуре, переполнение хранилищ данных, увеличение количества пользователей, увеличение объемов передаваемых и обрабатываемых данных, внедрение новых ИТ-сервисов без привлечения дополнительных ресурсов.

Несмотря на наличие дефицита ресурсов, ИТ УИТИ должна эффективно поддерживать систему бизнес-процессов за счет априорной поддержки наиболее важных бизнес-процессов и способности перераспределять ресурсы.

Для придания ИТ УИТИ такой способности необходимо решить задачи, среди которых по важности можно выделить следующие:

- управления доступом к ограниченным ресурсам с учетом важности бизнес-процессов;
- распределения ограниченных ресурсов при условии обеспечения необходимого уровня надежности функционирования бизнес-процессов;
- управления перераспределением ресурсов при изменении значимости бизнес-процессов и др.

Многообразие задач управления распределением ограниченных ресурсов порождено множеством условий и исходных данных, систематизированных на рис. 7.1. Отметим, что эта систематизация справедлива и для управления ИТ-инфраструктурой в условиях избытка ресурсов.

Рис. 7.1. Систематизация факторов, определяющих многообразие постановок задач управления ресурсами ИТ-инфраструктуры

Проблематика управления доступом к ресурсам ИТ-инфраструктуры рассматривается во многих работах, в частности [33, 54, 58, 173, 174, 233, 249, 251, 322]. К сожалению, они не учитывают все систематизированные выше факторы, в частности, виды ресурсов ИТ-инфраструктуры и ограничения. Поэтому необходимо разработать и общую схему решения задач этого класса, и комплекс моделей и методов, которые могут быть положены в основу компонентов ИТ УИТИ, непосредственно управляющих ресурсами ИТ-инфраструктуры, их выделением, учетом, планированием и оптимизацией в условиях их ограниченности.

7.1.2. Управление распределением ресурсов в условиях дефицита

Постановка проблемы. В условиях широкого распространения процессного управления ИТ УИТИ целесообразно рассматривать как основу платформы управления ИТ-инфраструктурой, которая обеспечивает эффективную реализацию системы бизнес-процессов предприятий. Управление бизнес-процессами целесообразно осуществлять посредством распределения ресурсов. Поэтому в основу ИТ УИТИ положен ресурсный подход. Таким образом, управление ИТ-инфраструктурой предопределяет необходимость решения задач мониторинга, анализа и управления ОМУ, а также управления доступом распределенных приложений к общим вычислительным и коммуникационным ресурсам и ресурсам хранения информации [114, 148, 156].

Учитывая ограниченность ИТ-ресурсов в условиях интеграции процессного и ресурсного подходов, необходимо выстроить рациональную схему планирования, управления и перераспределения ресурсов в соответствии с требованиями системы бизнес-процессов. Реализация этой схемы в ИТ УИТИ позволит суще-

ственно снизить расходы на содержание ИТ-инфраструктуры, обеспечив поддержку системы бизнес-процессов с использованием меньшего количества ресурсов, быстро восстанавливать работоспособность ИУС и перераспределять ресурсы в случае возникновения неисправностей, отдавая их наиболее значимым бизнес-процессам.

Более рационального использования ресурсов ИТ-инфраструктуры можно достичь за счет интеграции ресурсов различных ИТ, уменьшения объемов ресурсов, предназначенных для резервирования, использования одних и тех же ресурсов различными приложениями и процессами, перераспределения ограниченных ресурсов.

Управление доступом к общим ресурсам осуществляет ИТ УИТИ, которая в случае возникновения нештатных ситуаций обеспечивает перераспределение ресурсов и закрепление их за бизнес-процессами или приложениями с учетом их значимости. Для эффективной работы ИТ УИТИ необходимо разработать математические модели и алгоритмы распределения и перераспределения ресурсов ИТ-инфраструктуры, провести их исследование и практическую реализацию, убедиться в их работоспособности, полезности и эффективности.

Одна из важнейших функций ИТ-подразделения – планирование ресурсов. В частности, необходимо определить, какие новые ИТ должны быть приобретены и какие новые информационные, коммуникационные и другие ресурсы закуплены. При этом необходимо определить, закупать ли новые ресурсы, или новые ИТ можно развернуть на имеющихся ресурсах, перераспределив их соответствующим образом. Для решения этой задачи ИТ-подразделение должно иметь в СУИ специальное ПО, использующее соответствующие математические модели и методы.

Ведущие поставщики СУИ не раскрывают внутренних механизмов работы своего ПО, а библиотеки лучших практик ИТ-управления ITIL и решения ITSM содержат только формализованное описание организации работы ИТ-подразделения и управления ИТ-услугами. В тоже время для эффективной работы СУИ необходимо иметь множество разнообразных математических моделей и алгоритмов, предназначенных для решения различных задач управления доступом к ограниченным ресурсам ИУС.

Хотя, как подчеркивалось выше, проблематика управления доступом к ресурсам рассматривается во многих работах, полнофункциональных решений пока нет. Предложенные модели и методы распределения ресурсов либо ориентированы на распределенные системы других классов, либо не учитывают комплексный характер ИУС. Так, ограничен выбор универсальных моделей распределения ресурсов для больших распределенных ИТ-инфраструктур корпоративного уровня.

В работе [148] на основе двухуровневой модели предложена многоуровневая модель распределения ресурсов и способ перехода от многоуровневой модели к двухуровневой. Предложенная в работе [114] модель управления перераспределением ресурсов ИТ-инфраструктуры оперативно реагирует на изменение значимости бизнес-процессов, позволяет осуществлять эффективное распределение и перераспределение общих ресурсов при возникновении новых задач. Предложенный алгоритм выделения ресурсов учитывает изменяющуюся значимость задач и может быть использован в иерархических СУИ. Однако не рассмотрены вопросы практической реализации моделей в СУИ, не подтверждена пригодность разработок с точки зрения практики.

В работах [144, 149] приведены модели распределения ресурсов в ИТ-инфраструктуре специального назначения. Работа таких систем ограничивается соответствующими требованиями и регламентами, не позволяющими СУИ автоматически перераспределять ресурсы ИТ-инфраструктуры. Здесь функции СУИ ограничиваются выдачей рекомендаций дежурным и администраторам для принятия решений, а также осуществлением технологического управления оборудованием и ПО ФПС и ТПС.

Существует класс моделей управления ресурсами Grid, кластерных систем и сетей связи [33, 54, 58, 233, 249, 251, 322], но они не учитывают все виды ресурсов ИТ-инфраструктуры и оперативные ограничения. Управления сетями связи хорошо структурированы. Для них разработаны эффективные модели и методы управления потоками, нагрузками на входе и другими параметрами. Однако они не учитывают возможность балансирования нагрузки на вычислительные ресурсы, влияния размещения информационных и программных ресурсов и т. п. Кроме того, модели управления ресурсами Grid и кластерных систем в большинстве фокусируются только на вычислительных, информационных и программных ресурсах.

Поэтому необходимо разработать новые модели и методы, лишенные отмеченных недостатков, на их основе реализовать компоненты ИТ УИТИ, непосредственно управляющие ресурсами ИТ-инфраструктуры, их выделением, учетом, планированием и оптимизацией. Целью данной главы является разработка технологии управления ИТ-инфраструктурой предприятия в условиях ограниченности ресурсов.

Обоснование ресурсного подхода при управлении ИТ-инфраструктурой в условиях недостаточности ресурсов. В процессе функционирования ИТ-инфраструктуры предприятия, даже в том случае, когда ИТ-ресурсы на этапе проектирования выбраны с учетом резервирования, часто возникают ситуации, когда ресурсов становится недостаточно для эффективной работы бизнес-процессов или процессов деятельности. Причиной этому могут быть отказы средств коммутации, выход из строя серверов и других элементов ИТ-инфраструктуры, переполнение хранилищ данных, увеличение количества пользователей, внедрение новых ИТ и развертывание новых приложений без привлечения дополнительных ресурсов и др. Недостаток ресурсов наблюдается и тогда, когда с целью экономии на этапе проектирования и развертывания ИТ-инфраструктуры создаются ресурсные ограничения, т. е. ИТ-ресурсы приобретаются с априорной недостаточностью в предположении того, что не все приложения будут работать одновременно или время работы приложений определяется графиком или периодами суток. Например, резервное копирование планируется проводить ночью, когда уменьшается загруженность коммуникационной сети, а бухгалтерии большие объемы ресурсов будут выделяться только во время подготовки отчетов. При дефиците ресурсов бизнес-процессы лишаются или сильно ограничиваются поддержкой со стороны ИТ-инфраструктуры, менеджмент предприятия своевременно не получает всей необходимой информации и снижается эффективность управленческих решений. Часто ситуацию не может спасти и дублирование наиболее важных ресурсов или резервирование для низкоприоритетных задач. Действительно, дублирование и резервирование могут быть неэффективными в условиях сложной зависимости одних ресурсов от других.

Именно поэтому в ИТ УИТИ целесообразно использовать ресурсный подход к управлению ИТ-инфраструктурой, предоставляя первоочередное право доступа к ресурсам приложениям, поддерживающим наиболее важные бизнес-процессы, и ограничивая выполнение низкоприоритетных бизнес-задач. Для этого СУИ должна поддерживать режимы работы в условиях ресурсных ограничений. Теоретически возможно описать все возможные комбинации ограничений ресурсов и правила управления для каждой из этих комбинаций, однако практическая реализации ИТ УИТИ в этом случае будет громоздкой и не гибкой, поскольку количество комбинаций может быть огромным, а модернизация ИТ-инфраструктуры потребует полной переработки логики работы ИТ УИТИ. Поэтому работа ИТ УИТИ должна базироваться на совокупности математических моделей, учитывающих важность бизнес-процессов, претендующих на использование общих ограниченных ИТ-ресурсов, схеме выбора и взаимодействия моделей с учетом текущей ситуации в ИТ-инфраструктуре, методах управления доступом к ресурсам в условиях их ограниченности.

Иерархия задач управления ИТ-инфраструктурой при дефиците ресурсов. Поскольку система управления доступом к ресурсам должна учитывать особенности работы с ресурсами вычислительных систем, СУБД, систем хранения данных и других элементов ИТ-инфраструктуры, то в условиях реализации процессного и ресурсного принципов управления она должна строиться с учетом, как минимум, такой трехуровневой схеме процессов управления: выбор самых важных бизнес-процессов при условии их обслуживания в сложившейся ситуации; закрепление конкретных ресурсов за выбранными бизнес-процессами; диспетчерское управление предоставлением услуг.

Задача первого уровня состоит заключается в таком закреплении интегрированного ресурса за бизнес-процессами, при котором общая важность обслуживания бизнес-процессов будет максимальной при соответствующий ресурсных ограничениях. Можно говорить о закреплении интегрированного ресурса за теми бизнес-процессами, которые имеют наивысшую суммарную важность и могут быть эффективно поддержаны имеющимися ресурсами. При этом некоторые процессы могут быть переведены на ограниченное функционирование.

Задача управления ограниченными ресурсами второго уровня формулируется как задача закрепления за выбранными на первом уровне бизнес-процессами конкретных ресурсов ИТ-инфраструктуры, когда интегрированный критерий эффективности закрепления ресурсов получит максимальное значение при условии выполнения временны́х, технологических, ресурсных и других ограничений.

Диспетчерское управление заключается в составлении расписаний выполнения запросов бизнес-процессов на ресурсы ИТ-инфраструктуры с максимально эффективным использованием ресурсов при условии выполнения временны́х, технологических, ресурсных и других ограничений на выбранные бизнес-процессы и реализации этих расписаний.

К этим уровням стоит прибавить возможность закладывания параметров эффективного управления доступом на этапе проектирования ИТ-инфраструктуры и оптимизации наращивания ресурсов при ее модернизации и развитии.

Процессно-ресурсный подход подразумевает такое управление распределением ресурсов, чтобы обеспечивалась эффективная реализация системы бизнес-процессов. В этом случае ИТ УИТИ осуществляет управление доступом приложении $A = \{A_l\}$, $l = \overline{1,I}$ к ресурсам, предоставляя первоочередное право доступа

тем приложениям, которые поддерживают наиболее важные бизнес-процессы, и ограничивая выполнение низкоприоритетных бизнес-задач.

При процессно-ресурсном подходе решаемые задачи можно распределить по следующим шести иерархическим уровням, непосредственно связанным с жизненным циклом управления ИТ-инфраструктурой:

1) определение необходимых ресурсов для поддержания наиболее важных бизнес-процессов в условиях ограниченного финансирования на этапе планирования и проектирования ИТ-инфраструктуры;

2) определение значимости бизнес-процессов, поддерживаемых с помощью ИТ в сложившейся ситуации недостаточности ресурсов;

3) выбор наиболее важных бизнес-процессов, которые могут быть обслужены в ИТ-инфраструктуре в сложившихся условиях ограниченности ресурсов;

4) распределение или перераспределение конкретных ресурсов ИТ-инфраструктуры для поддержания бизнес-процессов, выбранных на предыдущем уровне иерархии задач;

5) реализация управления доступом бизнес-процессов к ограниченным ресурсам ИТ-инфраструктуры (диспетчерское управление предоставлением услуг);

6) оптимизация наращивания ресурсов при модернизации и развитии ИТ-инфраструктуры.

Модели и методы решения задач первого уровня. На этапе планирования и проектирования ИУС при отсутствии финансовых ограничений разрабатывается ИТ-инфраструктура, ресурсы которой будут достаточными для поддержания бизнес-процессов, удовлетворяя требованиям к производительности, объемом хранилищ данных и другим параметрам. Кроме того, наиболее важные ИТ-ресурсы дублируются или резервируются.

Если бюджет на создание ИТ-инфраструктуры ограничен, что характерно для большинства предприятий, то уже на стадии проектирования в ИТ-инфраструктуру закладывается дефицит ИТ-ресурсов, и полученная в итоге ИТ-инфраструктура не будет идеальной. Несмотря на это ИУС, спроектированная с дефицитом ресурсов, способна эффективно поддерживать бизнес-процессы, если априорно ИТ УИТИ обеспечивает поддержку наиболее важных бизнес-процессов, а СУИ может перераспределять ресурсы в процессе работы и имеется регламент использования ресурсов.

Задачу определения необходимых ресурсов для поддержания наиболее важных бизнес-процессов на этапе планирования ИТ-инфраструктуры при дефиците ресурсов сформулируем следующим образом.

Для заданного ограниченного бюджета и известной стоимости отдельных ресурсов необходимо найти такую конфигурацию распределения ресурсов, которая бы обеспечивала поддержание наиболее важных бизнес-процессов.

Введем обозначения:

$\hat{Z} = \{Z_i\}$, $i = \overline{1,n}$ – множество бизнес-процессов, поддержание которых обеспечивает эффективное функционирование предприятия как объекта управления;

$\hat{w} = \{w_i\}$, $i = \overline{1,n}$ – коэффициенты важности процессов $Z_1,...,Z_n$ соответственно;

$R = \{R_j\}$, $j = \overline{1,m}$ – ресурсы ИТ-инфраструктуры, необходимые для поддержания бизнес-процессов $\hat{Z}$;

$P = \left\| p_{ij} \right\|$ – матрица потребностей процессов $\hat{Z}$ в ресурсах, ее элемент p_{ij} равен количеству необходимого процессу Z_i ресурса R_j или 0, если ресурс не требуется;

$\hat{r} = \{r_j\}$, $j = \overline{1,m}$ – количество ресурсов $R_1,...,R_m$ соответственно;

$c = \{c_j\}$, $j = \overline{1,m}$ – стоимость единицы ресурса $R_1,...,R_m$ соответственно;

C – бюджет на создание ИТ-инфраструктуры.

Сначала рассмотрим дискретный случай, когда при планировании предусматривается, что бизнес-процесс или поддерживается в полном объеме, или полностью лишается поддержки.

Введем вектор $\overline{X} = (x_1,...,x_n)$, элемент x_i, $i = \overline{1,n}$ которого определяется следующим образом:

$$x_i = \begin{cases} 1, \text{ если бизнес-процесс } Z_i \text{ обслуживается,} \\ 0, \text{в противном случае.} \end{cases} \quad (7.1)$$

Целевая функция должна оценивать суммарную важность бизнес-процессов, которые будут обслуживаться. С учетом того, что в первую очередь необходимо обслужить наиболее важные для объекта управления бизнес-процессы, критерий приобретает следующий вид:

$$\max \sum_{i=1}^{n} x_i \cdot w_i \,. \quad (7.2)$$

Введем ресурсные ограничения:

$$\sum_{i=1}^{n} x_i \cdot p_{ij} \le r_j, \; j = \overline{1,m} \,. \quad (7.3)$$

Стоимость приобретаемых ресурсов не должна превышать бюджет C на создание ИТ-инфраструктуры:

$$\sum_{j=1}^{m} r_j \cdot c_j \le C \,. \quad (7.4)$$

Если ресурсы приобретаются в минимальном количестве, необходимом только для поддержания бизнес-процессов без резервирования и избыточности, которое требуется для развития, то можно записать:

$$r_j = \sum_{i=1}^{n} x_i \cdot p_{ij}, \quad j = \overline{1,m} \,. \quad (7.5)$$

Поскольку конфигурация ресурсов на этапе планирования еще не определена, а задан только общий бюджет, ограничения (7.3) с учетом (7.4) и (7.5) примут следующий вид:

$$\sum_{j=1}^{m} \sum_{i=1}^{n} x_i \cdot p_{ij} \cdot c_j \le C \,. \quad (7.6)$$

АУТС КШ

Задача (7.2), (7.6) – задача булевого программирования, для решения которой можно использовать соответствующие методы.

В условиях гибкого управления доступом к ресурсам перспективным будет метод управления, при котором полностью обслуживаются важные бизнес-процессы, а менее важные бизнес-процессы обслуживаются с ограниченной функциональностью. Для определения уровня поддержания бизнес-процесса Z_i будем использовать переменную $x_i \in [0,1]$.

Тогда критерий (7.2) и ограничения (7.3) остаются в силе и добавляется ограничение на первичную переменную в виде

$$0 \le x_i \le 1,\ i = \overline{1,n}\,. \tag{7.7}$$

Модель (7.2), (7.6), (7.7) – случай задачи линейного программирования.

После того, как задача будет решена относительно переменных x_i, искомые значения объемов ресурсов r_j, $j = \overline{1,m}$ могут быть получены по формуле (7.5).

Методы решения задач второго уровня. Эффективность бизнеса зависит от эффективности выполнения отдельных бизнес-процессов $\{Z_i\}$, поддерживаемых различными ИТ. Бизнес-процессы имеют различную значимость. Поэтому для управления ИТ-инфраструктурой СУИ должна обладать информацией о значимости $\{w_i\}$ бизнес-процессов $\{Z_i\}$, которая может быть определена, например, посредством приложений автоматизации управления выполнения бизнеса – ВРМ. Значимость бизнес-процессов может изменяться. СУИ должна оперативно реагировать на изменения. Значимость бизнес-процессов может изменяться и тогда, когда СУИ сообщит ВРМ о невозможности поддержания всех бизнес-процессов из-за недоступности части ресурсов вследствие, например, отказа в ИТ-инфраструктуре. В этом случае взаимодействие СУИ и системы ВРМ осуществляется в соответствии с принципами декомпозиционно-компенсационного подхода [105, 325, 326, 327].

При этом взаимодействие СУИ и ВРМ осуществляется посредством следующего итеративно выполняемого алгоритма.

ВРМ на основании стратегических целей ведения бизнеса, реальной обстановки и информации, получаемой от СУИ, о существующем уровне поддержания бизнес-процессов $Z_1,...,Z_n$ формирует коэффициенты важности $w_1,...,w_n$ и передает их в СУИ.

СУИ, рассчитав конфигурацию ресурсов, определяет степен поддержания бизнес-процессов и передает ее ВРМ.

В случае получения подтверждения от ВРМ, СУИ реализует рассчитанную на предыдущем шаге конфигурацию ресурсов.

Приведенный алгоритм можно использовать не только при отработке нештатных ситуаций, но и при изменении целей бизнеса, анализе возможностей модернизации ИУС и других подобных ситуациях.

Задачу определения наиболее важных бизнес-процессов, которые могут быть обслужены в ИТ-инфраструктуре, сформулируем следующим образом: для заданных значений тройки

$$< \hat{w}, \hat{r}, P > \tag{7.8}$$

определить процессы из множества $\hat{Z}$, для которых критерий (7.2) достигает максимума с учетом ресурсных ограничений (7.3).

Если СУИ только разрешает или блокирует выполнение отдельных бизнес-процессов, то задача (7.2), (7.3) является задачей булевого программирования.

Если СУИ полностью не прекращает выполнение части процессов, а только ограничивает их функциональность, предоставляя возможность менее важным процессам хоть как-то, но выполняться, то к задаче (7.2), (7.3) добавляются ограничения на непрерывную переменную x_i в виде (7.7) Тогда получаем и задачу линейного программирования (7.2), (7.3), (7.7).

После решения этой задачи объемы ресурсов, выделяемых бизнес-процессам, можно определить следующим образом:

$$d_{ij} = p_{ij} x_i,\ i = \overline{1,n},\ j = \overline{1,m}, \tag{7.9}$$

где d_{ij} – объем ресурса j-го типа, который используется i-м процессом.

Задачи третьего уровня. Ресурсы должны быть закреплены за $\hat{Z}$ так, чтобы при имеющихся ресурсных ограничениях общая важность обслуживания бизнес-процессов была максимальной. Можно говорить о закреплении ресурсов за теми бизнес-процессами, которые имеют наивысшую суммарную важность и могут быть эффективно поддержаны имеющимися ресурсами. При этом некоторые процессы могут быть переведены на ограниченное функционирование.

Модели и методы решения задач четвертого уровня. Задача управления этого уровня формулируется как задача закрепления за выбранными на третьем уровне бизнес-процессами конкретных ресурсов, при которых интегрированный критерий эффективности закрепления ресурсов получит максимальное значение при условии выполнения временных, технологических, ресурсных и других ограничений.

Для решения задачи распределения или перераспределения и закрепления конкретных ресурсов введем следующие обозначения:

R_j – пул ресурсов типа j, где $j = \overline{1,m}$. Это может быть пул серверов приложений, пул сетевых дисков, пропускная способность каналов связи и т. д.;

m_j – количество отдельных ресурсов в пуле R_j, $j = \overline{1,m}$;

R_{jk} – k-й ресурс в пуле R_j, $j = \overline{1,m}$, $k = \overline{1,m_j}$;

r_{jk} – объем ресурса k в пуле R_j, $j = \overline{1,m}$, $k = \overline{1,m_j}$;

e_{ijk} – техническая возможность поддержания бизнес-процесса Z_i ресурсом R_{jk}

$$e_{ijk} = \begin{cases} 1, & \text{если процесс может быть поддержан,} \\ 0, & \text{в противосм случае.} \end{cases} \tag{7.10}$$

Введем переменную y_{ijk}, которая определяет закрепление ресурсов за выбранными для поддержания бизнес-процессами:

$$y_{ijk} = \begin{cases} 1, & \text{если процесс } Z_i \text{ поддерживается,} \\ 0, & \text{в противном случае,} \end{cases} \tag{7.11}$$

для $i = \overline{1,n}$; $j = \overline{1,m}$; $k = \overline{1,m_j}$.

АУТСОРС

Закрепление ресурсов представим в следующем виде:

$$\sum_{k=1}^{m_j} e_{ijk} \cdot y_{ijk} = x_i \,,\ i = \overline{1,n} \,,\ \ j = \overline{1,m} \,. \tag{7.12}$$

Это выражение означает, что потребность в ресурсах выбранного для поддержания процесса x_i должна быть удовлетворена одним из конкретных ресурсов при условии, что этот ресурс может быть использован для поддержания указанного бизнес-процесса.

Ресурсные ограничения запишем в таком виде:

$$\sum_{i=1}^{n} y_{ijk} \cdot p_{ij} \le r_{jk} \,,\ \ j = \overline{1,m} \,,\ \ k = \overline{1,m_j} \,. \tag{7.13}$$

Если управление доступом к ресурсам реализовано так, что СУИ либо разрешает, либо блокирует работу приложений, поддерживающих бизнес-процессы, и закрепляет задачи процессов только за одним из ресурсов в каждом пуле ресурсов, то переменные x_i и y_{ijk} принимают значение 0 или 1, а задача (7.2), (7.12), (7.13) является задачей булевого программирования.

Если СУИ обеспечивает поддержку всех процессов, но некоторых частично, а задачам могут быть предоставлены несколько ресурсов из пула ресурсов, т. е. осуществляется балансировка нагрузки между серверами одного типа в пуле серверов, то переменные x_i и y_{ijk} необходимо рассматривать как непрерывные, и добавить ограничения (7.7) и

$$0 \le y_{ijk} \le 1 \,,\ i = \overline{1,n} \,,\ \ j = \overline{1,m} \,,\ \ k = \overline{1,m_j} \,, \tag{7.14}$$

Тогда задача (7.2), (7.7), (7.12)–(7.14) является задачей линейного программирования.

Решение задач (7.2), (7.12), (7.13) можно получить с помощью эвристических методов перебора с отсеиванием бесперспективных вариантов [37] или посредством вариантов генетического алгоритма, предложенных в главе 10.

Методы решения задач пятого уровня. Здесь можно использовать упорядочение по важности $\{w_i\}$, а в случае равенства важности преимущество отдавать бизнес-процессам, требующим меньше времени на выполнение. Но этот подход дает лишь локальный эффект. В таких сложных распределенных системах, как ИТ-инфраструктура, есть взаимозаменяемые ресурсы, управление доступом к которым на локальном уровне не дает желаемого эффекта, поскольку не обеспечивает их равномерного использования. Случайный характер процессов делает уместным диспетчирование доступа к таким ресурсам в реальных сложившихся условиях для их равномерной загрузки. Это также позволит повысить надежность системы без резервирования.

Диспетчерское управление заключается в составлении расписаний выполнения запросов бизнес-процессов на ресурсы с максимально эффективным их использованием при условии выполнения временны́х, технологических и других ограничений и реализации этих расписаний.

Одной из наиболее важных проблем управления ресурсами на этом уровне является проблема балансирования серверов приложений. В общем случае речь

идет о такой загрузке серверов приложений задачами, которая обеспечит эффективное использование вычислительных ресурсов, реализацию наиболее важных процессов деятельности при выполнении упомянутых выше ограничений. Поскольку серверы приложений базируются на серверах разной продуктивности, то необходимо говорить про рейтинговое балансирование.

Для реализации рейтингового балансирования в СУИ выделяется специальный компонент – диспетчер, который тесно взаимодействует с другими компонентами СУИ, в первую очередь – с системой мониторинга. Система мониторинга собирает информацию о производительности серверов, накапливает статистику реально использованного времени и других ресурсов на решение задач и формирует оценку загруженности серверов.

Копия диспетчера устанавливается на каждом сервере, но в каждый момент времени только одна из них активна, остальные экземпляры используются для резервирования. Диспетчер реализует выбранную стратегию распределения или переходит к новой стратегии в зависимости от ситуации, которая складывается в процессе обработки.

Задачи поступают в систему на соответствующие серверы. Активная копия диспетчера контролирует загруженность серверов и перераспределяет задачи в определенных, заведомо обусловленных ситуациях, связанных с неравномерной загруженностью серверов, их выходом из строя или невозможностью выполнить задачу на определенном сервере из-за существующих временны́х ограничений и т. д.

Считая задачи несвязанными по данным, после решения проблем выбора комплекса бизнес-процессов и распределения конкретных ресурсов будем использовать такую методику распределения нагрузки серверов приложений ИТС:

1) присвоение приоритетов на обслуживание в соответствии с важностью бизнес-процессов;

2) управления доступом к ресурсам по приоритетам при возникновении очередей;

3) перераспределение задач в случаях, требующих участия диспетчера;

4) отклонение запросов, являющихся наименее важными из бизнес-процессов, которые не входят в распределение, когда очереди переполняются.

Кратко рассмотрим этап перераспределения задач, который, по сути, и определяет эффективность балансирования. Отметим, что все задачи, которые поступают на выполнение, принадлежат к определенным бизнес-процессам. Все бизнес-процессы имеют определенный показатель качества и делятся на три основные группы, которые имеют разные права и возможности в борьбе за вычислительные и другие ресурсы:

1) постоянные бизнес-процессы: наиболее важные для предприятия бизнес-процессы; приложения, которые отвечают за их реализацию, постоянно находятся в ОЗУ, а выделенный им ресурс не учитывается при распределении для задач других бизнес-процессов;

2) долговременные бизнес-процессы: это важные для предприятия бизнес-процессы, приложения которых желательно держать в ОЗУ в случае резкого уменьшения ресурса и при появлении кратковременного процесса с более высоким приоритетом;

3) кратковременные бизнес-процессы: это бизнес-процессы, которые нелогично постоянно или долго держать в ОЗУ; они берут участие в борьбе за ре-

сурсы на основе приоритетов, определенных на основании их важности; занимают вычислительный ресурс на определенное время и освобождают его по окончанию задачи. Важную подгруппу составляют периодические бизнес-процессы, которые требуют адекватной реакции СУИ.

К этапу перераспределения задач диспетчер переходит в упомянутых выше ситуациях, которые распознаются диспетчером на основе собранной в результате мониторинга ИТ-инфраструктуры информации. Для каждой задачи, требующей перераспределения, диспетчер сначала определяет наиболее выгодный сервер в соответствии со стратегией выбора серверов приложений ИУС, а затем передает задачу соответствующему серверу приложений. Следующим шагом является переопределение части свободных ресурсов соответствующего сервера приложений (уменьшение поточного рейтинга этого сервера) на прогнозированное время выполнения задачи. В конце диспетчер корректирует стратегию выбора серверов приложений ИУС в соответствии с измененным распределением ресурсов. После выполнения задачи диспетчер управляет отправкой результатов решения задачи пользователю, приславшему запрос.

Прогнозирование времени выполнения задач можно осуществлять на основе накопленной статистики или с использованием марковской модели. Оценку производительности серверов приложений можно выполнять на основе одного из стандартных контрольно-оценивающих тестов, например Linpack, Perfect, Spec-89 (-92, -SDM, SFS) или AIM.

Можно применять несколько стратегий выбора серверов приложений ИУС, а потом выбрать лучшую из них на основе накапливаемой статистики. Наиболее применимой является стратегия предоставления задаче ближайшего незагруженного сервера, для которого выполняются временные ограничения и имеется соответствующий ресурс.

Автоматическая реализация управления доступом бизнес-процессов к ограниченным ресурсам ИУС может быть выполнена путем построения СУИ на основе принципов, изложенных в работах [120, 154], с использованием агентских технологий [117] и стандартных протоколов управления CMIP, SNMP и т. д.

Модели и методы решения задач шестого уровня. Рассмотрим задачу оптимизации наращивания ресурсов при модернизации и развитии ИТ-инфраструктуры, которую можно сформулировать следующим образом.

ИТ-инфраструктура была спроектирована решением задач первого уровня, когда не все бизнес-процессы поддерживаются в полном объеме.

Выделен бюджет развития ΔC. Требуется найти такое приращение ресурсов $\Delta r_1,...,\Delta r_m$ и обеспечить поддержания таких процессов $Z_1,...,Z_n$, чтобы максимизировать критерий (7.2) при следующих ограничениях. Ресурсные ограничения:

$$\sum_{i=1}^{n} x_i \cdot p_{ij} \le r_j + \Delta r_j \ , \tag{7.15}$$

или

$$\sum_{i=1}^{n} x_i \cdot p_{ij} - \Delta r_j \le r_j \ , \ \ j = \overline{1,m} \, . \tag{7.16}$$

Стоимостные ограничения:

$$\sum_{j=1}^{m} \Delta r_j \cdot c_j \le \Delta C . \qquad (7.17)$$

При этом каждый бизнес-процесс может быть поддержан полностью ($x_i = 1$) или частично ($x_i < 1$), т. е.

$$0 \le x_i \le 1, \ i = \overline{1,n} . \qquad (7.18)$$

Если в процессе развития ресурсы можно только приобретать, то должно выполняться условие

$$\Delta r_j \ge 0, \quad j = \overline{1,m} . \qquad (7.19)$$

Если неиспользуемые ресурсы могут быть проданы по первоначальной стоимости, а арендуемые – сданы в субаренду, то ограничение (7.19) принимает вид

$$\Delta r_j \ge -r_j, \quad j = \overline{1,m} . \qquad (7.20)$$

Задачи (7.2), (7.7), (7.16), (7.17), (7.19) и (7.2), (7.7), (7.16), (7.17), (7.20) являются задачами линейного программирования относительно переменных $x_1,...,x_n$, $\Delta r_1,...,\Delta r_m$ (всего $n + m$ переменных). Их решение относительно этих переменных позволяет одновременно получить оптимальные значения необходимого приращения ресурсов и степень поддержки бизнес-процессов в соответствии с критерием (7.2).

Таким образом, предложен процессно-ресурсный подход к управлению доступом к ограниченным ресурсам и разработаны модели и методы для ИТ УИТИ.

7.2. МОДЕЛИ И МЕТОДЫ УПРАВЛЕНИЯ РАСПРЕДЕЛЕНИЕМ ОГРАНИЧЕННЫХ РЕСУРСОВ КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРЫ

7.2.1. Двухуровневая и многоуровневая модели управления распределением ограниченных ресурсов

Двухуровневая модель управления распределением ограниченных ресурсов. Простейшая модель распределения ограниченных ресурсов может быть представлена в виде нескольких бизнес-процессов, использующих часть одного или нескольких независимых друг от друга ресурсов [148] (рис. 7.2).

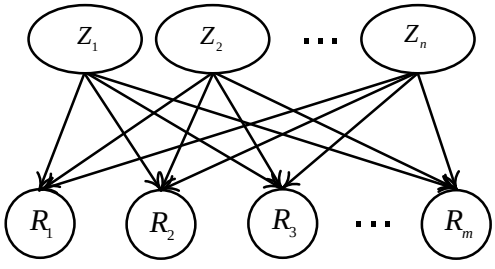


Рис. 7.2. Двухуровневая модель распределения ресурсов ИТ-инфраструктуры

Тогда задача распределения ресурсов ИТ-инфраструктуры формулируется следующим образом.

Пусть заданы значения пятерки:

$$< \hat{Z}, \hat{w}, R, P, \hat{r} > . \tag{7.21}$$

Необходимо определить такие элементы вектора *X*, приобретающие значения из (7.1), при которых критерий (7.2) принимает наилучшее значение при выполнении ресурсных ограничений (7.3).

В матричной форме задачи (7.2), (7.3) можно представить следующим образом:

$$\max X \cdot \hat{w}, \tag{7.22}$$

$$\overline{X} \times P \le \hat{r} . \tag{7.23}$$

Задача (7.21), (7.2), (7.3) – задача булевого программирования, для решения которой можно использовать соответствующие методы [37].

В специальной литературе можно встретить описание двухуровневых моделей и соответствующих методов распределения ресурсов. Однако тщательный анализ проблем распределения корпоративных ресурсов ИТ-инфраструктуры позволил выявить несовершенство двухуровневой модели, обусловившее переход к многоуровневым моделям.

Многоуровневая модель управления распределением ограниченных ресурсов. В ИТ-инфраструктуре часто ресурсы зависят друг от друга. Например, сервер приложений при обработке запросов пользователя может в свою очередь делать запросы к серверу БД, а внешнее хранилище данных при выполнении операций чтения/записи использует часть сетевых ресурсов.

Предположим, что при использовании ресурсов ИТ-инфраструктуры отсутствует взаимная зависимость между ресурсами. Т. е. если, например, использование ресурса R_1 требует использования ресурса R_2, то функционирование ресурса R_2 не должно прямо или косвенно зависеть от наличия ресурса R_1. Тогда на ориентированном графе зависимостей ресурсов циклы не возникают (рис. 7.3). Необходимо отметить, что подобная ситуация характерна для большинства ИУС, функционирующих по принципу клиент-сервер (как двух-, так и многозвенных).

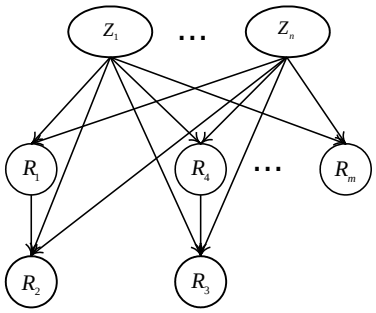


Рис. 7.3. Многоуровневая модель использования ресурсов ИТ-инфраструктуры

Для учета ресурсов, задействуемых для поддержания функционирования других ресурсов, введем матрицу $V = \|v_{kj}\|$ размером $m \times m$, где v_{kj} – количество ресурса R_j, $j = \overline{1,m}$, затрачиваемое при использовании единицы ресурса R_k, $k = \overline{1,m}$.

Тогда для трехуровневой модели использования ресурсов, когда любой путь на графе содержит не более двух ресурсов, ограничения (7.23) можно представить в виде

$$\overline{X} \times P + \overline{X} \times P \times V \le \hat{r} . \tag{7.24}$$

В выражении (7.24) первая составляющая отвечает за ресурсы, непосредственно задействуемые на обеспечение функционирования выбранных бизнес-процессов, а вторая – за потери, связанные с поддержанием использования данных ресурсов.

Очевидно, в общем случае для l уровней ограничения принимают вид

$$\overline{X} \times P \times \sum_{i=0}^{l-2} V^i \le \hat{r}\,, \tag{7.25}$$

причем $V^0 = E,\ V^1 = V,\ V^2 = V \times V\ ,$ где E – единичная матрица размером $m \times m$.

Поскольку матрицы P и V известны, выражение (7.25) можно записать в виде

$$\overline{X} \times P' \le \hat{r}\ , \tag{7.26}$$

где $P' = P \times \sum_{i=0}^{l-2} V^i$.

Таким образом, задача (7.22), (7.25) сводится к задаче (7.22), (7.23).

Преимуществом многоуровневой модели является совпадение ее структуры со структурой модели, используемой для корреляции событий при выявлении неисправного ОМУ.

Недостатком многоуровневой модели является то, что она не учитывает возможность использования ресурсами $R_j,\ j = \overline{1,m}$ вспомогательных ресурсов при обслуживании бизнес-процессов $Z_i,\ i = \overline{1,n}$. Этот недостаток может быть легко устранен путем построения модели, предусматривающей закрепление определенного количества вспомогательного ресурса за конкретным бизнес-процессом, как это происходит в двухуровневой модели, или путем замены конкретного ресурса на несколько абстрактных и учета дополнительных ограничений на эти абстрактные ресурсы.

В примере, приведенном на рис. 7.4, бизнес-процессы Z_1 и Z_2 используют сервер приложений S, причем только в интересах Z_1 сервер S обращается к серверу БД. Модель, представленная на рис. 7.4,*а*, неадекватна, поскольку предполагает доступ к серверу БД каждого из бизнес-процессов Z_1 и Z_2. Рассчитанные на ее основе потребности в ресурсах будут завышены.

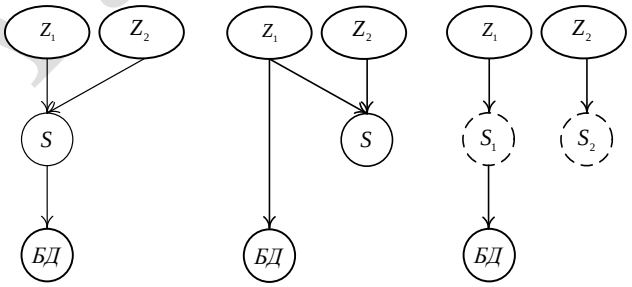


Рис. 7.4. Пример трехуровневой модели: *а* – неадекватная модель; *б* – комбинированная модель; *в* – абстрактные ресурсы

В модели, приведенной на рис. 7.4,*б*, для бизнес-процесса Z_1 явно указано использование ресурса сервера БД, как в двухуровневой модели, хотя в действительности этот ресурс используется опосредовано через сервер *S*. Такая модель позволяет точно рассчитать потребности в ресурсах, но она неэффективна для решения задач диагностики.

При использовании в ИТ УИТИ модели, приведенной на рис. 7.4,*в*, анализ сообщений об отказе в обслуживании процесса Z_1 и состояния серверов S_1 и S_2 позволяет коррелятору событий подсистемы мониторинга определить, что вышел из строя именно сервер БД.

Недостатком такого подхода является увеличение размеров матриц *P* и *V*, а также появление дополнительного ограничения: суммарное потребление абстрактных ресурсов S_1 и S_2 не должно превышать количество ресурса *S*.

7.2.2. Обеспечение бизнес-процессов ресурсами с определенным уровнем надежности

Одной из важнейших задач ИТ УИТИ является управление при возникновении неисправностей. Влияние неисправностей проявляется в отсутствии доступа к части ресурсов ИТ-инфраструктуры, что, в свою очередь, приводит к снижению эффективности выполнения бизнес-процессов. В этом случае СУИ должна перераспределить оставшиеся ресурсы с учетом важности $\hat{w}$ процессов $\hat{Z}$ и требований к надежности их функционирования, а при появлении неполадок, не позволяющих нормально работать всем процессам, обеспечить функционирование важнейших из них.

Математические модели и методы распределения ресурсов предложены, в частности, в работах [148, 149, 157]. Однако практически отсутствуют работы, в которых рассматриваются вопросы распределения ограниченных ресурсов для поддержания наиболее важных бизнес-процессов с обеспечением их функционирования с заданной надежностью. Поэтому целесообразно разработать модели управления распределением ограниченных ресурсов ИТ-инфраструктуры при условии обеспечения необходимого уровня надежности функционирования бизнес-процессов.

В модели, приведенной на рис. 7.2, для ресурсов R_j, $j=\overline{1,m}$, кроме вектора *W* коэффициентов важности w_i, $i=\overline{1,n}$ бизнес-процессов Z_i, $i=\overline{1,n}$ определен вектор $\hat{\mu}$ требований к показателям надежности функционирования бизнес-процессов:

$$\hat{\mu}=(\mu_1,...,\mu_n), \tag{7.27}$$

например, для коэффициента готовности – $0<\mu_i\le 1$, для всех $i=\overline{1,n}$.

Для интегрированных независимых ресурсов R_j, $j=\overline{1,m}$ определены показатели надежности ресурсов:

$$\hat{\zeta}=(\zeta_1,...,\zeta_m), \tag{7.28}$$

например, для коэффициента готовности – $0<\zeta_i\le 1$, для всех $j=\overline{1,m}$.

Кроме матрицы P, задающей количество требующихся процессам ресурсов и ограничений на ресурсы (7.3), имеется матрица D наличия потребностей бизнес-процессов в ресурсах ИТ-инфраструктуры:

$$D = \left\| d_{ij} \right\|. \tag{7.29}$$

Причем

$$\begin{cases} d_{ij} = 1, \text{ если } p_{ij} > 0, \\ d_{ij} = 0, \text{ если } p_{ij} = 0. \end{cases} \tag{7.30}$$

Для дискретного случая, когда выполняется условие (7.1), бизнес-процессы Z_i, $i = \overline{1,n}$ поддерживаются в полном объеме, либо полностью блокируются. Критерий оптимального распределения ресурсов представляется в виде (7.2) при ресурсных ограничениях (7.3).

Рассмотрим случай, когда бизнес-процесс Z_i, $i = \overline{1,n}$ поддерживается в полном объеме только при наличии всех требуемых ему ресурсов. Тогда для оценки надежности поддержания процесса воспользуемся последовательной схемой, при которой надежность системы рассчитывается как результат перемножения надежностей РОУ, в данном случае – надежности используемых ресурсов:

$$\zeta = \zeta_1 \cdot \zeta_2 \cdot \ldots \cdot \zeta_m. \tag{7.31}$$

При этом расчетная надежность должна быть не меньше заданной μ_i, $i = \overline{1,n}$.

Для того чтобы избавиться от операции умножения, выполним преобразование обеих частей выражения (7.31):

$$\ln \zeta = \ln \zeta_1 + \ln \zeta_2 + \ldots + \ln \zeta_m. \tag{7.32}$$

Учитывая только те ресурсы, которые задействуются для конкретного процесса Z_i, $i = \overline{1,n}$, запишем требования к надежности следующим образом:

$$\sum_{j=1}^{m} x_i \cdot d_{ij} \cdot \ln \zeta_j \geq \ln \mu_i, \; i = \overline{1,n}. \tag{7.33}$$

Поскольку μ_i задается в интервале (0, 1], правая часть выражения (7.33) всегда будет отрицательна. Для бизнес-процессов, которым будет отказано в обслуживании ($x_i = 0$), левая часть выражения (7.33) будет равна нулю. Следовательно, выражение (7.33) будет выполняться при любых значениях ζ_j и μ_i. Но это означает, что оно не будет препятствовать отказу от поддержания процесса из-за нехватки ресурсов. Для остальных процессов это выражение будет выполнять роль ограничения по надежности.

Задача (7.2), (7.3), (7.33) является задачей булевого программирования и может быть решена методами, описанными в работе [37].

Рассмотренная задача распределения ресурсов ближе к реалиям практики. Однако она не позволяет учесть возможность/невозможность поддержания про-

цесса Z_i, $i=\overline{1,n}$ реальными ресурсами, определяемую техническими или географическими ограничениями. Практический интерес представляет задача распределения конкретных реальных ресурсов.

Будем рассматривать интегрированный ресурс R_j как группу реальных ресурсов одного типа, но разного объема и надежности, как это показано на рис. 7.5.

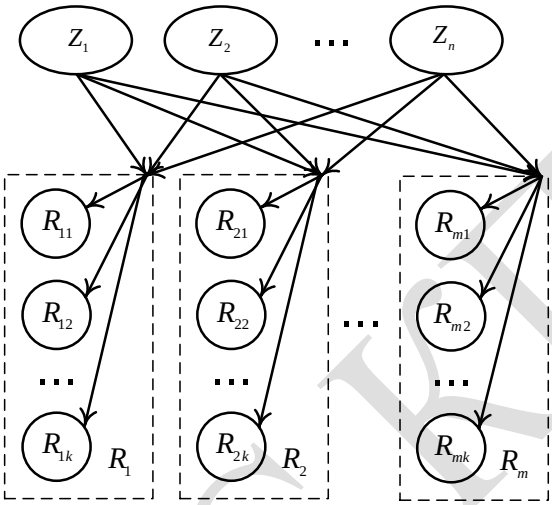


Рис. 7.5. Модель распределения определенных реальных ресурсов ИТ-инфраструктуры

Введем дополнительные обозначения:

m_j – количество отдельных ресурсов в группе R_j, $j=\overline{1,m}$;

R_{jk} – конкретные ресурсы, $j=\overline{1,m}$, $k=\overline{1,m_j}$;

r_{jk} – объем ресурса k в группе R_j, $j=\overline{1,m}$, $k=\overline{1,m_j}$;

ζ_{jk} – надежность ресурса k в группе R_j, $j=\overline{1,m}$; $k=\overline{1,m_j}$;

e_{ijk} – возможность поддержания процесса Z_i конкретным ресурсом R_{jk} (7.10);

y_{ijk} – определяет закрепление определенных реальных ресурсов за выбранными для поддержания процессами и соответствует (7.11) для $i=\overline{1,n}$, $j=\overline{1,m}$ и $k=\overline{1,m_j}$. В данном случае переменную y_{ijk}, определяющую закрепление определенных реальных ресурсов за выбранными для поддержания процессами, целесообразно определить следующим образом:

$$y_{ijk}=\begin{cases}1, \text{если ресурс } R_{jk} \text{ привлекается для поддержания процесса } Z_i;\\ 0, \text{ в противном случае,}\end{cases} \quad (7.34)$$

для $i=\overline{1,n}$, $j=\overline{1,m}$ и $k=\overline{1,m_j}$.

Закрепление ресурсов представим следующим выражением:

АУПСКП

$$\sum_{k=1}^{m_j} e_{ijk} \cdot y_{ijk} = d_{ij} \cdot x_i \ , \ i = \overline{1,n} \ , \ \ j = \overline{1,m} \ . \qquad (7.35)$$

Выражение (7.35) означает, что каждая потребность в ресурсах d_{ij} для поддержания выбранного процесса Z_i должна быть удовлетворена одним из конкретных ресурсов, если этот ресурс может быть выделен для поддержания данного процесса.

По аналогии с (7.3) ресурсные ограничения будут иметь вид

$$\sum_{i=1}^{n} y_{ijk} \cdot p_{ij} \le r_{jk} \ , \ i = \overline{1,n} \ , \ \ j = \overline{1,m} \ . \qquad (7.36)$$

Ограничения по надежности по аналогии с (7.33) будут иметь вид

$$\sum_{j=1}^{m} \sum_{k=1}^{m_j} y_{ijk} \cdot \ln \zeta_{jk} \ge \ln \mu_i \ , \ i = \overline{1,n} \ . \qquad (7.37)$$

Если рассматривать x_i и y_{ijk} как переменные, то задача (7.2), (7.35), (7.36), (7.37) является задачей булевого программирования. Следует отметить, что в связи с увеличение размерности задачи по сравнению с предыдущей для ее эффективного решения можно использовать эвристические и генетические алгоритмы.

Таким образом, разработаны модели распределения ограниченных ресурсов с учетом требований к надежности со стороны бизнес-процессов. На основе двухуровневой модели предложена многоуровневая модель управления распределением ограниченных ресурсов ИТ-инфраструктуры и способ перехода от многоуровневой к двухуровневой модели. Сравнены свойства моделей и предложены способы устранения выявленных недостатков многоуровневой модели. Целесообразность применения той или иной модели распределения ресурсов существенно зависит от того, предполагается ли ее использование также в качестве диагностической модели.

Применение моделей в СУИ позволит увеличить эффективность функционирования ИТ-инфраструктуры, что особенно важно для ИУС специального назначения.

7.2.3. Модель управления распределением ограниченных ресурсов в ИТ-инфраструктуре с иерархической архитектурой ресурсов

ИТ-инфраструктура обеспечивает доступ к общим ресурсам, предоставляет пользователям ИТ-услуги, осуществляет информационную связь между компонентами распределенных приложений и элементами ИТ-инфраструктуры. СУИ использует часть общих ресурсов для собственных нужд. Ресурсы ИТ-инфраструктуры ограничены и достаточно часто возникает ситуация, когда их не хватает для эффективного функционирования всех приложений и решения задач пользователей. Такие ситуации возникают вследствие, например, появления новых задач или пользователей, включения крупных компонентов ИТ-инфраструктуры. Тогда возникает необходимость использования различных методов управления доступом к ограниченным ресурсам. Распределение или перераспределение ресурсов

СУИ должна осуществлять с учетом общей цели функционирования ИТ-инфраструктуры – обеспечение высокой эффективности работы бизнес-процессов. При этом должны учитываться приоритеты задач, которые в настоящее время выполняются, и задействованные ими ресурсы, а также приоритеты новых задач и ресурсы, в которых они нуждаются.

Для разработки математической модели введем пространство:

$$H = \{h_i\},\ i = \overline{1,n}, \tag{7.38}$$

элементами которого являются все ресурсы и задачи ИТ-инфраструктуры.

Целесообразность такого объединения ресурсов и задач в едином пространстве обусловлена тем, что задачи могут требовать ресурсы, выделение которых порождает новые задачи. В пространство *H* включены все имеющиеся ресурсы ИТ-инфраструктуры и все возможные задачи, которые могут выполняться в ИТ-инфраструктуре для обеспечения работы имеющихся и планируемых распределенных приложений. При добавлении ресурсов вследствие расширения и модификации ИТ-инфраструктуры или установки новых приложений, которые не были предусмотрены, а также в случае безвозвратного удаления ресурсов или приложений размерность пространства *H* соответственно увеличивается или уменьшается.

В пространстве *H* выделим подпространства:

$$Z = \{z_i\},\ i = \overline{1,n}, \tag{7.39}$$

элементами которого являются задачи, поддерживающие бизнес-процессы;

$$R = \{r_i\},\ i = \overline{1,n}, \tag{7.40}$$

элементы которого – ресурсы, используемые задачами.

Для выбора ресурсов или задач из пространства *H* введем *n*-мерные векторы $\overline{V}_r$ и $\overline{V}_z$, первый из которых определяет ресурсы, а второй – задачи. Элементы векторов принимают значения 0 или 1, причем 1 устанавливается на тех позициях, например, вектора $\overline{V}_r$, которые соответствуют ресурсам, а 0 – задачам.

Зададим $\widehat{w} = \{w_i\},\ i = \overline{1,n}$ – коэффициенты важности ресурсов и задач.

Использование ресурсов будем определять с помощью матрицы

$$P = \left\| p_{i,j} \right\|,\ i, j = \overline{1,n}, \tag{7.41}$$

элемент p_{ij} которой представляет нормированный коэффициент использования элементом h_i элемента h_j.

Зададим вектор $\overline{X} = (x_1, ..., x_n)$, определяющий задачи из подпространства *Z*, которые в данный момент выполняются:

$$x_j = \begin{cases} 1, \text{ если задача } z_j \text{ выполняется;} \\ 0, \text{ в противном случае.} \end{cases} \tag{7.42}$$

АУТСОРС

Зададим вектор $\overline{D} = (d_1,...,d_n)$, выделяющий задачи из матрицы *P*, выполнение которых может быть отменено для высвобождения ресурсов, или использование ими ресурсов может быть уменьшено для обеспечения решения новых задач:

$$d_j = \begin{cases} 1, \text{ если выполнение задачи } z_j \text{ можно отменить;} \\ 0, \text{ в противном случае.} \end{cases} \quad (7.43)$$

Для выделения из пространства *H* задач или ресурсов используются следующие выражения:

$$L = \overline{V_z} \times H, \quad (7.44)$$

$$R = \overline{V_r} \times H. \quad (7.45)$$

Поскольку вектор $\overline{D}$ определяет только элементы, являющиеся задачами, то будет выполняться условие:

$$\overline{D} = \overline{D} \times \overline{V_r}. \quad (7.46)$$

Максимальная эффективность будет обеспечена тогда, когда будет выполняться критерий (7.2), подразумевающий достижение максимума суммарной важности выполняемых задач.

Поскольку элемент p_{ij} матрицы *P* является нормированным коэффициентом использования задачей z_j ресурса r_j, то общий объем использования любого ресурса не может превышать единицу, т. е. всегда будет выполняться условие:

$$\sum_{i=1}^{n} p_{ij} \cdot x_i \le 1, \ j = \overline{1,n}. \quad (7.47)$$

Пусть появляется новая задача z'_k, $k = \overline{1,n}$, для которой предварительно выполнялось условие $x_k = 0$. Если для ее выполнения недостаточно ресурсов, то необходимо высвободить ресурсы или принять решение, что z'_k не может быть выполнена, поскольку не имеет достаточной важности, с учетом политики ограничения на ресурсы, которая используется в СУИ в данный момент.

Определим количество r_j свободных ресурсов ИТ-инфраструктуры:

$$r_j = 1 - \sum_{i=1}^{n} p_{ij} \cdot x_i, \ j = \overline{1,n}. \quad (7.48)$$

Для каждого ресурса r_j, необходимого задаче z'_k, проверяется выполнение условия:

$$r_j \ge p_{kj}, \ j = \overline{1,n}. \quad (7.49)$$

и если условие выполняется, то это свидетельствует о том, что в ИТ-инфраструктуре имеется достаточное количество ресурсов, необходимых для решения задачи z'_k, и дополнительное управление по перераспределению ресурсов не требуется.

АУТСКП

Если в ИТ-инфраструктуре нет достаточного количества незадействованных ресурсов, необходимых задаче z'_k, то СУИ осуществляет освобождение части ресурсов, занятых менее важными задачами.

Определим Δ_j, $j=\overline{1,n}$, как количество ресурса r_j, недостающего для решения задачи z'_k:

$$\Delta_j = r_j - p_{kj}, \ j=\overline{1,n}. \tag{7.50}$$

Тогда для управления перераспределением ресурсов необходимо найти все такие задачи z_j, отмена которых освободит ресурсы, необходимые задаче z'_k.

Определим критерии, которые учитываются при решении задачи.

1. Изымать ресурсы у тех задач, суммарная важность которых минимальна:

$$\min \sum_{i=1}^{n} w_i d_i . \tag{7.51}$$

2. Прекращать выполнение наименьшего количества задач, которые находятся в стадии выполнения:

$$\min \sum_{i=1}^{n} d_i x_i . \tag{7.52}$$

3. При освобождении ресурса r_j может потребоваться прекращение выполнения некоторого количества низкоприоритетных задач. Учитывая взаимозависимость ресурсов и задач, это может привести к высвобождению большего количества некоторого ресурса r_j^*, чем необходимо. Это, в конечном итоге, приведет к снижению эффективность работы ИТ-инфраструктуры. Для предотвращения чрезмерного высвобождения ресурсов используется следующий критерий:

$$b_j = \sum_{i=1}^{n} p_{ij} d_i - \Delta_j; \ b_j \to \min; \ b_j \geq 0, \tag{7.53}$$

для всех $j=\overline{1,n}$, причем величина b_j – объем ресурса, который высвобождается, всегда будет положительной, поскольку необходимо освободить ресурс не менее требуемого объема.

Задачи управления с критериями (7.51)–(7.53) являются типичными задачами комбинаторики. Для поиска на заданном множестве набора элементов, удовлетворяющего определенным критериям, очень эффективны генетические алгоритмы.

Однако быстрый поиск набора задач, которые удовлетворяют данным критериям, не является возможным, поскольку требуется преобразовать большое количество комбинаций для каждого из ресурсов. Для ускорения поиска набора задач, удовлетворяющих данным критериям, введем параметр p_i^*, характеризующий значение усредненного использования ресурсов ИТ-инфраструктуры *i*-й задачей:

$$p_i^* = \frac{\sum_{j=1}^{n} p_{ij} u_j}{n^*}, \quad j = \overline{1,n}. \tag{7.54}$$

В выражении (7.54) n^* – количество ненулевых составляющих $p_{ij}u_j$, а u_j – определяет использование ресурса задачей z_j., причем

$$u_j = \begin{cases} 1, \text{ если задача } z_j \text{ использует ресурс } r_j; \\ 0, \text{ в противном случае.} \end{cases}$$

Введение параметра p_i^* осуществлено в предположении существования пропорциональной зависимости между коэффициентами использования различных ресурсов одной задачей. Считается, что если задача z_j. использует один из ресурсов с большим значением коэффициента, то и другие ресурсы также будут использоваться ею с большим коэффициентом. Например, если задача оперирует большими объемами данных, то она будет использовать и большие объемы памяти, необходимые для их хранения, большой процент процессорного времени для их обработки и т. п.

В процессе управления для разрешения конфликта между новой задачей z'_k и задачами, уже использующими эти же ресурсы ИТ-инфраструктуры, в зависимости от значения p_i^* применяется один из таких трех вариантов:

1. При $p_i^* = 0$ задача z_i использует ресурсы, на которые не претендует новая задача z'_k.

2. Если $p_i^* \ll p_k^*$, то задача z_i использует столь незначительные объемы ресурсов, на которые претендует задача z'_k, что удаление z_i не повлияет на решение проблемы.

3. В остальных случаях задача z_i использует значительные объемы тех же ресурсов, на которые претендует z'_k.

Приведенная выше логика разрешения конфликтов позволяет отбросить задачи, которые не влияют на управление распределением ресурсов, и осуществлять поиск среди задач, которые предназначены к удалению, как таких, которые могут высвободить существенное количество необходимых для задачи z'_k ресурсов.

Поскольку сформулированная проблема управления в реальных условиях требует больших вычислительных затрат, предлагается эвристический алгоритм ее решения, позволяющий существенно уменьшить перебор вариантов. При этом алгоритм дает точное решение за исключением тех случаев, когда один из коэффициентов p_{ij} существенно отклоняется от среднего значения p_i^*.

Рассмотрим основные шаги алгоритма, определяющие его сущность.

Все элементы сортируются по важности: $w_{i+1} > w_i$.

Среди элементов пространства H выделяются задачи: $Z = \bar{V}_z \times H$.

АУТСКИЙ

Исключаются из рассмотрения те задачи, важность w_i которых больше важности w_k задачи z'_k. Определяется $i^{\max}$ – верхняя граница для рассмотренных задач.

Для каждой задачи рассчитывается среднее использование ресурсов p_i^*, $i = 1,2,...,i^{\max}$.

Сокращение перебора при высвобождении ресурсов позволяет достичь удаления задач, определяемых с помощью таких простых критериев:

1) удаляются наименее важные задачи:

$D_i = 1$, если $\sum_{j=1}^{i} p_j^* < p_k^*,\ i = 1,2,...,i^{\max}$;

2) удаляется наименьшее количество задач:

$D_i = 1$, если $\sum_{j=i^{\max}}^{i} p_j^* < p_k^*,\ i = i^{\max},...,1,2$.

3) удаляются наиболее ресурсоемкие задачи с минимальной важностью. Для этого задачи предварительно сортируются по значению $q_i = p_i^* / w_i$, так, чтобы выполнялось условие $q_{i+1} < q_i$, а потом проводится выбор задач по второму критерию.

После определения задач, которые в принципе могут быть удалены, сравниваются объемы высвобождаемых ресурсов для каждого из ресурсов и определяется ресурс с максимальным дефицитом $\max\Delta_j$. Определяется набор задач, которые блокируются для освобождения ресурсов по одному из приведенных выше критериев, и заменяется p_j^* значением j-го ресурса.

Рассмотрим, каким образом СУИ, имеющая многоуровневую архитектуру и использующая агентскую технологию, решает задачу перераспределения ресурсов.

Каждый региональный сервер СУИ управляет распределением ресурсов в зоне своей ответственности, руководствуясь политиками, поступающими от центрального сервера СУИ, и управляя агентами, установленными на РОУ. В случае, когда региональный сервер получает запрос на выделение ресурса, находящегося в компетенции другого регионального сервера, происходит их взаимодействие через серверы СУИ высших уровней иерархии.

Обобщенный алгоритм работы сервера A^m m-го уровня можно описать следующим образом.

Шаг 1. Сервер A^m получает запрос на выделение ресурсов для задачи z'_k.

Шаг 2. Сервер A^m разделяет множество необходимых ресурсов R_p из множества сервера R_a, на два подмножества R_e и R_z:

$$\begin{aligned} R_e &= (R_p \setminus R_{es}) \cap R_a; \\ R_z &= (R_p \setminus R_{es}) \setminus R_a, \end{aligned} \qquad (7.55)$$

где R_e – множество ресурсов, закрепленных за данным сервером; R_z – ресурсы, находящиеся в ведении других серверов СУИ; R_{es} – ресурсы, зарезервированные узлами нижних уровней.

Шаг 3. Сервер A^m определяет возможность выделения ресурсов множества R_e с учетом существующей на данный момент политики управления. Если все

ресурсы могут быть выделены, то они выделяются. В противном случае сервер A^{m+1} получает отказ. Все выделенные ресурсы идентифицируются в R_{es}:

$$R_{es} = R_{es} + R_e. \tag{7.56}$$

Шаг 4. Если хотя бы один из ресурсов не закреплен за сервером, т. е. $R_z \notin \varnothing$, то посылается запрос серверу более высокого уровня иерархии.

Шаг 5. Если задача z'_k принадлежит множеству данного сервера, то задача начинает выполняться, ей выделяются ресурсы и алгоритм завершается.

Шаг 6. Если все ресурсы выделены $R_{es} = R_p$, тогда выделить зарезервированные ресурсы $R_{es} \cap R_a$ и отправить запрос на выделение зарезервированных ресурсов серверу A^{m-1}.

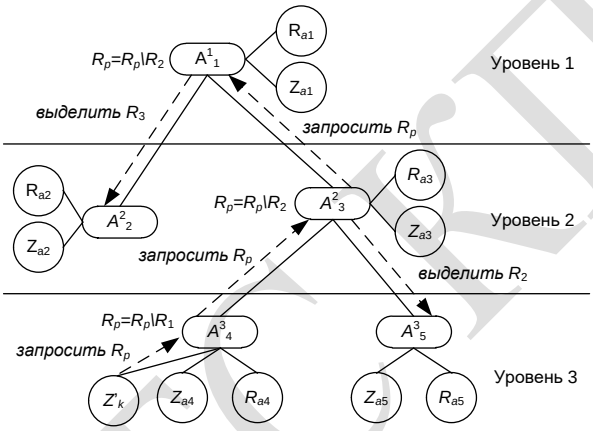


Рис. 7.6. Пример взаимодействия серверов СУИ с трехуровневой иерархией

На рис. 7.6 изображен пример взаимодействия серверов СУИ с трехуровневой иерархией. Задача z'_k требует множество ресурсов R_p, которое включает ресурсы R_1, R_2, R_3, т. е. $R_p = \{R_1, R_2, R_3\}$, однако ресурсы находятся в ведении серверов разных уровней иерархии $R_1 - A^3_4$, $R_2 - A^3_5$, $R_3 - A^2_2$.

Таким образом, разработана модель распределения ресурсов ИТ-инфраструктуры в случае появления новых задач, если они требуют для своего выполнения долю ресурсов, которые уже используются другими задачами. Перераспределение ресурсов осуществляется в иерархических СУИ с учетом важности выполняемых задач, важности новых задач и необходимых для их выполнения ресурсов.

7.2.4. Модель управления перераспределением ресурсов ИТ-инфраструктуры при изменении значимости бизнес-процессов

Эффективное поддержание современных систем бизнес-процессов нельзя удовлетворить за счет простого доступа к ресурсам ИТ-инфраструктуры, когда всем приложениям предоставляются равные права. Они предпочитают дифференцированный подход к выделению ресурсов требуемых объемов и качества, а также индивидуального обслуживания с предоставлением разных прав доступа.

Все бизнес-процессы имеют различную потребность в ресурсах ИТ-инфраструктуры, причем из экономических соображений задействуются одни и те же ресурсы для поддержания ряда бизнес-процессов. При этом требуемые объемы ресурса, а также необходимые и запрашиваемые условия предоставления ресурса отдельным приложениям могут иметь существенную динамику, независящую от других приложений. В то же время, ресурсы ИТ-инфраструктуры ограничены, а при одновременном обращении к общим ресурсам возникают конфликты, для разрешения которых средства управления ресурсом не всегда могут отдать предпочтение наиболее важному приложению по причине отсутствия информации о значимости приложений.

В процессе повседневной деятельности предприятия значимости бизнес-процессов могут изменяться. Эти изменения могут быть обусловлены как факторами, проявляющимися на организационном уровне, обусловленными трансформацией бизнес-целей предприятия, изменениями конъюнктурой рынка, форс-мажорными обстоятельствами и др., так и результатами оперативной диагностики ИТ-инфраструктуры, выявившей отказы фрагментов ИТ-инфраструктуры, делающих невозможным информационную поддержку отдельных бизнес-процессов. В этих и множестве других случаев происходит изменение значимости бизнес-процессов и, следовательно, приоритетов поддерживающих их ИТ-процессов, что в свою очередь влечет за собой необходимость перераспределения ресурсов ИТ-инфраструктуры.

Изменения значимостей бизнес-процессов могут происходить динамично и носить стохастический характер, например, перерывы или отсутствие связи на стороне ОТС, которыми активно пользуется предприятие, или, наоборот, быть предсказуемыми и планируемыми, например, на время проведения интернет-аукциона предприятию-организатору может потребоваться расширить полосу пропускания канала доступа в интернет для процессов поддержки аукциона за счет ограничения доступа для других процессов.

В работе [114] предложена модель управления перераспределением ресурсов ИТИ при изменении значимости бизнес-процессов. В этом случае при управлении ИТ-инфраструктурой СУИ должна руководствоваться значимостью всех бизнес-процессов и, соответственно, поддерживающих их приложений, определять приоритеты выполняемых задач, оценивать их потребности в общих ресурсах ИТ-инфраструктуры и осуществлять перераспределение ресурсов при появлении новых задач и при изменении значимости бизнес-процессов.

Связь между задачами в СУИ, способной оперативно реагировать на изменение значимости бизнес-процессов, приведена на рис. 7.7.

Задачи управления бизнес-процессами, определение значимости новых задач и приоритетов процессов решаются на IV уровне ИТ-инфраструктуры, где осуществляется комплексная автоматизация бизнес-процессов предприятия. Задачи управления ИТ-инфраструктурой, анализа состояния ИТ-инфраструктуры решаются СУИ на всех уровнях ИТ-инфраструктурой (см. рис. 1.2). Реализация команд управления при распределении и перераспределении ресурсов происходит на I и II уровнях.

Задачами анализа в частности являются получение информации о качестве ИТ-услуг, определение возможности предоставления ресурсов с требуемыми параметрами. Результаты измерения параметров качества ИТ-сервисов должны быть выражены в терминах бизнеса. Это сложная, но чрезвычайно важная задача

позволяет, например, трансформировать данные о продолжительности отсутствия доступа к сети интернет в финансовые потери от сокращения объемов продаж интернет-магазина.

Ориентированный на бизнес-процессы подход к управлению ИТ-сервисами, предложенный компанией IDC [31], включает в себя четыре этапа:

- идентификация всех бизнес-процессов и установление требований со стороны этих процессов к ИТ-сервисам;
- проецирование бизнес-процессов на элементы ИТ-инфраструктуры с определением технических требований к технологическим элементам;
- определение метрик, позволяющих проверить соответствие качества ИТ-услуг требованиям бизнес-процессов;
- контроль качества ИТ-услуг в заданных метриках и составление различной отчетности, позволяющей оценить соответствие уровня обслуживания, предоставляемого ИТ-инфраструктурой, потребностям бизнеса.

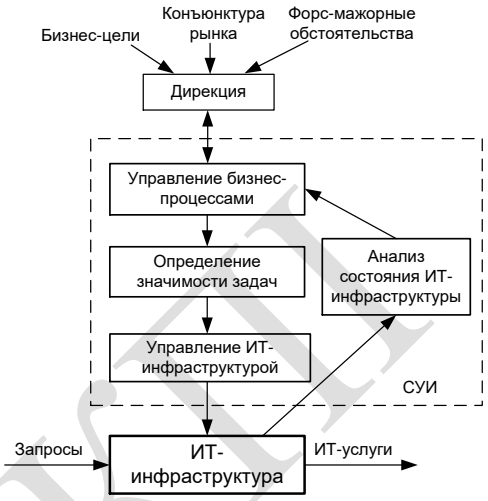


Рис. 7.7. Взаимосвязь задач, решаемых СУИ при изменении значимости бизнес-процессов

Недостаток этого подхода связан с тем, что рассматриваются только статические ситуации и предполагается, что соотношения значимостей бизнес-процессов не изменяются, так же как остаются неизменными технологические, структурные и функциональные связи бизнес-процессов и ИТ-инфраструктуры.

В реальных ситуациях статическая модель годится только для решения части задач планирования или управления и не подходит для решения задач, связанных с перераспределением общих ресурсов ИТ-инфраструктуры с целью адаптации к изменяющимся обстоятельствам ведения бизнес-процессов или условиям функционирования ИТ-инфраструктуры.

Предлагаемая модель управления перераспределением ресурсов ИТ-инфраструктуры учитывает изменчивость значимости бизнес-процессов и ограниченность ресурсов ИТ-инфраструктуры.

Сначала на основании значимости бизнес-процессов определяются приоритеты поддерживающих их приложений и решаемых задач. На основе этой информации обновляется политика управления, которая хранится в центральном сервере СУИ и реплицируется в региональные серверы СУИ. При изменении политики переопределяются приоритеты выполняемых задач и устанавливаются приоритеты новых задач. Изменение приоритетов выполняемых в данный момент задач может повлиять на их успешное завершение – выполнение задач может быть немедленно прекращено или отложено. Что касается новых задач, то в зависимости от их приоритета им может быть отказано в предоставлении ресурса. Все это определяется политикой управления, а также реальной ситуацией с ресурсами ИУС.

Определяются ресурсы, задействованные выполняемыми задачами, и устанавливаются объемы ресурсов, в которых нуждаются новые задачи. После чего происходит перераспределение ресурсов с учетом приоритетов задач и объемом задействованных и необходимых ресурсов при условии их ограниченности.

Выполним необходимые формальные построения.

Для исходных данных (7.38)–(7.40) $H = \{h_i\}$, $Z = \{z_i\}$, $R = \{r_i\}$, $i = \overline{1,n}$ зададим текущие значения $w_1,\ldots,w_n$ коэффициентов важности ресурсов и задач. Эти коэффициенты определяются политиками управления, а при изменении политик претерпевают изменения.

Зададим матрицу использования ресурсов P в виде (7.41) и векторов $\bar{X}$ и $\bar{D}$, элементы которых x_j, $j = \overline{1,n}$ и d_j, $j = \overline{1,n}$ определяются из выражений (7.42) и (7.43) соответственно.

Обеспечение максимальной эффективности функционирования свяжем с выполнением критерия (7.2), требующего достижения максимума суммарной важности задач, которые выполняются и планируются к выполнению с учетом новых значений коэффициентов важности, корректируемых при изменении политик управления.

С учетом ограничения (7.47) количество r_j свободных ИТ-ресурсов определяется в соответствии с выражением (7.48).

При появлении новой задачи z'_k, $k = \overline{1,n}$, для которой предварительно выполнялось условие $x_k = 0$, или повышении приоритета задачи z'_k, $k = \overline{1,n}$ до уровня, требующего ее выполнения с учетом новой политики, несмотря на отсутствие свободных ресурсов, необходимо высвободить ресурсы для выполнения z'_k.

Для каждого ресурса r_j, необходимого задаче z'_k, проверяется выполнение условия (7.49). Если условие (7.49) выполняется, то в ИТ-инфраструктуре достаточно свободных ресурсов для решения задачи z'_k и перераспределения ресурсов не требуется. Если в ИТ-инфраструктуре недостаточно незадействованных ресурсов для задачи z'_k, то СУИ высвобождает ресурсы, занятые менее важными задачами. Высвобождение ресурсов осуществляется с учетом критериев (7.51)–(7.53).

Для решения задач управления по высвобождению ресурсов с учетом критериев (7.51)–(7.53) можно быстро приспособить генетические алгоритмы.

Использование параметра p_i^*, определяемого в соответствии с выражением (7.54), позволяет использовать такие три варианта разрешения конфликта в процессе выделении ресурсов задаче z'_k, если эти ресурсы используются другими задачами:

1) если $p_i^* = 0$, то задача z_i использует ресурсы, на которые не претендует задача z'_k;

2) если $p_i^* \ll p_k^*$, то удаление z_i не повлияет на решение проблемы выделения ресурсов задаче z'_k;

3) в остальных случаях задача z_i использует достаточные объемы тех же ресурсов, на которые претендует z'_k.

Таким образом, СУИ, осуществляя выделение ресурсов для новой задачи z'_k и перераспределение ресурсов между существующими задачами z_i при изменении политик, должна выполнять поиск задач, подлежащих удалению, среди тех исполняемых задач, которые могут высвободить достаточный объем ресурсов.

Архив КГТУ

Для решения задача перераспределения ресурсов можно использовать разработанный выше алгоритм, который запускается, когда возникает новая задача или происходит перераспределение приоритетов выполняемых задач.

Таким образом, разработана модель управления перераспределением ресурсов ИТ-инфраструктуры при появлении новых задач, когда для своего выполнения они требуют части ресурсов, которые уже используются другими задачами или процессами. Модель учитывает важности всех задач и процессов, как тех, которые выполняются, так и новых, а также возможность изменения во времени приоритетов задач, обусловленных изменением значимости бизнес-процессов, которые они поддерживают. Использование предложенной модели в СУИ позволит осуществлять эффективное распределение и перераспределение общих ресурсов при возникновении новых задач и изменении значимостей бизнес-процессов.

ВЫВОДЫ К ГЛАВЕ 7

1. Разработан комплекс моделей и методов управления ограниченными ресурсами корпоративной ИТ-инфраструктуры.

2. Предложен процессно-ресурсный подход к управлению ограниченными ресурсами и разработана соответствующая технология управления Показано, что в случае дефицита ресурсов эффективность выполнения бизнес-процессов может быть обеспечена тогда, когда ресурсами обеспечиваются более важные бизнес-процессы за счет менее важных. Разработанный комплекс моделей и методов отличается учетом важности бизнес-процессов, что позволяет эффективно распределять и перераспределять ограниченные ресурсы при изменении важности бизнес-процессов для достижения максимальной суммарной эффективности выполнения основных из них.

Аутсорсинг

ГЛАВА 8

УПРАВЛЕНИЕ РЕСУРСАМИ И НАГРУЗКОЙ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

Глава посвящена разработке моделей и методов управления распределением ресурсов ЦОД и ресурсов корпоративной ИТ-инфраструктуры в условиях избыточности ресурсов. Управление ЦОД осуществляется на всех уровнях ИТ-инфраструктуры, но основную роль с точки зрения распределения нагрузки и оптимизации вычислительных ресурсов играет уровень вычислительных ресурсов.

8.1. АНАЛИЗ ПРОБЛЕМ УПРАВЛЕНИЯ РЕСУРСАМИ И НАГРУЗКОЙ ЦОД ПРИ ВИРТУАЛЬНОМ ХОСТИНГЕ

8.1.1. Актуальность задачи управления ресурсами ЦОД

В настоящее время прослеживается тенденция перехода от рассредоточенной обработки информации вычислительными мощностями подразделений и филиалов предприятий к централизованной обработке в корпоративных или коммерческих ЦОД – отказоустойчивых централизованных системах, обеспечивающих автоматизацию бизнес-процессов с высоким уровнем производительности и качества представленных услуг [158, 255]. Концентрация корпоративных вычислительных ресурсов в ЦОД позволяет не только проводить сложные научные и финансовые вычисления, консолидировано обрабатывать корпоративную информацию предприятий с территориально рассредоточенными филиалами, но и сократить расходы на коммуникационные услуги, обслуживание серверов, программного обеспечения и других компонентов. Свойства и преимущества информационно-коммуникационной инфраструктуры ЦОД наилучшим образом проявляются при организации перспективных ИТ. Например, ЦОД является системообразующим началом облачных технологий [161].

Концепция ЦОД используется многими корпорациями преимущественно для обеспечения доступа большого количества пользователей к ресурсам (сервисам, приложениям, вычислительным мощностям, данным и др.).

В условиях информатизации бизнеса возрастают требования пользователей к уровню сервиса, управляемости, надежности, доступности и масштабируемости ИТ-инфраструктуры. Это усложняет управление ИТ-инфраструктурой. В тоже время создание ЦОД для компаний малого и среднего бизнеса требует значительных средств, эффективной поддержки ИТ-инфраструктуры, наличия в штате высококвалифицированных специалистов. Кроме того, эффективное использование ЦОД требует решения ряда задач, прежде всего создания условий для функционирования информационно-вычислительных мощностей ЦОД, управления ресурсами, обеспечения надежности и безопасности. Поэтому все

больше компаний вкладывают средства в создание системы ЦОД для обслуживания пользователей других компаний, приобретая статус хостинговой компании [158, 159]. Для хостинговых компаний проблема создания и обеспечения эффективного функционирования ЦОД приобретает особую актуальность. Компании, предоставляющие услуги хостинга, вкладывая средства, надеются на получение прибыли. Поэтому они стремятся уменьшить эксплуатационные затраты ЦОД, снизить стоимость обслуживания пользователей и сократить другие расходы.

Выполнение требований пользователей к доступности и целостности данных, безопасности, надежности, масштабируемости, а главное – к качеству ИТ-услуг с наименьшими затратами является основной проблемой, которую необходимо решить при управлении ЦОД. Это сложная комплексная проблема, которую последовательно разбивают на ряд менее сложных проблем, наиболее важной из которых является задача управления ресурсами и нагрузкой ЦОД с целью удовлетворения потребностей пользователей при одновременном сокращении расходов на содержание ЦОД.

В работах [104, 144, 148, 154, 160] предложен подход к управлению ресурсами и нагрузкой ЦОД, на основе которого создана современная СУИ. Так, в работе [160] предложены модели и методы управления распределением ресурсов и нагрузкой ЦОД при виртуальном хостинге. Формализация проблемы привела к задачам булевого программирования – с довольно неудобными выражениями для некоторых ресурсных и технологических ограничений.

В данной главе разработаны модели и методы управления распределением ресурсов и нагрузкой ЦОД, в которых использованы преимущественно непрерывные переменные, что позволяет естественно выразить ресурсные и технологические ограничения и приводит к интересным случаям смешанных задач математического программирования. Соответствующие методы и алгоритмы решения этих задач рассмотрены в главе 10.

Для эффективного управления распределением ресурсов и нагрузки нужны гибкие решения, основанные на оценке и прогнозировании состояния ресурсов, объемов нагрузки, заключающиеся в правильной балансировке нагрузки и эффективном распределении информационно-коммуникационных ресурсов. Для систематического принятия рациональных решений необходимы инструментарий и комплекс методик решения задач управления ИТ-инфраструктурой. Их создание составляет важную научно-практическую проблему, решение которой требует глубокого понимания процессов, происходящих в хостинговых компаниях, функционирования ИТ-инфраструктуры ЦОД, четкой постановки задач исследования, разработки математических моделей и соответствующих методов решения задач и, наконец, реализации упомянутых выше инструментария и методик.

Необходимость наращивания вычислительных мощностей в связи с возрастанием требований пользователей влечет за собой существенные затраты на дополнительное оборудование, коммуникационные услуги, ПО и др. Поэтому актуальной становится задача управления ресурсами ЦОД с целью удовлетворения потребностей пользователей за счет имеющегося оборудования при одновременном сокращении расходов на содержание ЦОД без потери качества обслуживания.

На данный момент существует ряд подходов к решению задач и готовых решений по управлению распределением ресурсов и нагрузки, большинство из которых для поиска решений используют эвристические алгоритмы различной степени

сложности, матричные методы и другие аналитические подходы. Одним из методов, используемых для решения подобного класса задач, является метод на основе ГА, применение которого в контексте данной проблемы почти не исследовано.

Целесообразность использования ГА для решения задач распределения и управления ресурсами частично доказана в работах [143, 158]. Там же сформированы проблемы, требующих решения для полноценного использования ГА при управлении ресурсами ЦОД. Также важной является задача адаптации ГА для решения других задач из сферы ИТ-управления.

Таким образом, повышение эффективности использования ресурсов ЦОД является актуальной задачей.

8.1.2. Анализ ЦОД как объекта управления

Преимущества ЦОД. ЦОД традиционно представляет собой отдельное здание или совокупность помещений с мощной вычислительной, коммуникационной и инженерной инфраструктурой. Инженерная инфраструктура включает в себя инженерные конструкции для размещения оборудования, а также системы энергоснабжения, кондиционирования, пожаротушения, безопасности с обязательными элементами мониторинга и управления.

ЦОД предназначен для централизованной обработки данных, автоматизации бизнес-процессов, предоставления услуг большому количеству клиентов.

Не только значительная численность клиентов, но и большое количество серверов, огромные объемы обрабатываемых и хранимых данных, жесткие требования к зафиксированным в SLA показателям качества услуг, которые необходимо поддерживать при максимальных нагрузках, предъявляют к ИТ УИТИ повышенные требования эффективности управления ИТ-инфраструктурой. В условиях высокой плотности вычислительных ресурсов, жестких требований к информационной защите, повсеместного использования виртуализации, необходимости переключения на резерв без потери данных, резервирования каналов доступа пользователей к ресурсам ЦОД и других технологий [65, 335, 350] обеспечение эффективного использования ресурсов требует от разработчиков ИТ УИТИ решения множества сложных многокритериальных оптимизационных задач.

В пользу решения о строительстве или использовании ЦОД положительно влияют следующие факторы:

- необходимость консолидации большого количества разноплановых информационно-вычислительных ресурсов;
- существенное увеличение объемов обрабатываемой информации;
- большое количество клиентских подключений;
- существенное увеличение пользователей информационных ресурсов;
- возрастающий интерес к ЦОД не только крупных, но и мелких компаний, как к средству реализации надежных информационных систем.

Основными преимуществами использования ЦОД являются:

- снижение стоимости эксплуатации ИУС;
- высокая надежность ИУС;
- простота организации консолидированной обработки корпоративной информации;
- возможность оптимизации использования информационно-вычислительных ресурсов;
- гибкость и масштабируемость ИУС.

Классификация ЦОД. По принадлежности и использованию ЦОД делятся на:

• корпоративные, которые обслуживают только сотрудников корпорации и обеспечивают автоматизацию бизнес-процессов корпорации, владеющей ЦОД;

• коммерческие (хостинговые), которые обслуживают множество сторонних организаций, предоставляя вычислительные ресурсы для автоматизации их бизнес-процессов. Хостинговые ЦОД осуществляют выделение ресурсов или места и условий для размещения серверов, предоставляют выделенный и виртуальный хостинг, соответствующие услуги администрирования, резервного копирования данных клиентов и другие информационные услуги.

Для дальнейшего исследования наибольший интерес представляют коммерческие ЦОД, поскольку их использует большое количество сторонних организаций, предпочитающих аренду ресурсов и услуг, предоставляемых ЦОД вместо собственных ИТ-инфраструктур. При этом прибыль коммерческих ЦОД напрямую зависит не только от объема продаж услуг, но, в большей степени, и от эффективности использования ресурсов. Поэтому основное внимание в данной главе уделено разработке методов управления выделением ресурсов при предоставлении услуг хостинга – услуг по предоставлению места для размещения информации заказчика на серверах ЦОД с постоянным доступом по сети интернет.

Анализ задач управления хостингом. Увеличение количества ИТ-услуг, необходимых для успешного ведения бизнеса, повышение требований пользователей к качеству этих услуг вызывает соответствующее ужесточение требований к управляемости, надежности, доступности и масштабируемости ИТ-инфраструктуры. Для выполнения этих требований необходимо иметь сложную дорогостоящую СУИ и квалифицированный обслуживающий персонал. Не каждая организация способна выделять значительные средства для эффективной поддержки ИТ-инфраструктуры, как это происходит в ЦОД, поэтому все больше организаций становятся клиентами хостинговых компаний.

Эффективная эксплуатация и управление ИТ-инфраструктурой ЦОД возможны только при наличии организационной структуры, оснащенной современным инструментарием и комплексом методик сбора, анализа информации, принятия решений и управления их реализацией. Создание указанных инструментария и комплекса методик требует глубокого понимания процессов, происходящих в хостинговых организациях, функционирования ИТ-инфраструктуры, четкой постановки конкретных задач исследования, разработки математических моделей и соответствующих методов решения задач и, наконец, реализации упомянутых инструментария и методик в структуре СУИ.

На рис. 8.1 изображена стратификация, отображающая совокупность признаков и факторов, используемых при постановке задач распределения ресурсов и нагрузки ЦОД.

Разработка всего комплекса моделей и методов управления ресурсами ЦОД, покрывающего всю совокупность признаков и факторов, специфицирующих задачи управления, представляет собой трудоемкую задачу. Поэтому ниже разработан комплекс моделей распределения и управления нагрузкой и ресурсами ЦОД для технологии виртуального хостинга, представляющей наибольший практический интерес. При этом учитываются требования к ЦОД как объекту управления, сгруппированные по уровням архитектурной модели ЦОД [65, 335, 350].

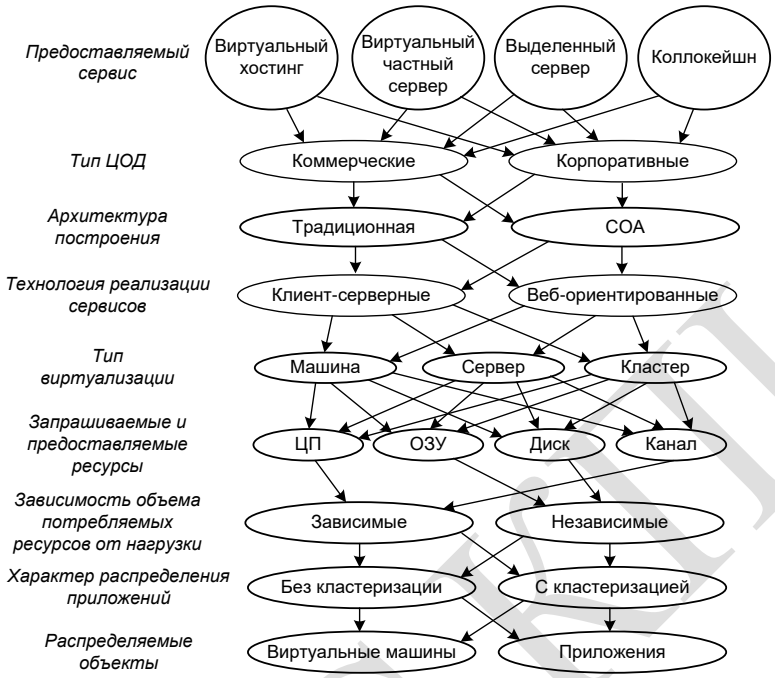


Рис. 8.1. Стратификация совокупности признаков, используемых для постановки задач распределения ресурсов и нагрузки ЦОД

Проблема управления ресурсами кластеров рассмотрена в работе [182]. В работах [273, 384] решается задача распределения ресурсов, оптимального с точки зрения равномерности загрузки серверов. Алгоритмический подход к распределению ресурсов представлен в работе [259]. В работах [209, 213, 229, 337, 341, 383] имеются решения отдельных задач по управлению ресурсами. В работах [173, 181, 231, 273, 374, 377, 384, 391, 392] проблема распределения ресурсов и управлением нагрузкой решена для виртуального хостинга. В то же время отсутствуют методы и модели управления распределением ресурсов и нагрузки ЦОД, ориентированные на поддержание согласованного уровня услуг. Не известны реализованные механизмы активного планирования и управления распределением ресурсов серверов между ВМ и нагрузки пользователей с учетом их важности.

Эффективности использования ресурсов ЦОД способствует технология виртуализации [238, 288], при этом существенные финансовые потери обусловлены нерациональным распределением ВМ по физическим серверам, когда при распределении ВМ используется метод последовательного заполнения или метод экспертных оценок, предполагающий распределение ВМ вручную [238]. В то же время, максимально плотное заполнение серверов позволяет сократить затраты на электроэнергию и обслуживание. Поэтому особую значимость приобретает разработка методов рационального распределения ВМ в ЦОД.

Оптимальному распределению ресурсов ЦОД в условиях их избытка посвящено множество работ. Так, в работах [147, 228, 363, 365, 367, 368] предложены модели и методы управления распределением ВМ при виртуализации серверов, а в работе [147] эта же задача решается в условиях живой миграции ВМ. В рабо-

тах [160, 364] рассматривается проблема управления ресурсами в условиях изменчивости среды функционирования, работах [159, 363] – управление нагрузкой и ресурсами ЦОД при выделенных серверах. Кроме того, в работе [158] рассматривается задача управления нагрузкой и ресурсами ЦОД при виртуальном хостинге, а в работе [146] – задача управления ресурсами в системах с серверной виртуализацией с учетом важности приложений. В работах [103, 141, 143, 228, 363, 364, 365, 366, 368] предложены модели и алгоритмы управления ресурсами и нагрузкой ЦОД. Ниже приведены некоторые из предложенных в этих работах моделей и методов управления ресурсами в условиях их избытка.

Комплекс постановки и исследования задач управления ресурсами ЦОД, которые необходимо решить при разработке ИТ УИТИ для ЦОД, зависит от полноты учета многих факторов, к которым прежде всего относятся:

1) модели хостинга (виртуальный хостинг, частные виртуальные или выделенные серверы, коллокейшн и др.);

1) архитектура построения программных систем;

2) технологии, на которых базируются технические решения ЦОД;

3) методы и модели управления хостингом и др.

Рассмотрим эти факторы подробнее.

Архитектура построения систем. Основными архитектурами являются традиционная архитектура и СОА. Традиционная архитектура основывается на абстракциях приложений и элементах их построения. При этом преимущественно используется трехуровневая архитектура с сервером БД, сервером приложений и клиентом. Идеология построения СОА основывается на абстракции сервисов. Цель разработки СОА заключается в создании комплекса приемлемого уровня абстракций сервисов, рассчитанных на многократное использование многими приложениями, а не жестко привязанных к одному из них. Это позволяет увеличить гибкость вычислительных процессов, уменьшить повторяемость кода функционала, возможность реализации платформенно независимых решений определенных задач.

Основные технологии реализации сервисов. Для масштабирования систем, гибкого распределения ресурсов между клиентами и балансировки нагрузки хостинговые компании используют клиент-серверные и веб-ориентированные технологии построения и организации доступа к прикладному ПО, технологии виртуализации и кластеризации.

Клиент-серверные и веб-технологии построения и организации доступа к прикладному ПО предназначены для эффективной реализации функциональности прикладных программных систем, создания условий для эффективной реализации системы бизнес-процессов, поддерживающих деятельность бизнес-персонала компании.

В трехуровневых клиент-серверных технологиях процессы обработки информации делятся между сервером БД, клиентской (далее программа-клиент) и серверной (далее сервер приложений) частями. Сервер приложений обслуживает много приложений, каждое из которых обслуживает много клиентов, а один клиент может работать с многими приложениями. Клиенты общаются с сервером приложений, отправляя запросы, а получая ответы. За данным к серверу БД обращается обычно сервер приложений. Принципиальные преимущества этих технологий: снижение уровня требований к мощностям рабочих станций; сегментация логики путем перемещения части кода на вторичные серверы, распределение

процессорной нагрузки, повышение надежности и доступности; упрощения проблемы масштабирования; оптимизация серверов для выполнения конкретного круга задач, уменьшение непроизводительных расходов.

Веб-технологии построения и организации доступа к прикладному ПО близки к клиент-серверным технологиям, поскольку их появление было реакцией на те же проблемы и они имеют такое же назначение. В них процессы обработки информации бизнес-пользователей распределяются между тремя типами взаимодействующих программных компонентов – сервером БД, клиентом и веб-сервером приложений. Иногда для организации работы с пользователями выделяется отдельный веб-сервер, который управляет запросами к серверам приложений. Эти технологии обладают большинством преимуществ клиент-серверных технологий и, кроме того, имеют большую степень независимости основных компонентов программных систем. Однако по сравнению с клиент-серверными технологиями они менее безопасны.

Приложение клиент-серверных и веб-технологий обеспечивает снижение уровня требований к рабочим станциям, распределение процессорной нагрузки и оптимизацию серверов, масштабирование систем и повышение их быстродействия, уменьшение непроизводительных расходов.

Технологии виртуализации предназначены для абстрагирования от особенностей отдельных групп ресурсов. Данная технология предоставляют пользователю возможность объединить ресурсы в виртуальной ИТ-инфраструктуре нужной конфигурации и упростить управление этими группами ресурсов. Различают виртуализацию платформ, которая заключается в создании ВМ, и виртуализацию ресурсов, которая обобщает подходы по созданию ВМ и переносит их на все виды ресурсов – оборудование ИТ-инфраструктуры, пространство имен, сети и т. п. В общем случае ВМ – это программные абстракции, запускаемые на платформе реальных аппаратно-программных систем (хостов), которые эмулируют работу реального или виртуального аппаратного обеспечения, используя реальные ресурсы хостовой машины, в то время как ВМ является гостевой машиной. Есть несколько видов виртуализации, которые отличаются полнотой симуляции аппаратного обеспечения: виртуализация (эмуляция), которая разветвляется на полную, аппаратную и частичную; паравиртуализация; контейнерная виртуализация (виртуализация уровня ОС).

Виртуализация обеспечивает снижение операционных затрат на поддержку парка серверов, расположение нескольких серверных приложений на одном физическом сервере, повышение доступности ИТ-сервисов и другие преимущества.

Технологии кластерного хостинга нацелены на решение проблем типовых распределенных ИТ-инфраструктур. Кластеры, как правило, состоят из стандартных компьютеров – узлов кластера, причем на каждом из них установлено собственное ядро ОС, а физический накопитель данных (массив дисков) не используется совместно.

Совместная эксплуатация нескольких систем улучшает отношение цены к качеству, обеспечивает достижение высокой готовности и производительности, а также балансировку нагрузки. Решения кластерного хостинга позволяют распознавать отказы узлов и передавать отказавшего задания узла другому узлу, используя сеть хранения данных. Это дает возможность пользователям управлять безопасностью, балансировкой нагрузки и ресурсами.

После получения аккаунта пользователь может заказывать ресурсы хостинга в пределах физических ограничений платформы. Причем при кластерном хостинге пользователи могут использовать ресурсы нескольких серверов, а их приложения распределяются в реальном времени за счет виртуализации ресурсов, предоставляемых физическими серверами. Изменения в аккаунте пользователя распространяются без задержки на каждый сервер кластера. Комбинированное управление в виде динамической балансировки запросов между несколькими приложениями и добавления серверов в кластер защищает пользователей от спада производительности и позволяет справляться с большими объемами трафика.

Однако даже крупные клиенты требуют только часть серверного пула. Управление емкостями осуществляется путем добавления дополнительных серверов в кластер без негативного влияния на существующих пользователей, причем изменения в потребностях ресурсов или настройках распространяются мгновенно на все серверы.

Модели хостинга. Выделяют четыре основных модели хостинга [143, 158]:

1) виртуальный хостинг – до $n \cdot 10^4$ аккаунтов пользователей могут размещаться на одном сервере, соревнуясь за серверные ресурсы. При этом пользователи получают дисковое пространство на сервере для размещения собственной информации и борются за серверные ресурсы, необходимые для работы их функционала. В связи с этим пользователи используют общие ресурсы, такие как пропускная способность канала, ОЗУ, частота процессора, что негативно влияет на работу хостинга в условиях высокой нагрузки. Благодаря небольшой стоимости этот вид хостинга получил наибольшее распространение;

2) виртуальный выделенный сервер (VDS) – это технология совместного использования ресурсов на основе виртуализации, когда клиенты арендуют ресурсы на сервере и эти ресурсы резервируются для них. Этот вид хостинга является наиболее дорогим и наиболее качественным видом виртуального хостинга. В этом случае каждому пользователю выделяется определенный сегмент серверной мощности хостинга. Общая картина выглядит такой же, как и при работе с выделенным сервером, однако в этом случае на одном сервере могут находиться десятки VDS. Этот вид хостинга подходит для серьезных проектов, чьи запросы, учитывая повышенную потребность в ресурсах, уже не может удовлетворить обычный виртуальный хостинг. Минусы этого хостинга заключаются в том, что полностью отделить один виртуальный хостинг от другого не всегда представляется возможным. К тому же его мощности так же ограничены, как и в случае виртуального хостинга. Более того, хостинг-провайдеры иногда злоупотребляют возможностями оборудования, сознательно размещая на серверах больше VDS, чем можно позволить в условиях повышенной нагрузки, аргументируя это тем, что пользователи не будут использовать полностью мощности VDS;

3) выделенный сервер – клиенты арендуют физические серверы, которые резервируются для них, чем обезопасивают клиента от перегрузки серверов, плохой производительности и несовершенных программ его соседей по хостингу. Этот вид хостинга используют для решения нестандартных задач или поддержки сложных веб-приложений, которые требуют для реализации своих задач больших мощностей сервера. Такой вид хостинга привлекателен для небольших хостинг-компаниями, использующих арендованные серверные мощности для организации виртуального хостинга. Стоимость такого хостинга больше, чем у двух предыдущих моделей;

4) коллокейшн (Collocated Hosting). Этот вид хостинга заключается в физическом размещении оборудования клиента (или арендованного у хостинг-провайдера) на территории хостинговой компании с подключением его к высокоскоростным каналам доступа в интернет. На это оборудование распространяется весь сервис поддержки работоспособности – источники бесперебойного питания, системы кондиционирования и пожаротушения, а также ряд оговоренных услуг связанных рабочим процессом – изменение конфигурация системы, резервное копирование, мониторинг и др.

В данных исследованиях в качестве модели хостинга коммерческого ЦОД выбрана модель виртуального хостинга.

Особенности виртуального хостинга. Самой распространенной моделью хостинга является виртуальный хостинг, основанный на СОА, веб-технологиях, технологиях виртуализации и кластеризации. В этом случае большое количество пользователей используют услуги веб-приложений. Один ЦОД может поддерживать различные веб-серверы приложений, развернутых на физических серверах от $n \cdot 10$ до $n \cdot 100$. Экземпляр приложения A_i, $i = \overline{1, N}$ может быть запущен на любом физическом сервере, который удовлетворяет ресурсным требованиям приложения A_i, а также специальным требованиям, которые не могут быть изменены с помощью функционала СУИ, например, настройки сети, каналов связи, присутствия программных библиотек и т. п. Пользователи получают доступ к экземплярам приложений, не зная, на каких физических серверах они расположены. В этой среде веб-запросы пользователей привязываются к веб-серверам приложений и передаются конкретным экземплярам приложений для обработки. Экземпляры приложений, которые запускаются и поддерживаются веб-серверами приложений, обрабатывают эти запросы, и веб-серверы генерируют некоторое статическое или динамический содержимое.

Серверный парк должен поддерживать работу такого количества экземпляров приложений, которое способно обеспечить суммарные требования пользователей соответствующих приложений. Также могут иметь место упомянутые выше специальные требования. Поскольку физические серверы ЦОД часто не имеют достаточного объема вычислительной мощности и памяти, чтобы один из них мог поддерживать работу всех развернутых приложений A_i, $i = \overline{1, N}$, администраторы веб-серверов разворачивают приложения на свободных физических серверах. Физические серверы объединяют в кластер. Экземпляры одного и того же приложения, развернутые на нескольких серверах, образуют, таким образом, единое виртуальное приложение. Можно объединить физические серверы в несколько кластеров, но тогда нужно решить несложные технические задачи кластеризации серверов и разделения веб-приложения между кластерами. В этом случае формируется распределенная вычислительная среда, изображенная на рис. 8.2.

Эффективность функционирования такой среды можно увеличить за счет рационального размещения экземпляров приложений на физических серверах. Отсутствие жесткой привязки запросов пользователей к экземплярам приложений позволяет динамически распределять запросы между экземплярами соответствующего приложения с учетом загруженности физических серверов. Поскольку требованиям пользователей присущи изменения, при их накоплении в определенных объемах необходимо переходить к новому плану размещения приложений, выделяя для приложений дополнительные вычислительные ресурсы или, наоборот, ограничивая выделенные приложению ресурсы.

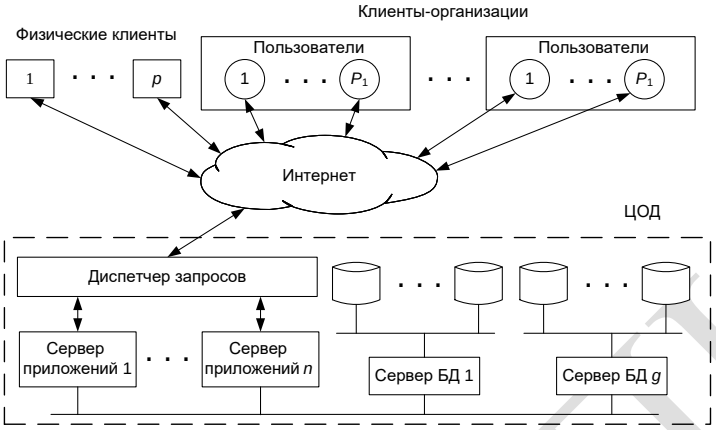


Рис. 8.2. Схема доступа к ресурсам при виртуальном хостинге

Если новый план размещения определен, то переход к нему заключается в настройке веб-серверов приложений, развертывании или свертывании приложений согласно новому плану, изменениях конфигурационных файлов и других настроек среды. Использование других механизмов балансирования, например, остановки и запуска веб-серверов приложений на физических серверах, где находятся экземпляры приложений, позволяет избежать потенциально больших затрат времени на операции развертывания или свертывания приложений, но существенно ограничивает гибкость решений, поскольку не позволяет изменять набор развернутых приложений на сервере приложений.

Значительное количество приложений с разными ресурсными требованиями и физических серверов с различными характеристиками превращают проблему рационального размещения экземпляров приложений в далеко не тривиальную задачу. Поэтому достижение преимуществ от внедрения рассмотренных выше технологий возможно лишь при условии решения ряда проблем, прежде всего оценивания и прогнозирования нагрузок и оптимизации выполнения запросов клиентов, управления ресурсами и нагрузкой и др. Поскольку процессы принятия решений становятся сложнее и требуют более развитого инструментария их поддержки, возникает необходимость разработки моделей, методов и средств мониторинга, анализа и управления информационно-вычислительными процессами ЦОД с учетом специфики этих процессов.

Большая часть существующих сегодня подходов и реализаций базируется на нескольких общих принципах построения и организации процесса управления, которые можно сформулировать следующим образом:

- управление ресурсами и нагрузками реализуется через сворачивание или развертывание экземпляров приложений на конкретных серверах. В случае использования виртуализации процессы свертывания и развертывания могут переноситься и на виртуальные машины;
- планирование распределения в большинстве случаев делится на два отдельных этапа: этап инициализации или стартового распределения, который выполняется на основе текущей статистики соотношений ресурсов серверов и требований приложений, прогнозирования нагрузок на определенный период работы, и

этап динамического распределения, который осуществляется в ходе функционирования системы путем ответной реакции на изменения потребностей пользователей;

• в структуре систем выделяются отдельные блоки, специализирующиеся на реализации конкретных этапов функционирования системы: блок сбора и анализа текущей статистики системы, блок принятия решений и формирования плана распределения, блок реализации решений, конечные функциональные блоки реализации распределения на каждом конкретном сервере и сигнализации изменений конфигурации;

• процессы построения планов распределения приложений по серверам в большинстве случаев базируются на использовании эвристик разного уровня сложности. В некоторых случаях использованы матричные и аналитические подходы к организации поиска распределения, построены направленные графы решений.

8.1.3. Анализ существующих решений управления ресурсами и нагрузкой

Несмотря на накопленный опыт создания и эксплуатации ЦОД, остаются нерешенными задачи управления: нагрузкой и ресурсами; резервированием, обеспечением безопасности; надежной работой ПО; технической поддержкой ИТ-инфраструктуры.

Достижение преимуществ, предоставляемых ЦОД, возможно лишь при условии решения проблем оценивания и прогнозирования нагрузок и оптимизации выполнения запросов клиентов, управления ресурсами и нагрузкой и др. Поскольку процессы принятия решений становятся сложнее и требуют более развитого инструментария их поддержки, возникает необходимость разработки моделей, методов и средств мониторинга, анализа и управления информационно-вычислительными процессами ЦОД с учетом специфики применяемых технологий.

Создание таких сложных систем как СУИ обусловило возникновение многих научных проблем. В ведущих научных центрах ученые интенсивно изучают проблемы распределения ресурсов, управления нагрузкой, балансировкой и др.

Распределение ресурсов, основанное на прогнозировании, в рамках единой аппаратной системы является хорошо исследованной темой. Значительные успехи достигнуты в разработке принципов распределения ресурсов [384]. Много подходов предложено для предполагаемого планирования ресурсов процессора и распределения сетевой пропускной способности, памяти, дискового пространства и общедоступных сервисов на одиночных серверах [148, 181, 273, 377].

Однако подходы, пригодные для управления ресурсами одиночных серверов, не всегда являются лучшими для управления группами взаимосвязанных серверов и кластерных систем. Кроме того, имеются такие специфические комбинации технологий ЦОД, для которых нет соответствующих решений, например, при распределении ресурсов между ЦОД.

Поэтому необходимо построить модели для управления ресурсами в указанных случаях. Необходимо такое комплексное решение для распределения и управления ресурсами ЦОД, которое можно было бы применять в любой ситуации, настраивая его на особенности среды применения.

Сегодня отсутствуют программные решения, функционал которых покрывает весь комплекс задач СУИ. Эффективность решений существенно зависит от моделей распределения нагрузки и ресурсов. Поэтому необходимо разработать

соответствующие модели и методы управления нагрузкой и ресурсами для различных условий функционирования ИУС.

В настоящее время существует ряд подходов к организации управления ресурсами и нагрузкой. Разработке этих подходов посвящено большое количество работ. Программная реализация разработанных методов воплощена в программных продуктах, используемых в сфере управления информационными структурами различных предприятий. Рассмотрим более подробно основные из методов управления ресурсами и нагрузкой, а также программные продукты, реализующие эти методы.

Подходы к организации управления ресурсами и нагрузкой ЦОД. Создание ИТ УИТИ требует решения ряда научных проблем, прежде всего управления распределением ресурсов, нагрузкой, балансировкой и многие другие. Проблема распределения ресурсов успешно решается в рамках единой аппаратной системы.

Комплекс моделей распределения и управления нагрузкой и ресурсами ЦОД для различных технологий хостинга предложен в работах [146, 148, 158, 159, 228, 363, 364, 365, 366, 367, 368], в частности для случаев:

- виртуального хостинга [158]: рассмотрены дискретный (ресурсы одного сервера используются для удовлетворения потребностей одного приложения) и непрерывный (обеспечение потребностей несколькими серверами, распределенные вычисления) варианты модели;
- выделенных серверов [159];
- серверной виртуализации [146].

В каждой из указанных работ проблема распределения и управления нагрузкой и ресурсами ЦОД решена путем реализации последовательно (при необходимости, итеративно) выполняемых этапов планирования, управления и диспетчеризации ресурсов и нагрузки.

Проблемы профилирования экземпляров приложений, развернутых на серверах, рассматриваются в работе [159], где предлагаются подходы по распределению ресурсов, резервирования определенного объема ресурсов для экземпляров определенных приложений, их закрепление за определенным экземпляром приложения на сервере, а также высвобождение.

Проблема управления ресурсами серверных кластеров рассмотрена в [182]. Изложены общие положения проблемы и предложены пути решения поставленных задач, формализованные в виде методик размещения приложений и распределения ресурсов серверных кластеров.

Значительные успехи достигнуты в разработке моделей распределения ресурсов в сетях [384]. Рассмотренный в этой работе подход основан на решении проблемы распределения дискретных ресурсов при наличии ограничений. Задача представлена в виде направленного графа, вершинами которого являются конкретные серверы, а ребра представляют собой взаимосвязи между серверами парка. Процесс распределения разделен на две компоненты: статическую и динамическую.

Статическая компонента выполняет формирование стартового распределения на основе исходных данных, статистики и прогнозов. При этом используется эвристический подход к решению задачи.

Динамическая компонента предназначена для распределения запросов, поступающих в процессе функционирования системы. На каждом шаге происходит поиск оптимального решения и его реализация путем перенаправления запросов

к нужному серверу. Подход был реализован для организации равномерной загрузки системы серверов при обеспечении поддержки множества сайтов, обслуживающих набор пользователей. При этом базовые аспекты, такие как разделение основного функционала на статическую и динамическую компоненты, механизмы управления и распределения нагрузки, могут быть использованы для решения других задач, в частности задачи управления распределением ресурсов при виртуальном хостинге.

Похожий на предыдущий подход рассмотрен в работе [273]. Поставленная в этой работе задача заключается в поддержании динамического распределения приложений на серверах, оптимального с точки зрения равномерности нагрузки. В работе рассматриваются одновременно два вида параметров серверов (зависящих от нагрузки и не зависящих от нее), а именно – процессорное время и ОЗУ. Сходство с предыдущей задачей состоит в организации процесса распределения, а именно в разделении процесса на начальный и динамический итеративные этапы. Введенное авторами ограничение касается скорости поиска решения на каждой отдельной итерации. Быстродействие системы управления должно соответствовать динамике поступления запросов от пользователей ЦОД. Поставленная задача рассматривается с учетом того, что она является NP-полной. Для решения этой задачи предложено построение графов поиска решения. Для реализации поиска используется эвристический подход. Если за определенное конечное время алгоритму удается найти лучшее решение в определенном направлении поиска по графу, то часть графа, в которой велся поиск, удаляется, а сам граф изменяет свою структуру. После выполненных преобразований поиск решения начинается снова.

Некоторые алгоритмические подходы к общей проблеме распределения ресурсов представлены в работе [259]. Приведены примеры реализаций различных задач по распределению конечного объема определенных ресурсов на определенные нужды. Приведены несколько примеров реализации задач по распределению информационных ресурсов на удовлетворение потребностей пользователей вычислительных систем.

Комплексный подход к организации управления группой серверов предложен в работе [270]. Авторами разработана структура СУИ, распределены роли между отдельными составляющими системы. Следует отметить четкую структурированность описанной компоненты управления размещением приложений на серверах, блок-схема которой приведена на рис. 8.3.

Среди введенных функциональных составляющих выделяются следующие:

- контроллер размещения – основной блок системы, который на основании получаемой статистики (отчетов об изменении конфигурации размещения приложений по серверам, о текущем размещении приложений и о прогнозах следующих запросов) вырабатывает новый план размещения;
- блок оценки требований приложений занимается сбором текущей статистики размещения приложений, поступления новых и прогнозирует возможные запросы от пользователей на определенный период. Проанализированная и определенным образом обработанная статистика предоставляется контроллеру размещения для поиска необходимой стратегии размещения на новый плановый период;
- блок фиксации распределения фиксирует развертывание или свертывание определенного приложения на определенном сервере. Он предоставляет отчет о процессах свертывания и развертывания блоку управления экземплярами приложений;

• блок управления экземплярами приложений на основе полученного от контроллера распределения плана размещения экземпляров приложений на серверах вырабатывает стратегию внедрения плана, реализует ее и отчитывается о результатах внедрения;

• блок исполнения стратегии развертывания на основе полученных от контроллера распределения команд реализует стратегию внедрения плана на каждом отдельном сервере.

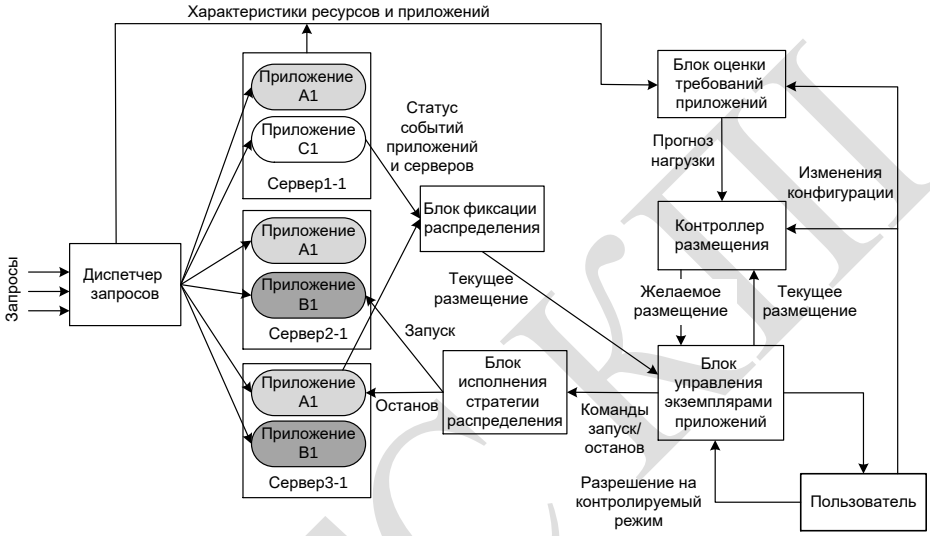


Рис. 8.3. Архитектура компоненты управления размещением приложений на серверах

В основу функционирования контроллера размещения при поиске оптимального плана положен эвристический подход.

В работах [337, 383] можно найти интересные решения по управлению ресурсами при организации видеосвязи и передачи видеоизображений или подходы к организации решения конкретных задач [209, 213, 229, 341]. Но управление группами серверов и кластерных систем требует дополнительных исследований. Имеются специфические комбинации технологий ЦОД, для которых нет соответствующих решений, например распределение ресурсов между ЦОД. Необходимо построить модели для управления ресурсами в указанных случаях. Кроме того, необходимо комплексное решение для распределения и управления ресурсами ЦОД, которое можно было бы применять в любой ситуации, настраивая его на особенности среды применения. Поэтому необходимо разрабатывать новые эффективные модели и методы управления нагрузкой и ресурсами в различных условиях функционирования ИТ-инфраструктуры ЦОД.

Обзор аналогов среди программных продуктов. Развитие ИТ и телекоммуникаций превратило ИТ-инфраструктуру в объект управления. Накопленный опыт эксплуатации больших распределенных ИУС привел к формированию представлений об ИТ-инфраструктуре как сложном объекте управления. Назначение новых нагрузок на отдельные серверы с соответствующими возможностями привело к значительному увеличению мощностей ЦОД. Изменчивость приложений, отсутствие специализированного инструментария планирования и

управления использованием ресурсов повлекли высокую стоимость владения, лицензий на ПО и администрирование.

Поэтому с целью увеличения гибкости, обеспечения возможности быстро перенацеливать серверные мощности под нужды приложений и снижение общей стоимости владения некоторые корпорации начали использовать пулы серверных ресурсов на основе виртуализации и кластеризации. Сложность таких сред обусловила появление проблем планирования, распределения и управления использованием ресурсов соответствующей сложности.

Основные производители компьютерного, телекоммуникационного оборудования и ИТ вышли на рынок с предложениями, предназначенными для решения этой проблемы [44, 236, 267, 334, 356, 357, 358, 360, 376, 389,]. Предложенные решения разработаны на основе ITIL, на которой базируется концепция ITSM. Базовыми функциями СУИ являются мониторинг, управление нагрузкой и ресурсами, администрирование ИТ-инфраструктуры. ITIL и ITSM исходят из позиций поддержки системы бизнес-процессов компании. Принципиальными положениями являются поддержка жизненного цикла ИУС, создание единой информационной среды, в которой централизованно поддерживаются процессы развертывания и эксплуатации ИТ-инфраструктуры.

Ведущие позиции занимают MS Data Center компании Microsoft и IBM IT Process Model, которую поддерживает линейка продуктов Tivoli. Компанией HP предложена ITSM Reference Model, которая поддерживается продуктом HP OpenView Service Desk, в состав которой входит Automation Manager. Появились и постоянно расширяют свои возможности продукты других производителей, например AppCenter, Remedy, Front Range Solutions. Компоненты СУИ подключают к продуктам других классов, прежде всего систем хранения данных, например Veritas Storage Foundation от Symantec, Rainfinity от EMC и др. Кроме того, разработчики распространенных систем класса ERP, например, PeopleSoft и SAP, включили в состав своих основных продуктов компоненты СУИ. Эти продукты реализуют мониторинг, управление нагрузкой и ресурсами, администрирование ИТ-инфраструктуры, исходя из позиций поддержки системы бизнес-процессов компании. На основе этих продуктов создается единая информационная среда для централизованной поддержки развертывания и эксплуатации ИТ-инфраструктуры.

В СУИ часто включают компоненты продуктов Storage Resource Management (SRM). Каждый поставщик завершенной системы хранения данных (SAN) предлагает в составе своего решения SRM-систему для управления SAN. SRM-системы нацелены на снижение эксплуатационных расходов, улучшение использования ресурсов хранения и управляемости, контролируемости и доступности SAN. Известные представители этого класса, прежде всего Veritas Storage Foundation от Symantec, Rainfinity от EMC, реализуют широкий набор базовых функций, динамическое уровневое управление хранением данных, управление и мониторинг емкости хранения; централизованное управление системами и сетями хранения, управления производительностью и событиями, управление изменениями и конфигурацией и др. Их механизмы управления приложениями, настройками приложений и серверов в реальном времени с единой консоли, управления емкостью общего файлового ресурса и другими параметрами ИТ-инфраструктуры можно также использовать как компоненты СУИ.

В работе [341] описан комплексный продукт FLEX, предназначенный для управления виртуальным хостингом на серверном парке ЦОД. Авторами работы

разработан алгоритм распределения, названный «flex-alpha», который обеспечивает равномерность распределения нагрузки между серверами пула. Также в этой работе предложен набор шаблонов трафика, использование которых увеличивает производительность работы системы при определенных параметрах внешних нагрузок на серверы.

8.1.4. Виртуальный хостинг как объект управления

Рассмотрим особенности функционирования ЦОД, который предоставляет услуги большому количеству пользователей (см. рис. 8.2).

Виртуальный хостинг основан на СОА, веб-технологиях, технологиях виртуализации и кластеризации. ЦОД поддерживает различные веб-серверы приложений, развернутые на множестве N физических серверов (от $n \cdot 10$ до $n \cdot 100$).

Экземпляр веб-приложения A_l, $l = \overline{1,m}$ может быть запущен на любом физическом сервере N_i, $i = \overline{1,N}$, удовлетворяющем ресурсным требованиям приложения A_l. Кроме того, этот сервер должен удовлетворять специальным требованиям, которые не могут быть изменены с помощью функционала СУИ, например, настройки сети, канала связи, присутствие программных библиотек и т. п. Пользователи получают доступ к экземплярам нужных приложений, не зная, на каких физических серверах они расположены. В программной среде веб-серверов веб-запросы пользователей привязываются к веб-серверам приложений и передаются их конкретным экземплярам приложений для обработки. Экземпляры веб-приложений, которые запускаются и поддерживаются веб-серверами приложений, обрабатывают веб-запросы пользователей, и веб-серверы генерируют некоторое статическое или динамическое содержимое.

Серверный парк из n серверов должен поддерживать работу такого количества экземпляров приложений A_l, $l = \overline{1,m}$, которое должно обеспечить суммарные требования пользователей к объемам ресурсов, обеспечивающих требуемый уровень услуг. Могут иметь место некоторые аппаратные или другие ограничения, которые делают невозможным запуск определенных приложений. Когда серверы N_i, $i = \overline{1,N}$ не имеют достаточного количества r_{ki} ресурса R_k, $k = \overline{1,L}$, чтобы один из них мог поддерживать работу всех развернутых приложений A_l, $l = \overline{1,m}$, то приложения разворачивают на свободных физических серверах, которые объединяют в кластер. Таким образом формируется распределенная вычислительная среда с трехуровневой архитектурой, изображенной на рис. 8.2.

Эффективность функционирования такой среды можно повысить рациональным размещением экземпляров приложений A_i, $l = \overline{1,m}$ на серверах N_i, $i = \overline{1,N}$. Отсутствие жесткой привязки запросов пользователей к экземплярам приложений позволяет динамически распределять запросы между экземплярами соответствующего приложениями с учетом загруженности физических серверов. Поскольку требованиям пользователей присущи изменения, при накоплении изменений требований пользователей необходимо переходить к новому плану размещения приложений, увеличивая или ограничивая их ресурсы.

Если новый план размещения определен, то переход к нему заключается в настройках веб-серверов приложений, развертывании или свертывании приложений согласно новому плану, изменениях конфигурационных файлов и т. п. Использование других механизмов балансирования, например остановки и запуска

веб-серверов приложений на физических серверах, где находятся экземпляры приложений, позволяет избежать потенциально времязатратных операций развертывания или свертывания приложений, однако существенно ограничивает гибкость решений, поскольку не позволяет изменять набор развернутых приложений на сервере.

Значительное количество приложений $A_i,\ l=\overline{1,m}$ с разными ресурсными требованиями $r_{kl},\ k=\overline{1,L},\ l=\overline{1,m}$ и серверов $N_i,\ i=\overline{1,N}$ с различными характеристиками $r_{kl},\ k=\overline{1,L},\ i=\overline{1,N}$ превращают проблему рационального размещения экземпляров приложений $A_i,\ l=\overline{1,m}$ в далеко нетривиальную задачу.

Существующие подходы к управлению ресурсами ЦОД базируются на следующих общих принципах построения и организации процесса управления:

- управление ресурсами и нагрузкой реализуется путем сворачивания и разворачивания экземпляров приложений на серверах или на ВМ;
- планирование распределения состоит из этапов инициализации или стартового распределения и динамического перераспределения.

Для реализации этих принципов в СУИ необходимо внести компоненты, которые будут находить наилучшее размещение приложений и выполнять диспетчеризацию запросов между экземплярами приложений в зависимости от текущей нагрузки. Схема такой СУИ изображена на рис. 8.4.

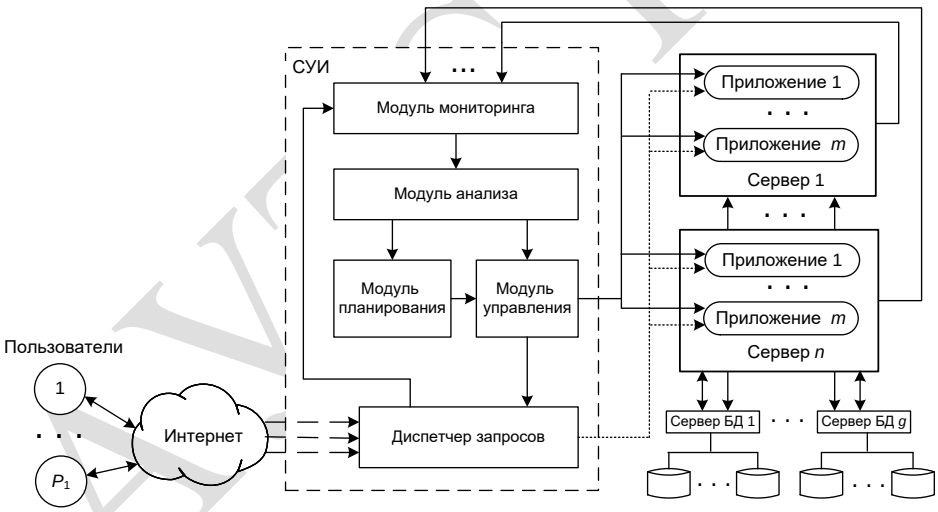


Рис. 8.4. Структура системы управления нагрузкой и ресурсами

СУИ управляет обслуживанием запросов и выделяет ресурсы приложениям в рассредоточенной вычислительной среде и распределяет для них ресурсы в соответствии с требованиями пользователей. При этом реализуется двухуровневая схема управления:

1) уровень планирования: определяется оптимальное размещение экземпляров приложений на физических серверах, при котором каждый сервер приложений получит достаточный объем ресурсов;

2) диспетчеризация потока запросов пользователей к серверам приложений с целью балансировки нагрузки физических серверов, на которых размещены экземпляры приложений.

Ресурсы сервера делятся на зависимые и независимые от его нагрузки. Использование первых напрямую зависит от нагрузки на приложение, в то время как загруженность приложений на сервере не влияет на независимые от нагрузки ресурсы. Требования приложения к зависимым от нагрузки ресурсам определяются на основе анализа требований совокупности пользователей, требования приложения к независимым от нагрузки ресурсам определяются собственно приложением.

Из зависимых от нагрузки ресурсов (процессорное время, отклик на запросы к жестким дискам, количество запущенных потоков и другие) будем учитывать только процессорное время, как наиболее критический для приложений параметр. Вычисляет этот параметр специальный модуль СУИ в тактах процессора в секунду, которые тратятся на обработку запроса пользователя к приложению. Среди независимых от требований приложений ресурсов, необходимых для развертывания экземпляра приложения на сервере приложений, учитываются объем оперативной памяти, каналы связи и память на жестких дисках, хотя многие исследователи учитывают только оперативную память.

Управление ресурсами и нагрузкой основывается на определении загруженности отдельных серверов и требованиях приложений к ресурсам.

Составляющие СУИ, приведенные на рис. 8.4, имеют следующее назначение:

модуль мониторинга осуществляет сбор информации о количестве задействованных и свободных ресурсов серверов, количестве необработанных запросов и других параметров, осуществляет контроль функционирования серверов и приложений, информирует элементы СУИ высшего уровня;

модуль анализа выполняет обработку статистической информации, уточняет требования, прогнозирует состояние серверов и требования приложений, определяет момент времени, когда необходимо рассчитывать новое размещение экземпляров приложений;

модуль планирования рассчитывает новый план размещения экземпляров приложений и распределения ресурсов с учетом возможностей серверов и требований приложений к зависимым и независимым от нагрузки ресурсам;

модуль управления реализует новое размещение приложений в системе с учетом предыдущего плана размещения приложений (разворачивает и сворачивает экземпляры приложений на серверах) и формирует правила выбора диспетчером экземпляров приложений для направления запросов пользователей с учетом плана размещения и текущей нагрузки серверов;

диспетчер запросов направляет запросы пользователей экземплярам приложений на основе правил, полученных от модуля управления.

Модуль управления старается минимизировать влияние операций сворачивания и разворачивания приложений на уже работающие приложения. Поэтому не более одного экземпляра приложения в каждый момент времени сворачивается или разворачивается на любом сервере, при этом отслеживаются ограничения независимой от нагрузки емкости каждого сервера при изменении его распределения. Как минимум один экземпляр приложения должен быть всегда запущен в системе.

Предложенная схема обеспечивает гибкость и масштабируемость СУИ. Появляется возможность изменять множество параметров конфигурации, например добавляя или удаляя серверы, создавая или удаляя кластеры, изменяя серверы и настройки кластеров и серверов, причем некоторые в реальном времени. Предполагается, что часть информации, например о требованиях приложений, администраторы вводят вручную.

8.2. УПРАВЛЕНИЕ НАГРУЗКОЙ И РЕСУРСАМИ ЦОД ПРИ ВИРТУАЛЬНОМ ХОСТИНГЕ

8.2.1. Постановка задачи управления ресурсами и нагрузкой ЦОД при виртуальном хостинге

Пусть в ЦОД принята модель виртуального хостинга с поддержкой технологии СОА. Физические серверы могут объединяться в кластеры. Серверный парк ЦОД обслуживает пользователей, предоставляя им доступ к экземплярам множества приложений. Серверы характеризуются набором технических параметров, а приложения – набором требований к ресурсам. Мощные приложения требуют консолидации ресурсов, т. е. несколько серверов, объединенных в кластер, обеспечивают функционирование одного приложения.

Необходимо разработать математические модели и соответствующие методы планирования и диспетчирования нагрузки и распределения ресурсов физических серверов между приложениями с учетом целесообразных критериев эффективности при выполнении ресурсных, технологических и других ограничений.

При этом моделируется функционирование ЦОД, который предоставляет услуги веб-приложений пользователям по схеме, приведенной на рис. 8.2.

8.2.2. Математическая модель распределения ресурсов при виртуальном хостинге

Для формулирования задач оптимального распределения ресурсов введем некоторые обозначения. Необходимо специфицировать показатели и параметры, с помощью которых можно описать систему, предоставляющую услуги пользователям, распределяя их запросы экземплярам приложений на задействованных серверах.

Основу ИТ-инфраструктуры ЦОД составляет множество $N = \{N_i\},\ i = \overline{1,n}$ физических серверов, количество которых равно n. На серверах работают приложения из множества $A = \{A_l\},\ l = \overline{1,m}$. Приложения развернуты на множестве ВМ $\{V_j\},\ j = \overline{1,m}$. Приложения из множества A поддерживают бизнес-процессы Z_p, $p = \overline{1,z}$.

Каждый из процессов $Z_p,\ p = \overline{1,z}$ характеризуется коэффициентом важности $w_p,\ p = \overline{1,z}$, значение которого выбирается из фиксированного интервала натуральных чисел.

Каждая j-я ВМ использует количество r_{kj} ресурса $R_k,\ k = \overline{1,L}$.

Каждый сервер $N_i,\ i = \overline{1,n}$ характеризуется набором параметров, которые вместе определяют его мощность:

$$(\Omega_i, \Gamma_i, \Phi_i, \Lambda_i), \tag{8.1}$$

где Ω_i – процессорная емкость сервера N_i; Γ_i – емкость ОЗУ сервера N_i; Φ_i – емкость жестких дисков сервера N_i; Λ_i – емкость каналов сервера N_i.

Каждое приложение A_l, $l=\overline{1,m}$ выдвигает набор требований к ВМ или серверу, на которых оно будет развернуто:

$$(\omega_l, \gamma_l, \varphi_l, \lambda_l), \tag{8.2}$$

где ω_l – требования к процессорной емкости серверов, на которых расположены экземпляры применения A_j; γ_l – требования к ОЗУ серверов, на которых расположены экземпляры применения A_j; φ_l – требования к объемам памяти жестких дисков серверов, на которых расположены экземпляры применения A_j; λ_l – требования к емкости каналов связи к серверам, на которых расположены экземпляры приложения A_j.

Ресурсы i-го сервера r_{ki} и требования приложений делятся на два класса, зависимые от нагрузки:

$$(\Omega_i, \omega_l, \Lambda_i, \lambda_l), \tag{8.3}$$

и независимые от нагрузки:

$$(\Gamma_i, \gamma_l, \Phi_i, \varphi_l). \tag{8.4}$$

Требования приложения к зависимым от нагрузки ресурсам определяются на основе анализа требований совокупности пользователей, требования приложения к независимым от нагрузки ресурсам определяются собственно приложением.

На этапе планирования сложно предсказать будущие потребности каждой j-й, $j=\overline{1,m}$ ВМ в ресурсах, поскольку в процессе функционирования нагрузка на приложения A_l, $l=\overline{1,m}$ варьируется [136]. Поэтому для поддержания согласованного уровня услуг необходимо осуществлять динамическое выделение дополнительного количества Δr_{kj} k-го ресурса, $k=\overline{1,L}$ j-й ВМ, что влечет за собой необходимость перемещения ВМ между серверами и переход к новому плану размещения ВМ.

Другой актуальной задачей является снижение энергопотребления серверной инфраструктуры, что требует разработки методов минимизации задействованных ресурсов R_k, $k=\overline{1,L}$ путем оптимизации распределения ВМ V_j, $j=\overline{1,m}$ и последующего выключения высвобожденных серверов.

Введем вектор $C = \{C_1,\ldots,C_m\}$, элементы которого

$$C_l = \begin{cases} 1, & \text{если СУИ может разворачивать и сворачивать приложение } A_l; \\ 0, & \text{в противном случае;} \end{cases} \tag{8.5}$$

характеризуют и матрицу $G = |G_{li}|_{m\times n}$, элемент которой

$$G_{li} = \begin{cases} 1, & \text{если экземпляр приложения } A_l \text{ можно развернуть на сервере } N_i; \\ 0, & \text{в противном случае;} \end{cases}$$

характеризует соотношение серверов и приложений.

Из описания смоделированной системы легко понять, что требования к суммарной процессорной емкости серверов и емкости каналов, которые определяются на основе потребностей всех пользователей, могут обеспечить несколько серверов своей суммарной процессорной емкостью и емкостью каналов. Тогда на каждом из этих серверов размещают экземпляр приложения. Но величины γ_l и φ_l определяются для экземпляра приложения, и если ОЗУ сервера меньше γ_l или объем его жестких дисков меньше φ_l, то экземпляр приложения A_l можно разместить на сервере.

Введем переменную $x_{li} \in [0, 1]$ – доля серверных ресурсов в части процессорной емкости сервера N_i, выделяемая экземпляру приложения A_l.

Задачу планирования распределения ресурсов целесообразно рассматривать как задачу определения матрицы $\mathbf{X} = \|x_{li}\|$, которая определяет доли ресурсов серверов, выделяемых экземплярам приложений. В качестве критериев целесообразно использовать: максимальную суммарную важность приложений, которые получили ресурсы; равномерную загрузку серверов; минимальные эксплуатационные расходы; минимальную стоимость перехода к новому плану размещения приложений.

Оптимальное размещение должно учитывать следующие ограничения:

1) суммарные независимые от нагрузки требования назначенных для размещения на сервере экземпляров приложений не должны превышать соответствующие параметры сервера;

2) зависимые от загрузки требования приложений следует обеспечить суммарными возможностями зависимых от загрузки параметров тех серверов, на которых размещаются экземпляры этого приложения;

3) ограничения, определяемые матрицей G и вектором C.

Из-за особенностей критериев целесообразно рассматривать несколько моделей с существенными различиями в обеспеченности серверами.

Поскольку для распределения ограниченных ресурсов ИУС можно использовать модели, предложенные в работах [114, 144, 148], остановимся *на моделях планирования при избытке ресурсов*. Такие условия характерны для современных ЦОД. В зависимости от стратегии хостинговой организации можно выбрать одну из моделей, которые различаются критериями оптимизации и ограничениями.

Задача 1. Если хостинговой организации важна экономия ресурсов, то возникает задача минимизации расходов на поддержку клиентских запросов. При этом выбирается такое минимальное по затратам на их поддержку подмножество серверов, которое может поддержать необходимое для удовлетворения требований существующих пользователей множество экземпляров приложений.

Полагаем, что базы данных организованы в хранилищах, доступ к которым посредством SAN имеют все экземпляры приложений, на каких бы серверах они не размещались.

Обозначим через s_i эксплуатационные расходы на поддержку сервера N_i и введем переменную y_i, $i = \overline{1,n}$, принимающую следующие значения:

$$y_i = \begin{cases} 1, \text{ если сервер } N_i \text{ задействован для поддержания одного и больше} \\ \quad \text{экземпляров одного или разных приложений;} \\ 0, \text{ если физический сервер } N_i \text{ не поддерживает ни одного приложения.} \end{cases}$$

Для удобной формулировки ограничений на независимые от нагрузки ресурсы введем дискретную переменную:

$$\tilde{x}_{li} = \begin{cases} 1, \text{ если } x_{li} > 0; \\ 0, \text{ если } x_{li} = 0. \end{cases}$$

Выражение для установления связи между переменными имеет вид

$$y_i = x_{1i} \vee x_{2i} \vee \ldots \vee x_{mi}, \; i = \overline{1,n}. \tag{8.6}$$

Критерий для этой задачи имеет следующий вид:

$$\min \sum_{i=1}^{n} s_i y_i \; . \tag{8.7}$$

Минимальное значение необходимо выбирать из решений, соответствующих структуре булевых значений матрицы G и удовлетворяющих ограничениям (8.8)–(8.13):

$$\sum_{l=1}^{m} \tilde{x}_{li} \cdot \gamma_l \le \Gamma_i, \;\; i = \overline{1,n}, \tag{8.8}$$

требующее, чтобы емкость ОЗУ сервера N_i была достаточной для размещения запланированных для него экземпляров приложений;

$$\sum_{l=1}^{m} \tilde{x}_{li} \cdot \varphi_l \le \Phi_i; \;\; i = \overline{1,n}, \tag{8.9}$$

требующее, чтобы емкость дисковой памяти сервера N_i была достаточной для работы всех экземпляров приложений, запланированных для него;

$$\sum_{l=1}^{m} x_{li} \le 1, \;\; i = \overline{1,n}, \tag{8.10}$$

требующее, чтобы емкость процессоров сервера N_i была достаточной для размещения и эффективной работы всех экземпляров приложений в долях, запланированных для него;

$$\sum_{l=1}^{m} x_{li} \cdot \lambda_l \le 1, \; i = \overline{1,n}, \tag{8.11}$$

требующее, чтобы емкость каналов связи сервера N_i была достаточной для размещения и эффективной работы всех экземпляров приложений в долях, запланированных для него;

$$\sum_{i=1}^{n} x_{li} \cdot \Omega_i \le \omega_l, \;\; l = \overline{1,m}, \tag{8.12}$$

АУТСКП

требующее, чтобы потребности всех пользователей приложения A_j в емкости процессоров удовлетворяли совокупности выбранных долей физических серверов с экземплярами этого приложения;

$$\sum_{i=1}^{n}]x_{li}[\geq 1,\ l = \overline{1,m}, \tag{8.13}$$

требующее, чтобы для каждого приложения имелся хотя бы один экземпляр с ненулевой долей на каком-то сервере.

Задача 2. В случаях, когда загрузке серверов свойственны пиковые периоды, для хостинговой организации важно обеспечить равномерную загрузку серверов. Тогда серверы высвобождаются и их недогрузка компенсирует всплеск запросов пользователей. Возникает задача минимизации суммарного недогрузки физических серверов при условии выполнения ограничений на ресурсы и технологических ограничений.

С использованием введенных выше обозначений критерий этой задачи можно записать в таком виде:

$$\min \sum_{i=1}^{n} \left(1 - \sum_{l=1}^{m} x_{li} \right)^2. \tag{8.14}$$

Минимальное значение следует выбирать из решений, которые соответствуют структуре булевых значений матрицы G и удовлетворяют ограничениям (8.8)–(8.12) и

$$\sum_{l=1}^{m} \tilde{x}_{li} \geq 1,\ i = \overline{1,n}. \tag{8.15}$$

Ограничение (8.15) требует, чтобы на каждом сервере выполнялся хотя бы один экземпляр какого-то приложения.

Задача 3. В некоторых случаях затраты на реализацию перехода к новому оптимальному плану размещения приложений сравнимы с расходами на поддержку клиентских запросов. Тогда важно учесть расходы на реализацию перехода к новому оптимальному плану размещения приложений в критерии оптимальности нового плана. Возникает задача минимизации суммарных затрат на поддержку клиентских запросов и реализацию перехода к новому оптимальному плану размещения приложений. Иначе говоря, выбирается такое минимальное по сумме расходов на реализацию перехода к новому оптимальному плану размещения приложений и на их поддержку подмножество серверов, которое сможет поддержать необходимое для удовлетворения требований существующих пользователей множество экземпляров приложений.

С использованием введенных выше обозначений критерий этой задачи можно записать в виде

$$\min \left(\sum_{i=1}^{n} \left(s_i y_i + \sum_{l=1}^{m} p_{il} \left| \tilde{x}'_{li} - \tilde{x}_{li} \right| \right) \right). \tag{8.16}$$

БГТС КП

В выражении (8.16) p_{li} обозначает затраты, связанные с переходом к новому оптимальному плану размещения приложений при его отличии от старого плана для приложения A_l на сервере N_i; $\tilde{x}'_{li}$ – значение переменной $\tilde{x}_{li}$ в старом плане.

Минимальное значение необходимо выбирать из решений, соответствующих структуре булевых значений матрицы G и удовлетворяющих ограничениям (8.8)–(8.12).

Задача 4. Если для хостинговой организации важна равномерная загрузка серверов, а затраты на реализацию перехода к новому оптимальному плану размещения приложений сравнимы по потерям от недозагрузки серверов до полной мощности, то возникает задача минимизации суммарных потерь от равномерной недогрузки физических серверов и расходов на реализацию перехода к новому оптимальному плану размещения приложений при условии выполнения ограничений на ресурсы и технологических ограничений.

Критерий для этой задачи можно записать в виде

$$\min \sum_{i=1}\left(\delta_i (1-\sum_{l=1}^{m} x_{li}) + \sum_{l=1}^{m} p_{il} \left| \tilde{x}'_{li} - \tilde{x}_{li} \right| \right), \qquad (8.17)$$

где δ_i обозначает потери, связанные с недогрузкой сервера N_i.

Минимальное значение необходимо выбирать из решений, соответствующих структуре значений матрицы $R = |R_{ji}|_{m \times n}$ и удовлетворяющих ограничениям (8.8)–(8.13).

Приведенные выше задачи оптимизации размещений для одного кластера – это, по сути, линейные и нелинейные варианты задачи смешанного программирования с непрерывными и булевыми переменными. В случае нескольких кластеров они являются вариантами задачи смешанного программирования с непрерывными и булевыми переменными на множестве серверов, ограниченном кластерами. Установлено, что эти задачи являются NP-полными. Поскольку обойти переборные методы их решения нельзя, оправданным будет использование эвристических методов или методов на основе генетических алгоритмов, применение которых дает контролируемое приближение к точному результату [281].

Модели динамического управления нагрузкой. Модуль управления определяет для диспетчера правила, которые устанавливают доли серверов для каждого типа приложений, к которым направляются запросы. Правила устанавливаются модулем управления с учетом загруженности серверов и выбранной стратегии балансировки.

8.2.3. Алгоритмы решения задач

Для решения сформулированных выше задач оптимального размещения предлагаются алгоритмы трех типов:

- варианты ГА, предложенные в работах [141, 160, 161];
- варианты эвристического алгоритма, который учитывает специфику критериев и ограничений и построен на идеях, предложенных в работе [160];
- варианты комбинированного алгоритма, построенные на применении на первом этапе решения задачи алгоритмов линейного программирования и применения на втором этапе алгоритмов перебора для переменных, значения которых оказались не целыми.

АЛТГТУ

Варианты генетического алгоритма. Рассмотрим универсальный вариант такого алгоритма, разработанный на основе работы [169], который можно модифицировать для каждой из приведенных задач:

Шаг 0. Устанавливаем $i = 0$. Генерируем случайную начальную популяцию $P_k(0) = \{p_1(0),...,p_k(0)\}$ с учетом возможного отсеивания при проверке ограничений.

Шаг 1. Для каждой цепочки популяции проверяем выполнение ограничений соответствующей задачи, например, для задачи 1– ограничений (8.8)–(8.13). Для тех цепочек $p_q(0)$, для которых ограничения выполняются, оцениваем $f(p_q(0))$ целевую функцию соответствующей задачи, например, для задачи 1 это $\sum_{i=1}^{n} s_i y_i$, и выбираем наилучшее решение.

Шаг 2. Отбираем представителей для новой популяции (цепочек с лучшим значением функции $f(p_q(0))$.

Шаг 3. $L = l + 1$. Создаем новую популяцию, применяя оператор рекомбинации.

Шаг 4. Улучшаем популяцию, применяя оператор мутации.

Шаг 5. Оцениваем каждую цепочку популяции касательно выполнения ограничений и значений функции $f(p_q(l))$, если ограничение выполняется.

Шаг 6. Если решение лучше предыдущего, то завершаем выполнение алгоритма, в противном случае возвращаемся на шаг 3.

Выбираем лучшее решение в образованной популяции $P_k(l)$.

Здесь каждая цепочка (особь) любой популяции составляет совокупность двоичных векторов D, имеющих размер $m \times n$, что определяется количеством физических серверов и приложений, т. е. задает набор значений $\{\tilde{x}_{li}\}$, $l = \overline{1,m}$, $i = \overline{1,n}$. Двоичные представления значений переменных генерируются только для элементов двоичного вектора D, которые приобретают значение 1.

Для рекомбинации сначала используется одноточечный кроссовер со случайным выбором точки кроссовера по вектору D. Если при этом в векторе D значение элемента $\tilde{x}_{li}$ изменяется с 1 на 0, то соответствующий элемент массива B двоичных представлений также приобретает значение 0. После этого выполняется одноточечный кроссовер со случайным выбором точки кроссовера по случайно выбранным элементами массива B. Реализация мутаций осуществляется на основе популярной схемы инверсии участков цепочек.

Для новой популяции особи отбирают на основе ранжирования по значению целевой функции особей предыдущей популяции, причем ранжируются лишь особи, удовлетворяющие ресурсным, технологическим и иным ограничениям, прежде всего ограничениям, определенным матрицей R и вектором C, соответствующей задачи 1–4.

Варианты эвристического алгоритма. Применение алгоритмов, основанных на интуитивных представлениях, учитывающих роль зависимых и независимых ресурсов в виде критерия, в работе [160] привело к обнадеживающим результатам. Учитывая, что самыми критичными из ресурсов являются процессорная емкость и ОЗУ, причем процессорную емкость можно обеспечить совокупно несколькими серверами, а ОЗУ только одним, логично сначала разместить приложения, имеющие высокие потребности в ОЗУ. Если взять это за основу работы алгоритма, то необходимо лишь учесть особенности критерия. Как и в работе

[160], каждому физическому серверу N_i припишем коэффициент P_i, в качестве значения которого примем результат деления процессорной емкости сервера N_i на объем его ОЗУ. Также каждому приложению A_l поставим в соответствие коэффициент ρ_l, который принимает значение результата деления требований приложения A_l к процессорной емкости на требования этого приложения к емкости ОЗУ сервера.

Предварительный этап для любого из приведенных ниже вариантов эвристического алгоритма заключается в сортировке: физических серверов по значению их коэффициента P в порядке возрастания и приложений по значению их коэффициента ρ в порядке убывания.

Эвристический алгоритм для задачи 1. Выполняется предварительный этап.

Шаг 1. Поиск приложения, которое имеет наименьшее значение коэффициента ρ.

Шаг 2. Поиск сервера с наибольшим значением коэффициента P.

Шаг 3. Если для избранных сервера и применения не выполняются ограничения задачи 1 касательно независящих от нагрузки ресурсов, то возвращаемся на шаг 2 для поиска следующего сервера.

Шаг 4. Если для выбранных сервера и приложения выполняются ограничения задачи 1 по независящим от нагрузки ресурсов, причем зависимые от нагрузки требования полностью удовлетворены, то переходим на шаг 5. Если выполняются ограничения задачи 1 по независящим от нагрузки ресурсов, но сервер не может полностью удовлетворить зависимые от нагрузки требования приложения, то переходим на шаг 6.

Шаг 5. Приложение исключается из списка приложений, корректируется коэффициент P сервера и он включается в список серверов согласно новому значению коэффициента. Переход на шаг 7.

Шаг 6. Оставшиеся ресурсы сервера отдаются приложению и сервер изымается из списка серверов. Переходим на шаг 2.

Шаг 7. Если список приложений пуст, то конец, иначе переходим на шаг 1.

Вычислительная сложность эвристического алгоритма составляет $m \times n$.

Для других критериев используем другую логику выбора серверов из их упорядоченного списка.

Варианты комбинированного алгоритма. Продемонстрируем на примере задачи 1, что для задач 1–4 можно использовать методы смешанного программирования с непрерывными и булевыми переменными. В задаче 1 принимаем $\tilde{x}_{li}$ за x_{li} и ограничения (8.8)–(8.13) записываем в таком виде:

$$\sum_{i=1}^{n} x_{li} > 0,\ l = \overline{1,m}. \qquad (8.18)$$

Тогда задача (8.8)–(8.13), (8.18) является задачей линейного программирования, для решения которой можно применять аффинно-масштабирующий алгоритм [30] и модифицированный симплекс-метод [34].

После получения решения задачи линейного программирования применяем схему метода неявного перебора к дробным компонентам матрицы $\mathbf{X} = \|x_{li}\|$. При этом необходимо добиться выполнения ограничений (8.8), (8.9) и не выйти за ограничения (8.10)–(8.13).

Выигрывая во времени и несколько пренебрегая точностью, полную схему неявного перебора из дробных компонентов матрицы $\mathbf{X} = \|x_{li}\|$ можно заменить схемой разгрузки некоторых серверов, для которых не выполняются ограничения (8.8) и (8.9), перенеся экземпляры некоторых приложений на недогруженные серверы. Для этого начинаем с перегруженных серверов с наименьшим значением коэффициента Р, а приложения переносим на недогруженные серверы с наибольшим значением коэффициента Р. Прежде всего снимаем с перегруженных серверов приложения с наименьшим значением коэффициента ρ.

Алгоритм динамического управления приложениями. Диспетчер распределяет запросы в соответствии с правилами, полученными от модуля управления, балансируя загрузку серверов в пределах определенного плана размещения. Для каждого запроса, который требует распределения, диспетчер определяет сервер согласно правилам, которые устанавливают доли серверов для каждого типа приложений, к которым направлены запросы. Для этого вводится указатель порядкового номера последнего запроса для каждого приложения. Правила определяют сервер для каждого порядкового номера. Нумерация циклично обновляется, когда распределены все части цикла.

Таким образом, разработаны модели управления нагрузкой и ресурсами ЦОД при виртуальном хостинге с кластеризацией ресурсов для распределения экземпляров приложений между серверами с учетом различных критериев эффективности и ограничений. В отличие от других технологий разработанные модели предоставляют администраторам возможность использования ряда критериев, учета нескольких типов ресурсов, гибкость в переходе к новым планам размещения приложений.

8.3. УПРАВЛЕНИЕ НАГРУЗКОЙ И РЕСУРСАМИ ЦОД ПРИ ВЫДЕЛЕННЫХ СЕРВЕРАХ

8.3.1. Постановка проблемы распределения и управления распределением ресурсов при выделенных серверах

Предлагается подход к управлению распределением ресурсов, обеспечивающим эффективное их использование в процессе функционирования ЦОД по схеме выделенных серверов, когда клиенты из множества $K = \{K_j\}$, $j = \overline{1, p}$ не соревнуются за ресурсы, а могут использовать резервируемые для них физические серверы из множества *N*, считая, что это защитит их от перегрузки серверов, плохой производительности и несовершенных программ соседей по хостингу. Серверы разрешается объединять в кластеры. На рис. 8.4 показана схема организации доступа к ресурсам при выделенных серверах. На каждом из серверов развернуты веб-сервер приложений и несколько приложений клиента.

Достижение преимуществ клиент-серверных и веб-технологий, технологий кластеризации и виртуализации требует дополнительных усилий, поскольку процессы принятия решений становятся сложнее и требуют развитого инструментария их поддержки. Необходимо определить состав кластеров, параметры ВМ и эффективно управлять их нагрузкой и ресурсами. Возникает необходимость разработки моделей, алгоритмов и средств мониторинга, анализа, планирования и

управления информационно-вычислительными процессами ЦОД с учетом специфики архитектурных решений, особенностей кластеризации и виртуализации и реализации их в СУИ.

Эффективность управления СУИ существенно зависит от моделей и методов решения научных проблем, связанных с распределением ресурсов, управлением ресурсами и нагрузкой, балансировкой и т. д. На сегодня проблема распределения ресурсов и управлением нагрузкой решена не только в рамках единой аппаратной системы, но и в сетях, прежде всего для виртуального хостинга [160, 181, 231, 273, 374, 377, 384, 391, 392]. Предложены подходы для планирования и управления ресурсами и нагрузкой [144, 148, 173]. Однако распределение и управление серверами и группами серверов при использовании хостинговой компанией модели выделенных серверов требует дополнительных исследований. Кроме того, разработка моделей и методов эффективного управления нагрузкой и ресурсами ЦОД при хостинге на основе выделенных серверов составляет важную часть комплексного решения для распределения и управления ресурсами ЦОД, которое можно использовать в любой ситуации, настраивая его на особенности среды применения.

Модель хостинга с выделенными серверами поддерживается клиент-серверными и сервис-ориентированными технологиями. Физические серверы разрешается объединять в кластеры для удовлетворения требований клиентов. Серверный парк обслуживает множество пользователей, предоставляя им доступ к множеству соответствующих приложений клиентов хостинговой компании. Серверы характеризуются набором технических параметров, а клиенты – набором требований, зависящих от множества пользователей и приложений клиентов. Требования приложений такие, что несколько серверов обеспечивают экземпляры одного приложения.

Необходимо разработать математические модели и соответствующие методы распределения физических серверов между клиентами по целесообразным критериям эффективности при выполнении ресурсных, технологических и других актуальных ограничений.

8.3.2. Описание функционирования ЦОД на основе выделенных серверов

Рассмотрим функционирование ЦОД, когда клиенты выбрали модель хостинга на основе выделенных серверов. Хостинговая компания и организации-клиенты имеют возможность выбирать технологии построения приложений, кластеризации и виртуализации. Ниже описаны три варианта системы, которая предоставляет хостинговые услуги на основе модели выделенных серверов.

Выделенные серверы и СОА. Каждый клиент получает в свое распоряжение один физический сервер или группу физических серверов. В последнем случае используются технологии кластеризации для объединения физических серверов в кластер. На серверах и кластерах могут использоваться технологии виртуализации. Каждый из клиентов создает свои программные системы на основе СОА. Веб-приложения обрабатывают веб-запросы пользователей клиента и веб-серверы генерируют некоторое статическое или динамическое содержимое.

Администраторы устанавливают веб-сервер, сервер БД и приложения клиента на физических серверах и кластерах клиента, формируя вычислительную среду для пользователей этого клиента. На схеме, приведенной на рис. 8.5, показана организация доступа к ресурсам при выделенных серверах.

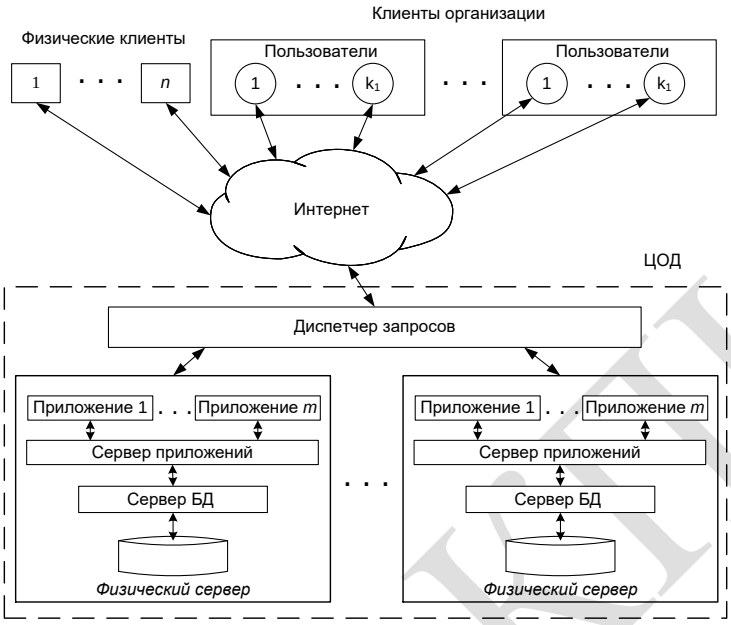


Рис. 8.5. Схема доступа к ресурсам при выделенных серверах

На каждом из физических серверов (см. рис. 8.5) развернуты сервер БД, веб-сервер приложений и несколько приложений отдельного клиента. Один ЦОД может обслуживать много разнообразных веб-серверов приложений, развернутых на физических серверах или в кластерах в количестве от $n \cdot 10$ до $n \cdot 100$. Поскольку отдельные физические серверы ЦОД могут не иметь достаточных вычислительных мощностей, ОЗУ и других ресурсов, администраторы ЦОД создают для клиентов с высоким уровнем требований к ресурсам кластеры на основе групп физических серверов.

Программные среды веб-серверов приложений используют программные реализации диспетчеров запросов, чтобы доставить веб-серверные запросы приложениям. При виртуализации могут применяться технологии распределения ресурсов ВМ и нагрузки на основе специальных правил для выбора ВМ и приложений. В СУИ также реализуется отслеживание всего пути запросов и предотвращения ситуаций, когда запрос теряется из-за зависшего приложения или технических проблем.

Выделенные серверы и традиционная трехуровневая архитектура. Каждому клиенту выделяется отдельный физический сервер или отдельная их группа при использовании технологии кластеризации. Каждый из клиентов создает свои программные системы на основе традиционных клиент-серверных технологий. Серверы приложений анализируют запросы пользователей, проверяют определенные параметры и направляют запросы соответствующим приложениям. Приложение обрабатывает запросы пользователей клиента и результаты через сервер приложений возвращаются пользователям. Серверы приложений, при необходимости, обращаются за данными к серверам БД.

Администраторы устанавливают сервер приложений, сервер БД и приложение клиента на физическом сервере, формируя вычислительную среду для пользователей этого клиента. В случае группы серверов такая среда формируется на

кластере. Как для отдельных серверов, так и для кластеров могут применяться технологии виртуализации.

Программные среды серверов приложений также используют программные реализации диспетчеров запросов, чтобы доставить запросы приложениям, а ответы – пользователям. Они привязывают запросы к серверу приложений и передают их конкретным приложениям для обработки. Если задействованы технологии виртуализации, то применяются технологии распределения ресурсов ВМ и нагрузки на основе специальных правил для выбора ВМ и приложений.

Гибридная архитектура. Принципиально возможна реализация модели, когда часть клиентов строят свои программные системы на основе СОА и используют каналы интернета для обеспечения доступа пользователей к своим веб-приложениям, а другие клиенты используют каналы связи с хостинговой компанией или выделенные каналы и традиционные клиент-серверные технологии для обеспечения доступа своих пользователей к приложениям.

Особенности СУИ. На сегодня для любого из приведенных вариантов построения системы предоставления услуг с выделенными серверами проблема распределения физических серверов и групп серверов между клиентами остается не решенной. Когда ЦОД обслуживает значительное количество клиентов, к большому количеству приложений которых обращается много пользователей, причем каждый клиент и его приложение имеют свои системные требования, а мощный серверный парк характеризуется существенным разнообразием параметров, проблема наилучшего закрепления серверов за клиентами становится далеко не тривиальной.

Схема СУИ изображена на рис. 8.4. Система является распределенной и на схеме изображен только компонент, который отвечает за сбор информации о состоянии и загруженности серверов и приложений, анализ собранной информации, распределение серверов между клиентами. На схеме не показан компонент системы, который размещается на каждом сервере и реализует функции мониторинга, анализа и управления приложениями. Соответственно не нашел отражение и компонент управления кластером. Специфика модулей СУИ на рис. 8.4 для решения рассматриваемых в разделе задач заключается в следующем: диспетчер запросов осуществляет маршрутизацию запросов в соответствии с распределением серверов между клиентами; модуль планирования определяет новое закрепление физических серверов (групп серверов) между клиентами; модуль управления осуществляет развертывания веб-серверов и приложений согласно новому закреплению физических серверов между клиентами, их настройки и формирование правил для диспетчера. Назначение остальных модулей не изменяется.

Главным элементом модели является модуль планирования, который рассчитывает новое закрепление физических серверов за клиентами в зависимости от требований клиентов, потребностей их приложений к ресурсам. В каждом новом цикле работы системы модуль планирования рассчитывает новую матрицу закрепления с целью удовлетворить требования всех клиентов и их приложений в ресурсах, не превышая емкости узлов. При работе модуль планирования учитывает следующие факторы:

- потенциальное время- и ресурсозатратность процессов свертывания и развертывания приложений на выделенных для клиентов серверах;

• если существует много матриц закрепления, которые минимизируют расходы на поддержку выделенных серверов и обеспечивают выполнение ресурсных ограничений, то лучшим из закреплений будет то, которое обеспечивает равномерное использование ресурсов;

• технологические ограничения, которые делают невозможным использование некоторых серверов для некоторых клиентов, например настройки сети, присутствие программных библиотек и т. п.

Требования клиентов и приложений традиционно делятся на зависимые и независимые от нагрузки. Целесообразно также учесть и затем контролировать качественные требования клиентов, указанные в SLA.

Здесь не рассматриваются планирование и управление операциями сворачивания и разворачивания экземпляров приложений, поскольку это присуще только виртуальному хостингу. Можно использовать операции запуска и остановки ВМ, если используются технологии виртуализации, но это тема отдельного рассмотрения.

Необходимо разработать модель закрепления клиентов за физическими серверами и группами физических серверов с учетом приведенных выше особенностей ИТ-инфраструктуры.

8.3.3. Разработка математической модели

Для удобного формулирования задач оптимального распределения ресурсов уточним базовые понятия и введем необходимые обозначения.

Итак, клиентам K_j, $j=\overline{1,p}$ предоставляются услуги по доступу пользователей к ресурсам за счет распределения запросов между экземплярами приложений A_{lj}, $l=\overline{1,m}$, $j=\overline{1,p}$ на физических серверах N_i, $i=\overline{1,n}$.

Для каждого физического сервера N_i, $i=\overline{1,n}$ в соответствии с выражением (8.1) определены пять параметров, которые вместе характеризуют его мощность:

$$(\Omega_i, \Gamma_i, \Phi_i, \Lambda_i, \Delta_i), \tag{8.19}$$

где Ω_i – процессорная емкость сервера N_i; Γ_i – емкость ОЗУ сервера N_i; Φ_i – емкость жестких дисков сервера N_i; Λ_i – емкость каналов сервера N_i; Δ_i – надежность сервера N_i.

Каждый клиент K_j, $j=\overline{1,p}$ выдвигает требования к параметрам, определенным в выражении (8.2): ω_j – к процессорной емкости серверов, на которых расположены приложения, которые обслуживают пользователей клиента K_j; γ_j – к ОЗУ серверов, на которых расположены приложения, которые обслуживают пользователей клиента K_j; φ_j – к памяти жестких дисков серверов, на которых расположены применения, которые обслуживают пользователей клиента K_j; λ_j – к емкости каналов серверов, на которых расположены применения, которые обслуживают пользователей клиента K_j; δ_j – к надежности оборудования, обслуживающего пользователей клиента K_j.

Зададим матрицу $G = |G_{ji}|_{p\times n}$, элементы которой определяют возможность разворачивания приложений на серверах:

АУТСКИ

$$G_{ji} = \begin{cases} 1, & \text{если приложение клиента } K_j \text{ можно развернуть на сервере } N_i; \\ 0, & \text{в противном случае.} \end{cases}$$

Как приведено выше, и характеристики серверов, и требования приложений делятся на два класса (8.3) и (8.4): зависимые от нагрузки (Ω_i, ω_j, Λ_i, λ_j) и независимые от нагрузки (Γ_i, γ_j, Φ_i, φ_j, Δ_i, δ_j). Требования к зависимым от нагрузки параметрам серверов определяются суммарными требованиям пользователей клиента, а требования к независимым от загрузки параметрам серверов – суммарными требованиями всех приложений клиента.

Из описания моделируемой системы легко понять, что величину ω_j могут обеспечить несколько серверов своей суммарной процессорной емкостью. Тогда на каждом сервере размещается экземпляр приложения. Но величины γ_j определяются потребностями приложений клиента, и если память сервера меньше требований приложения, то это приложение на сервере не может быть размещено. Приложение клиента может использовать ОЗУ только того сервера, на котором оно размещается. Если приложение размещается на нескольких серверах, то каждый из них должен выделить для него ОЗУ в соответствии с требованиями приложения. Это же касается и памяти жестких дисков серверов. Поэтому требования каждого клиента K_j, $j = \overline{1, p}$, к ОЗУ состоят из суммарных требований $\gamma_{\Sigma j}$ к ОЗУ всех приложений клиента K_j и требований $\gamma_{\max j}$ самого требовательного к ОЗУ приложения клиента K_j. Аналогично требования каждого клиента K_j, $j = \overline{1, p}$, к памяти жестких дисков состоят из суммарных требований $\varphi_{\Sigma j}$ к памяти жестких дисков всех приложений клиента K_j и требований $\varphi_{\max j}$ самого требовательного к памяти жестких дисков приложения клиента K_j.

Введем переменную, определяющую размещение приложений на серверах:

$$x_{ji} = \begin{cases} 1, & \text{если сервер } N_i \text{ выделяется для клиента } K_j; \\ 0, & \text{в противном случае.} \end{cases}$$

Исходя из особенностей структуры СУИ, задачу планирования размещения приложений на серверах разбиваем на две задачи:

1) планирование размещения приложений на серверах (на период, при определенных ситуациях);

2) диспетчеризация нагрузки в течение планового периода.

Первая из этих задач, в свою очередь, разбивается на подзадачи закрепления серверов за клиентами и подзадачи размещения экземпляров приложений. Вторая подзадача может сводиться к тривиальному размещению всех примеров приложений на сервере, причем в случае виртуализации она составляет интересную задачу размещения приложений на ВМ. Но это тема отдельных исследований. Если клиент для доступа своих пользователей к приложениям, расположенным на группе серверов, использует виртуальный хостинг, то можно использовать модели и подходы, предложенные в работе [160].

Диспетчеризация при выделенных серверах также может свестись к жестким схемам направления запросов к выделенным серверам, но при виртуализации и виртуальном хостинге на группе выделенных для клиента серверов модели могут быть значительно интереснее. Начнем с моделей закрепления серверов за клиентами.

Модели планирования распределения ресурсов. Задачу закрепления серверов за клиентами целесообразно рассматривать как задачу определения матрицы $\mathbf{X} = ||x_{ji}||$, которая определяет привязку клиентов к серверам.

Оптимальное размещение должно учитывать следующие ограничения:

- суммарные требования γ_j, φ_j, δ_j, $j = \overline{1, p}$, независимые от нагрузки, определенных для сервера экземпляров приложений не превышают независимые от нагрузки параметры Γ_i, Φ_i, Δ_i сервера;
- требования ω_j, λ_j приложений, зависящие от загрузки, определяются на основе потребностей всех пользователей и должны быть обеспечены суммарными возможностям зависимых от загрузки параметров Ω_i, Λ_i тех серверов, на которых размещаются экземпляры этих приложений;
- ограничения, определенные матрицей *G*.

Критериями могут быть:

- максимальная суммарная важность приложений, получивших ресурсы;
- равномерная загрузка серверов;
- максимальное сокращение эксплуатационных расходов за счет высвобождения серверов;
- минимальная суммарная стоимость реализации операций перехода от предыдущего размещения к новому.

Очевидно, что речь должна идти об одной многокритериальной задаче. Но особенности критериев таковы, что более целесообразно рассматривать несколько моделей с различиями в обеспеченности ресурсами, чем одну многокритериальную задачу. При этом нужно выбрать одну из следующих моделей.

Модели планирования при ограниченных ресурсах. Для этого можно использовать модели, предложенные для распределения ограниченных ресурсов ИУС в работах [144, 148, 364, 367, 368].

Модели планирования при избытке ресурсов. В зависимости от стратегии хостинговой организации можно выбрать одну из моделей, которые различаются критериями.

Задача 1. Если кластеризация серверов не поддерживается и для хостинговой компании важна экономия ресурсов, то возникает задача минимизации расходов на поддержку той части серверов, которая будет обеспечивать обработку запросов пользователей клиента. Иначе говоря, выбирается такое минимальное по затратам на их поддержку подмножество серверов, которые смогут обеспечить выполнение требований приложений.

Считаем, что дисковая память организована в хранилищах SAN-системы, доступ к которым имеют все экземпляры приложений, на каких бы серверах они не размещались.

Обозначим через s_i эксплуатационные расходы на поддержку сервера N_i (электроэнергия, работы, комплектующие и т. д.). Тогда критерий этой задачи приобретет следующий вид:

$$\min \sum_{j=1}^{p} \sum_{i=1}^{n} s_i \cdot x_{ji}, \tag{8.20}$$

Минимальное значение надо выбирать из решений, которые соответствуют структуре булевых значений матрицы *G* и ограничениям (8.21)–(8.24):

$$\omega_j x_{ji} \le \Omega_i; \quad \lambda_j x_{ji} \le \Lambda_i; \quad \gamma_{\Sigma j} x_{ji} \le \Gamma_i; \quad \varphi_{\Sigma j} x_{ji} \le \Phi_i; \quad i = \overline{1,n}, \quad j = \overline{1,p}, \qquad (8.21)$$

требующему, чтобы потребности всех пользователей в емкости процессоров, каналов связи, ОЗУ и дисковой памяти удовлетворял сервер N_i;

$$\sum_{j=1}^{m} x_{ji} \le 1, \; i = \overline{1,n}, \qquad (8.22)$$

требующему, чтобы каждый сервер выделялся не более, чем для одного клиента;

$$\sum_{i=1}^{n} x_{ji} = 1, \; j = \overline{1,p}, \qquad (8.23)$$

требующему, чтобы каждому клиенту выделялся один сервер;

$$\delta_j x_{ji} \le \Delta_i, \; i = \overline{1,n}, \quad j = \overline{1,p}, \qquad (8.24)$$

требующему, чтобы надежность выделенного сервера N_i обеспечивала требования клиента K_j к надежности.

Задача 2. Если кластеризация серверов поддерживается и для хостинговой компании важна экономия ресурсов, то возникает задача минимизации расходов на поддержку той части серверов, которая будет обеспечивать обработку запросов пользователей клиента. Иначе говоря, выбирается такое минимальное по затратам на их поддержку подмножество серверов, которые смогут поддержать необходимые для удовлетворения требований существующих пользователей приложений, причем каждому клиенту выделяется один физический сервер или кластер серверов.

Критерий задачи имеет вид (8.20).

Минимальное значение выбирается из решений, соответствующих структуре булевых значений матрицы G и ограничениям (8.25)– (8.31).

Ограничение (8.25) требует, чтобы сервер N_i или кластер удовлетворял требованиям всех пользователей приложений клиента K_j, соответственно, в емкости процессора, каналов связи, и суммарным требованиям $\gamma_{\Sigma j}$ к емкости ОЗУ всех приложений клиента K_j, если речь идет о кластере, создаваемом для повышения производительности. Здесь π – коэффициент падения суммарной процессорной емкости кластера за счет необходимости поддержания взаимодействия серверов кластера; α – коэффициент, учитывающий дополнительные затраты ОЗУ при размещении приложений на нескольких серверах:

$$\pi \cdot \sum_{i=1}^{n} \Omega_i x_{ji} \ge \omega_j; \quad \sum_{i=1}^{n} \Lambda_i x_{ji} \ge \lambda_j; \quad \sum_{i=1}^{n} \Gamma_i x_{ji} \ge \gamma_{\sum j} \cdot (1+\alpha), \quad j = \overline{1,p}. \qquad (8.25)$$

Ограничение (8.26) требует, чтобы сервер N_i или каждый сервер кластера удовлетворял суммарным требованиям $\gamma_{\Sigma j}$ к емкости ОЗУ всех приложений клиента K_j, если речь идет о кластере, создаваемом с целью повышения надежности:

$$\gamma_{\sum j} x_{ji} \le \Gamma_i, \; i = \overline{1,n}, \; j = \overline{1,p}. \qquad (8.26)$$

АИС КП

Ограничение (8.27) требует, чтобы сервер N_i или один из серверов кластера удовлетворял требованиям γ_{maxj} самого требовательного к ОЗУ приложения клиента K_j, если речь идет о кластере, создаваемом для повышения производительности:

$$\Gamma_1 x_{j1} \geq \gamma_{\max j} \vee \Gamma_2 x_{j2} \geq \gamma_{\max j} \vee \ldots \vee \Gamma_n x_{jn} \geq \gamma_{\max j}, \quad j = \overline{1, p}. \tag{8.27}$$

где β – коэффициент, учитывающий дополнительные затраты памяти на жестких дисках при размещении приложений на нескольких серверах.

Ограничение (8.28) требует, чтобы сервер N_i или кластер удовлетворял суммарным требованиям $\varphi_{\Sigma j}$ к емкости памяти жестких дисков всех приложений клиента K_j, если речь идет о кластере, создаваемом для повышения производительности:

$$\sum_{i=1}^{n} \Phi_i x_{ji} \geq \varphi_{\Sigma j}(1+\beta), \quad j = \overline{1, p}, \tag{8.28}$$

Ограничение (8.29) требует, чтобы выделенный сервер N_i или каждый сервер кластера удовлетворял суммарным требованиям $\varphi_{\Sigma j}$ к емкости памяти жестких дисков всех приложений клиента K_j, если речь идет о кластере, создаваемом с целью повышения надежности:

$$\varphi_{\sum_j} x_{ji} \leq \Phi_i, \quad i = \overline{1, n}, \quad j = \overline{1, p}. \tag{8.29}$$

Ограничение (8.30) требует, чтобы выделенный сервер N_i удовлетворял требованиям φ_{maxj} самого требовательного к памяти дисков приложения клиента K_j:

$$\Phi_1 x_{j1} \geq \varphi_{\max j} \vee \Phi_2 x_{j2} \geq \varphi_{\max j} \vee \ldots \vee \Phi_n x_{jn} \geq \varphi_{\max j}, \quad j = \overline{1, p}. \tag{8.30}$$

Должны выполняться ограничения (8.22) и (8.23).

Ограничение (8.31) требует, чтобы выделенный сервер N_i или кластер удовлетворял требованиям клиента K_j к надежности. Здесь функция f определяет подход к определению надежности кластера в зависимости от надежности серверов:

$$f(\Delta_1 x_{j1}, \ldots, \Delta_n x_{jn}) \geq \delta_j, \quad j = \overline{1, p}. \tag{8.31}$$

Для оценки надежности сервера (кластера) будем применять коэффициент готовности сервера (кластера). В зависимости от того, с какой целью создаются кластеры, ограничение (8.31) может принимать следующий вид:

1) кластер создается для обеспечения надежности. При кластере на двух серверах значение функции f можно определить как

$$f(\Delta_1, \Delta_2) = \Delta_1 + \Delta_2 - \Delta_1 \cdot \Delta_2, \tag{8.32}$$

где Δ_1 и Δ_2 – коэффициенты готовности сервера.

Требования приложений клиента K_j к емкости ОЗУ и процессорной емкости должны выполняться для каждого сервера кластера;

2) кластер создается для повышения производительности. Производительность кластера определяется по левой части неравенства ограничения (8.25). Кроме того, должны выполняться ограничения (8.27), (8.28) и (8.30);

3) кластер создается для обеспечения производительности и надежности по схеме $n + 1$, когда n серверов увеличивают производительность и один сервер используется для повышения надежности. Например, на рис. 8.6 приведен вариант схемы $n + 1$ при $n = 1$ с приложениями A_1 и A_2. Кластер используется для повышения надежности приложения A_1 и увеличения производительности приложения A_2. Надежность приложения A_1 определяется надежностью кластера, а надежность приложения A_2 – надежностью одновременного функционирования обоих серверов.

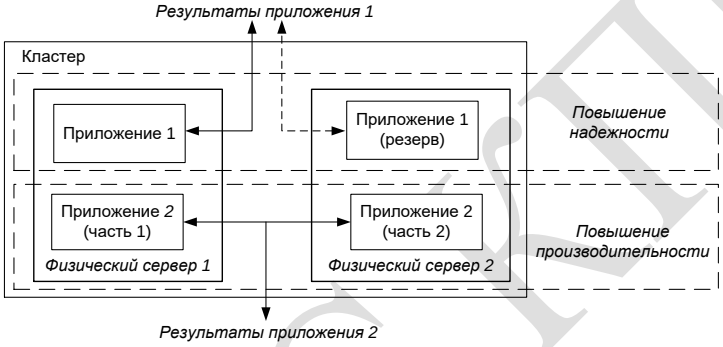


Рис. 8.6. Вариант схемы $n + 1$ с двумя приложениями

Задача 2′. Интересный случай представляет задача, когда за клиентом с целью повышения производительности путем балансировки нагрузки закрепляется несколько серверов, на которых устанавливается одинаковый набор экземпляров приложений. В этом случае требования к объему ресурсов, которые не зависят от нагрузки, должны удовлетворяться на каждом из серверов, а требования к ресурсам, зависящим от нагрузки, должны обеспечиваться суммарной емкостью ресурсов этих серверов:

$$\pi\sum_{i=1}^{n}\Omega_i x_{ji} \geq \omega_j\,,\ \ j=\overline{1,p}\,. \tag{8.33}$$

Ограничение (8.33) требует, чтобы потребности всех приложений клиента K_j в процессорном времени удовлетворялись суммарными ресурсами серверов;

$$\sum_{i=1}^{n}\Lambda_i x_{ji} \geq \lambda_j\,,\ \ j=\overline{1,p}\,. \tag{8.34}$$

Ограничение (8.34) требует, чтобы суммарная емкость каналов связи, оперативной и дисковой памяти выделенных для пользователя K_j серверов обеспечивала работу всех приложений. Взяв критерий (8.20) и ограничения (8.21)–(8.23), (8.25)–(8.31), (8.33), (8.34), получим линейную или нелинейную задачу булевого программирования в зависимости от наличия требований клиентов к надежности.

Аутсорсинг

Задача 3. Если для хостинговой компании важна экономия ресурсов, а затраты на реализацию перехода к новому оптимальному плану закрепления серверов сравниваемы с расходами на поддержку серверов, то возникает задача минимизации суммарных затрат на поддержку серверов и реализацию перехода к новому плану закрепления серверов. При этом выбирается такое минимальное по сумме расходов на реализацию перехода к новому оптимальному плану закрепления серверов и на их поддержку подмножество серверов, которые смогут поддержать необходимое для удовлетворения требований существующих пользователей множество экземпляров приложений. С использованием введенных выше обозначений критерий этой задачи можно записать в виде

$$\min\left(\sum_{i=1}^{n}\left(\sum_{j=1}^{p} s_i \cdot x_{ji} + \sum_{j=1}^{p} c_{ji}\left|x'_{ji} - x_{ji}\right|\right)\right). \qquad (8.35)$$

В выражении (8.35) c_{ji} обозначает затраты, связанные с переходом к новому оптимальному плана закрепления серверов в части, касающейся клиента K_j и сервера N_i, x'_{ji} – значение переменной x_{ji} в предыдущем плане. Расходы c_{ji} учитываются, если значение нового и предыдущего планов в ji-й позиции различаются.

Минимальное значение необходимо выбирать из решений, соответствующих структуре значений матрицы G и удовлетворяющих ограничениям (8.25)–(8.31) с учетом замечаний по выбору ограничений для различных подходов к созданию кластеров, приведенных в постановке задачи 2.

Приведенные выше задачи оптимизации размещений – это линейные и нелинейные варианты задачи булевого программирования. Поскольку обойти переборные методы их решения нельзя, оправданным будет использование эвристических методов или генетических алгоритмов [139, 169, 281], а также декомпозиции и аффинно-масштабирующих алгоритмов [30, 34, 68].

Таким образом, разработаны модели закрепления за клиентами физических серверов при различных критериях эффективности с учетом ограничений.

8.3.4. Алгоритмы решения задач оптимального размещения приложений

Для решения указанных выше задач оптимального размещения приложений предлагаются варианты ГА и эвристического алгоритма, который учитывает специфику критериев и ограничений. Кроме того, предлагается комбинированный алгоритм, построенный с использованием методов решения задач линейного программирования и неявного перебора.

Варианты генетического алгоритма. Рассмотрим универсальный вариант такого алгоритма для задачи 2, которая отличается нелинейностью ограничений. Этот вариант можно довольно легко модифицировать для каждой из приведенных задач.

Шаг 0. Задаем точность вычислений σ. Устанавливаем $l = 0$. Формируем случайным образом начальную популяцию $P_k(0) = \{p_1(0),...,p_k(0)\}$ с учетом возможного отсеивания при проверке ограничений.

Шаг 1. Проверяем для каждой цепочки популяции выполнение ограничений соответствующей задачи, например, для задачи 2 – ограничений (8.25)–(8.31). Для цепочек $p_q(0)$, для которых ограничения выполняются, определяется оценка $f(p_q(0))$ целевой функции соответствующей задачи и выбор наилучшего решения.

Шаг 2. Отбираем представителей для новой популяции (цепочек с лучшим значением функции $f(p_q(0))$.

Шаг 3. $L = l + 1$. Создаем новую популяцию, применяя оператор рекомбинации.

Шаг 4. Улучшаем популяцию применением оператора мутации.

Шаг 5. Проверяем для каждой цепочки популяции выполнение ограничений и оцениваем функцию $f(p_q(l))$, если ограничения выполняются.

Шаг 6. Если полученное решение лучше предыдущего с заданной точностью σ, то завершаем выполнение алгоритма, в противном случае возвращаемся на шаг 3.

Выбирается лучшее решение в образовавшейся популяции $P_k(l)$.

В приведенном варианте ГА каждая цепочка (особь) любой популяции составляет двоичный вектор длины $t = m \times n$, определяемой количеством физических серверов n и клиентов m. Элемент q вектора приобретает значение 0 или 1 соответствующих переменных x_{ji}. Длина цепочек постоянная.

Отбор особи для новой популяции выполняется на основе ранжирования по значению целевой функции особей предыдущей популяции, причем ранжируются лишь особи, которые удовлетворяют ресурсным и другим ограничениям.

Для рекомбинации применяется одноточечный кроссовер со случайным выбором точки кроссовера. Реализация мутаций осуществляется на основе схемы инверсии участков цепочек.

Варианты эвристического алгоритма. Эти алгоритмы основаны на интуитивных представлениях, которые учитывают роль зависимых и независимых ресурсов и критерии. Поскольку процессорную емкость можно обеспечить совокупно несколькими серверами, а ОЗУ – только одним сервером, то будет логичным сначала разместить приложения, имеющие большие потребности в ОЗУ, чем в процессорном времени. Взяв это за правило, далее необходимо лишь учесть особенности критерия. Каждому физическому серверу N_i припишем коэффициент P_i, в качестве значения которого примем отношение процессорной емкости сервера N_i к емкости его ОЗУ, каждому клиенту K_j – коэффициент ρ_j, значением которого является отношение требований приложений клиента K_j к процессорной емкости к суммарным требованиям $\gamma_{\Sigma j}$ приложений клиента K_j к емкости ОЗУ.

Продемонстрируем вариант эвристического алгоритма на примере задачи 2. С очевидными оговорками его можно модифицировать для других задач.

Предварительный этап для любого из следующих вариантов эвристического алгоритма заключается в сортировке физических серверов по значению их коэффициента P в порядке возрастания и сортировке клиентов по значению их коэффициента ρ в порядке убывания.

Эвристический алгоритм для задачи 2. Предварительный этап выполнен.

Шаг 1. Поиск клиента, который имеет наименьшее значение коэффициента ρ.

Шаг 2. Поиск сервера с наибольшим значением коэффициента P.

Шаг 3. Если для выбранных сервера и клиента не выполняются ограничения задачи 2 относительно требований $\gamma_{\max j}$ самого требовательного по ОЗУ приложения клиента K_j, то возвращаемся на шаг 2 для поиска следующего сервера.

Шаг 4. Если для избранных сервера и клиента не выполняются другие ресурсные ограничения задачи 2, то возвращаемся на шаг 2 для поиска следующего сервера.

Шаг 5. Если для выбранных сервера и клиента выполняются ограничения задачи 2, причем зависимые от нагрузки требования полностью удовлетворены, то

переходим на шаг 6. Если выполняются ограничения задачи 2, но сервер не может полностью удовлетворить зависимые от нагрузки требования применения, то переходим на шаг 7.

Шаг 6. Клиент исключается из упорядоченного списка клиентов, корректируется коэффициент P сервера и он встраивается в упорядоченный список серверов согласно новому значению коэффициента. Переход на шаг 8.

Шаг 7. Оставшиеся ресурсы сервера достаются приложениям клиента и сервер изымается из упорядоченного списка серверов. Переходим на шаг 2.

Шаг 8. Если упорядоченный список клиентов пуст, то конец работы алгоритма, иначе переход на шаг 1.

Вычислительная сложность эвристического алгоритма составляет $m \times n$.

Для других критериев используется другая логика выбора серверов в их упорядоченном списке.

Комбинированный алгоритм. Для линейных задач булевого программирования, к которым сведены задачи закрепления серверов за клиентами, предложен ряд методов, в первую очередь неявного перебора, которые используют для сокращения количества перебираемых вариантов, кроме известных приемов, удобный вид ограничений задачи. Однако специфика матрицы ограничений задачи 1 не слишком способствует широкому применению традиционных приемов сокращения перебора. В то же время она требует поиска комбинированного подхода к решению задачи. Сравнительно небольшое количество ограничений предусматривает применение эффективных методов решения задачи линейного программирования, поскольку ее решения будут содержать небольшое количество нецелочисленное компонентов. Схема перебора применяется именно к указанным нецелочисленным компонентам. Именно такой подход с обнадеживающими результатами был применен к линейным задачам булевого программирования в работе [81]. В основу подхода положена схема, которая включает следующие этапы:

1) решение задачи 1 без условий целочисленности переменных;

2) подстановка целочисленных компонентов полученного решения в исходную задачу;

3) решение полученной задачи методом неявного перебора.

К сожалению, увеличение количества переменных решения задачи первого этапа усложняется, поскольку резко возрастают требования к памяти и объему вычислений. Современное состояние линейного программирования позволяет решать задачи значительно большей размерности с помощью декомпозиции и аффинно-масштабирующих алгоритмов [30, 34, 68].

8.4. МОДЕЛИ И МЕТОДЫ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В СИСТЕМАХ С СЕРВЕРНОЙ ВИРТУАЛИЗАЦИЕЙ С УЧЕТОМ ВАЖНОСТИ ПРИЛОЖЕНИЙ

8.4.1. Особенности виртуализации серверов

В последнее время при создании серверных комнат или ЦОД для корпоративных ИУС наблюдается тенденция сокращения количества физических серверов для поддержки различных сервисов путем внедрения технологии виртуализации серверов [288]. Такой подход за счет большей загрузки серверов позволяет не только уменьшить их количество, но и сократить расходы на энергоснабжение, поддержание условий микроклимата, уменьшить площади для размещения

серверов и т. д. Таким образом удается уменьшить совокупную стоимость владения ИТ-инфраструктурой.

Однако более плотное расположение приложений на серверах значительно повышает риск возникновения дефицита ресурсов в случае увеличения потока запросов пользователей. Поэтому актуальной становится задача по управлению распределением ресурсов таким образом, чтобы в первую очередь обеспечить функционирование приложений, связанных с наиболее важными бизнес-процессами даже за счет менее важных [114, 144, 148, 149, 154, 157]. Для решения средствами СУИ задач управления виртуализацией серверов требуется разработка соответствующих моделей и методов или адаптация уже известных с учетом виртуализации серверов [146].

Особенности применения виртуализации серверов. Виртуализация серверов может осуществляться одним из следующих способов:

- полная виртуализация;
- паравиртуализация;
- виртуализация на уровне операционной системы (ОС).

Принцип полной виртуализации заключается в том, что для гостевых ОС осуществляется полная имитация аппаратной среды средствами гипервизора, которая выполняется поверх реального аппаратного обеспечения хоста или поверх хостовой ОС. Для такого подхода характерно незначительное падение производительности вследствие эмуляции системных вызовов гостевой ОС к оборудованию.

При паравиртуализации гостевые ОС перед использованием подлежат модификации, в результате которой системные вызовы к аппаратному обеспечению заменяются вызовами к гипервизору. Такой подход, с одной стороны, позволяет уменьшить дополнительные затраты ресурсов на эмуляцию системных вызовов, но с другой – требует внесения изменений в код гостевой ОС, что не всегда возможно (например, для ОС с закрытым кодом).

Виртуализация на уровне ОС позволяет создавать в ОС изолированные друг от друга области (так называемые «jails» в FreeBSD, «containers» в Solaris, «виртуальные серверы» в Linux-VServer), что позволяет значительно уменьшить затраты ОЗУ по сравнению с описанными выше способами, поскольку исчезает необходимость в отдельных экземплярах гостевых ОС для каждого из виртуальных серверов. Недостатком такого подхода является то, что гостевая ОС не может отличаться от хостовой ОС.

В случае использования технологий виртуализации СУИ должна предоставлять возможности планировать и управлять ресурсами и нагрузкой серверов с учетом особенностей бизнес-процессов, выбранного способа виртуализации и других важных факторов.

Далее рассматривается серверная виртуализация, построенная на принципах полной виртуализации или паравиртуализации. Особенности управления распределением ресурсов в этом случае заключаются в возможности освобождать не только ресурсы, занятые непосредственно приложениями, связанными с низкоприоритетными бизнес-процессами, но и ресурсы, затрачиваемые на функционирование гостевых операционных систем виртуальных серверов.

Анализ существующих решений. Современные системы виртуализации в той или иной степени обеспечивают возможность распределения ресурсов между

ВМ. Например, VMware ESX позволяет назначать основные параметры ВМ (тактовая частота процессора, объем ОЗУ, емкость жесткого диска и т. д.), а в процессе работы их перераспределять.

Процессорное время может автоматически перераспределяться в случае, если одна из ВМ простаивает (процессор находится в состоянии простоя), в то время как другой машине не хватает вычислительных ресурсов.

Оперативная память, до недавнего времени жестко закрепляемая за ВМ перед ее запуском, теперь также может перераспределяться динамично за счет механизма, который получил название Memory Overcommitment. Сущность его заключается в том, что для ВМ кроме параметра «reservation», который устанавливает нижний предел объема ОЗУ, назначается параметр «limit», который устанавливает верхний предел. При этом суммарное значение параметров «limit» может превышать объем ОЗУ сервера. ВМ, которая требует дополнительного объема ОЗУ, память придается за счет других ВМ, в которых память используется не очень интенсивно, хранится в файле подкачки с помощью программы Memory Balloon Driver (MBD). MBD выполняется в гостевой ОС и по требованию гипервизора захватывает физическую память, которая затем перераспределяется между другими ВМ. При этом гостевая ОС, в случае недостатка физической памяти, хранит в файле подкачки часть своей памяти, которая меньше используется. К сожалению, не все современные гипервизоры могут динамические перераспределять ресурсы между ВМ. Например, в Microsoft Hyper-V возможность «Memory Overcommitment» отсутствует.

Существующие алгоритмы распределения ресурсов хорошо себя зарекомендовали для случаев, когда на физическом сервере функционируют независимые друг от друга ВМ. В случае, когда в разных виртуальных серверах выполняются приложения, которые поддерживают выполнение отдельных частей единого бизнес-процесса, такой подход может быть очень неэффективным. Неизвестны эффективно реализованные механизмы активного планирования и управления распределением ресурсов серверов между ВМ и нагрузки пользователей с учетом их важности.

Постановка задачи. Рассмотрим следующий случай:

- ресурсы одного физического сервера разделяются несколькими ВМ;
- применяется технология полной виртуализации или паравиртуализации;
- на одной ВМ развернут сервер БД, а на других ВМ работают серверы приложений, которые используют указанный сервер БД в другой ВМ (такое может случиться, если, например, эти серверы требуют различные ОС).

При увеличении количества клиентских запросов к серверу приложений соответственно увеличится и количество запросов от сервера приложений к серверу БД. Тогда возникает необходимость увеличения ресурсов обеих ВМ за счет остальных ВМ. Увеличение ресурсов лишь одной из этих ВМ не приведет к значительному повышению эффективности, поскольку другая будет «узким местом». Увеличение ресурсов одной ВМ за счет другой, даже наоборот, может привести к уменьшению общей эффективности трехуровневых приложений и в результате – к еще большему обострению проблемы. К тому же следует учитывать, что некоторые приложения могут быть задействованы для поддержки нескольких бизнес-процессов.

Необходимо разработать модели планирования и управления ресурсами физического сервера и нагрузкой ВМ с учетом важности приложений, ресурсных,

технологических и других существенных ограничений. В такой постановке задача актуальна только для корпоративных ЦОД, при управлении ресурсами которых СУИ может учесть известные значимости приложений, поддерживающих бизнес-процессы. Коммерческие ЦОД обязаны предоставлять обслуживание запросов всех пользователей, независимо от значимости приложений.

8.4.2. Особенности СУИ для систем с виртуализацией серверов

Учитывая вышеизложенное, предлагается строить часть СУИ, осуществляющую распределение ресурсов сервера, по такой двухуровневой схеме:

1-й уровень: управление отдельными виртуальными серверами;

2-й уровень: управление отдельными приложениями, работающими в среде виртуальных серверов.

Схема взаимодействия СУИ с системой виртуализации изображена на рис. 8.7.

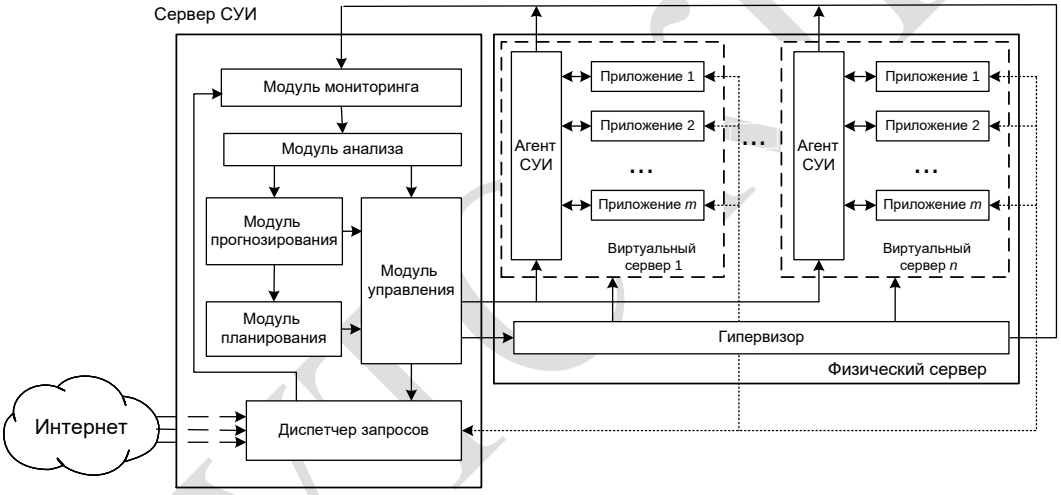


Рис. 8.7. Взаимодействие СУИ с системой виртуализации

На физическом сервере под управлением гипервизора работает *n* виртуальных серверов. В каждом из них помимо клиентских приложений установлена гостевая ОС – приложение 1 и агент СУИ, который осуществляет мониторинг и управление приложениями виртуального сервера и распределяет ресурсы виртуального сервера между приложениями средствами ОС и самих приложений (при наличии у них соответствующих API).

Информация от агентов СУИ и гипервизора попадает в модули мониторинга и анализа.

Модуль управления предназначен для выработки команд управления агентам СУИ (запуск/остановка приложений, распределение ресурсов виртуального сервера между приложениями), гипервизору (запуск/остановка виртуальных серверов, распределение ресурсов физического сервера между виртуальными серверами) и диспетчеру запросов (управление нагрузкой).

Модуль прогнозирования осуществляет анализ трендов с целью дальнейшей реализации краткосрочного проактивного управления, в данном случае – перераспределения ресурсов между приложениями до того, как возникла их нехватка

для приложений, задействованных в поддержке бизнес-процессов высокой важности.

Модуль планирования предназначен для долгосрочного управления ресурсами ИТ-инфраструктуры при ее развитии или существенных изменениях в ее функционировании.

Диспетчер запросов предназначен для ограничения клиентского трафика заданного типа с целью высвобождения ресурсов, потребление которых зависит от нагрузки.

На рис. 8.7 СУИ изображена отдельно от физического сервера, которым она управляет, но она может быть размещена на одном из виртуальных серверов при наличии для этого свободных ресурсов.

8.4.3. Математическая модель серверной виртуализации

Пусть имеется сервер, на котором под управлением гипервизора функционирует несколько виртуальных серверов S_i, $i = \overline{1,n}$ с использованием технологии полной виртуализации или паравиртуализации. На виртуальном сервере могут быть развернуты несколько приложений, которые могут бороться за ресурсы, и агент СУИ. Виртуальный сервер эмулирует работу отдельного физического сервера.

В каждом из виртуальных серверов S_i установлена своя операционная система (A_{i1}), а также одно или несколько приложений A_{ij}, где $j = 2,...,m_i$; $m_i > 1$. Будем считать, что ресурсы, необходимые для функционирования самого виртуального сервера и агента СУИ, отнесены к A_{i1}.

Каждое из приложений в зависимости от потока клиентских запросов использует определенное количество ресурсов типа R_k, $k = \overline{1,l}$ (память, процессорное время, дисковое пространство, пропускная способность внешних каналов связи и т. п.). Схема распределения ресурсов физического сервера между виртуальными серверами приведена на рис. 8.8.

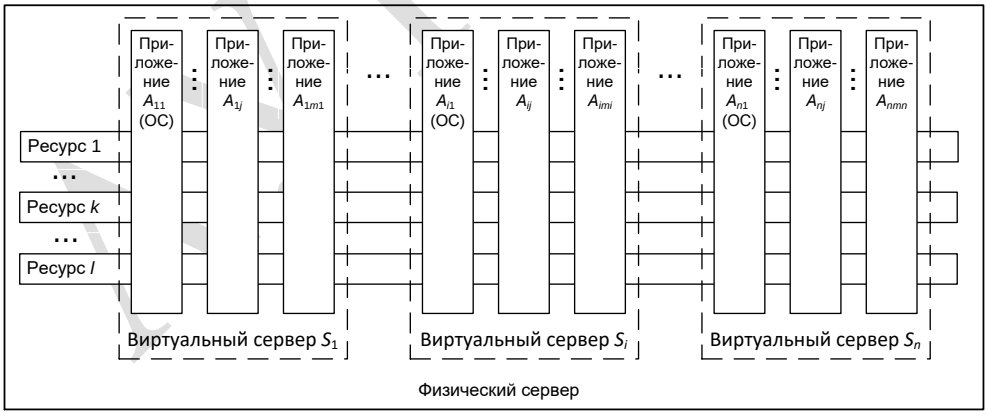


Рис. 8.8. Схема распределения ресурсов физического сервера между виртуальными серверами

Введем необходимые для построения моделей обозначения:
r_k – количество ресурсов типа R_k, установленных на физическом сервере,
r_{ki} – количество ресурсов типа R_k, предназначенных для виртуального сервера S_i,
r_{kij} – количество ресурсов типа R_k, потребляемое приложением A_{ij}.
Очевидно, что должно выполняться следующее условие:

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} r_{kij} \le r_k\ ,\ k=\overline{1,l}\ . \tag{8.36}$$

В зависимости от реализации системы виртуализации для ресурсов некоторых типов невозможно динамически осуществлять их перераспределение между виртуальными серверами в результате жесткого закрепления этих ресурсов за виртуальными серверами (наиболее распространенный случай – объем ОЗУ, который закрепляется за виртуальным сервером перед его запуском). Тогда для таких ресурсов кроме условия (8.36) должны выполняться условия

$$\sum_{j=1}^{m_i} r_{kij} \le r_{ki}\ ,\ i=\overline{1,n}\,, \tag{8.37}$$

и

$$\sum_{i=1}^{n} r_{ki} \le r_k\ . \tag{8.38}$$

Потребности приложения A_{ij} в ресурсах типа k выразим с помощью формулы

$$r_{kij} = b_{kij}\, a_{ij} + c_{kij}, \tag{8.39}$$

где a_{ij} – среднее количество клиентских запросов к приложению A_{ij} за единицу времени; b_{kij} – среднее количество ресурсов типа k, используемых приложением A_{ij} на обработку одного клиентского запроса; c_{kij} – количество ресурсов типа k, используемых приложением A_{ij}, независимо от количества клиентских запросов.

В зависимости от особенностей приложений можно использовать упрощенный вид формулы (8.39). Так, затраты процессорного времени большинство серверных приложений или использования пропускной способности каналов связи напрямую зависят от количества клиентских запросов и при их отсутствии приблизительно равны нулю. Тогда формула (8.39) примет вид

$$r_{kij} = b_{kij}\, a_{ij}. \tag{8.40}$$

Потребности приложения A_{ij} в ОЗУ в общем случае можно выразить формулой (8.39), поскольку даже при отсутствии клиентских запросов ПО сервера использует определенный объем ОЗУ.

Иногда ПО использует фиксированный объем памяти, указываемый администраторами в файлах конфигурации. Эта память используется для хранения кода серверного ПО и данных, обрабатываемых в настоящее время. Остальная память используется для кэширования. В этом случае формула потребления ресурсов (8.39) примет такой упрощенный вид:

$$r_{kij} = c_{kij}. \tag{8.41}$$

Введем понятие поддержки бизнес-процесса. Под поддержкой бизнес-процесса Z_p будем понимать взаимодействие нескольких приложений, направленное на достижение единого результата.

Пусть сервер используется для поддержки z бизнес-процессов.

Введем переменную x_p, $p=\overline{1,z}$, принимающую значение:

$$x_p = \begin{cases} 1, & \text{если бизнес-процесс поддерживается в полном объеме;} \\ 0, & \text{если бизнес-процесс не поддерживается.} \end{cases} \quad (8.42)$$

Введем понятие важности w_p бизнес-процесса Z_p, $p = \overline{1,z}$, которое будем использовать при выборе бизнес-процессов в условиях нехватки ресурсов.

Тогда в качестве критерия можно использовать

$$\sum_{p=1}^{z} x_p w_p \to \max . \quad (8.43)$$

Пусть бизнес-процесс Z_p осуществляет в среднем a_{ijp} клиентских запросов к приложению A_{ij}. Тогда общее количество клиентских запросов этого приложения можно определить с помощью выражения

$$a_{ij} = \sum_{p=1}^{z} a_{ijp} x_p . \quad (8.44)$$

Это позволяет выразить потребности ресурса r_{kij} в общем случае следующей формулой:

$$r_{kij} = \sum_{p=1}^{z} a_{ijp} x_p b_{kij} + c_{kij} , \; k = \overline{1,l} . \quad (8.45)$$

Используя приведенные обозначения, сформулируем задачи управления распределением ресурсов на этапах планирования, управления и диспетчеризации.

Группа задач планирования. В зависимости от того, определена ли конфигурация сервера при решении задачи планирования, возможны две ситуации:

1) если конфигурация физического сервера еще не определена, то стоит задача ее выбора и распределения ресурсов между виртуальными серверами и приложениями;

2) конфигурация физического сервера уже известна и стоит задача распределения ресурсов между виртуальными серверами и приложениями.

Задача планирования 1. В первом случае на основе системных требований приложений к ресурсам различных типов (b_{kij}, c_{kij}), ожидаемой интенсивности клиентских запросов (a_{ij}) и запланированной степени поддержки бизнес-процессов ($x_p = 1$, или с учетом плана развития $x_p > 1 + f_p$) с помощью формулы (8.45) рассчитывается необходимое количество ресурсов каждого типа r_{kij}. После этого с помощью формулы (8.36) рассчитываются общие объемы ресурсов сервера r_k каждого типа, которые затем могут быть уточнены экспертами.

Группа задач планирования 2. В этом случае конфигурация сервера определена, т. е. значения r_k уже известны, и нужно найти такое распределение ресурсов между виртуальными серверами (r_{ki}) и приложениями (r_{kij}), которое обеспечит наиболее эффективное поддержание бизнес-процессов.

В зависимости от конфигурации сервера и системных требований приложений на этом этапе может возникнуть ситуация с дефицитом или избытком ресурсов.

Задача планирования 2.1. В случае возникновения дефицита ресурсов нельзя найти такое распределение r_{ki} и r_{kij}, чтобы обеспечить поддержание всех

бизнес-процессов. Иными словами, все переменные x_p, $p = \overline{1,z}$, не могут получить значение 1. Тогда единственным выходом становится временный отказ от поддержания отдельных низкоприоритетных бизнес-процессов для обеспечения ресурсами более приоритетных бизнес-процессов. В этом случае задача заключается в том, чтобы найти такие значения поддержания бизнес-процессов, которые обеспечивают наилучшее значение критерия (8.43), а затем, на основе выражений (8.37) и (8.45), найти нужные значения r_{ki} и r_{kij}. Поскольку значения r_{ki} и r_{kij} не закреплены, на этапе максимизации используются те же ограничения, что и в задаче управления 3 (см. ниже).

Группа задач планирования 2.2. В случае возникновения избытка ресурсов актуальной становится задача их наиболее эффективного использования. Одним из возможных вариантов является создание резерва вычислительных мощностей на случай резкого возрастания интенсивности клиентских запросов или постепенного увеличения запросов в процессе эксплуатации ИУС. Отметим, что в обеих задачах этой группы переменная x_p является непрерывной.

Задача планирования 2.2.1. При наличии данных о прогнозируемом увеличении активности отдельных бизнес-процессов нужно найти наилучшее решение согласно критерия (8.43) при ограничениях (8.36), (8.45) и дополнительных ограничениях $x_p \geq 1 + f_p$, для $p = \overline{1,z}$, где f_p – прогнозируемое увеличение поддержки бизнес-процесса Z_p, не приводящее к возникновению дефицита ресурсов.

Задача планирования 2.2.2. Если данные прогноза относительно возможного увеличения активности отдельных бизнес-процессов отсутствуют, вполне естественно возникает задача создания равномерного запаса для всех бизнес-процессов.

Для этого на первом этапе выдвигаем дополнительное требование $x_1 = x_2 = x_3 = ... = x_Z \geq 1$ и находим наилучшее решение согласно критерию (8.43) при ограничениях (8.36), (8.45).

В результате получаем решение $x_1 = x_2 = x_3 = ... = x'$.

Для наиболее эффективного распределения остатка ресурсов, оставшихся после получения на первом этапе решения, еще раз находим наилучшее решение согласно критерию (8.43) при ограничениях (8.36), (8.45), и дополнительном ограничении $x_p \geq x'$.

Группа задач управления. Задачи управления решаются на этапе функционирования ИУС. В зависимости от того, имеется ли возможность останавливать отдельные приложения и виртуальные серверы с целью высвобождения ресурсов и их последующего перераспределения, сформулируем ряд задач.

Задача управления 1. Пусть вследствие технических или административных ограничений нет возможности останавливать приложения и виртуальные серверы с целью освобождения их ресурсов. Тогда задачу можно сформулировать следующим образом.

Найти наилучшее решение согласно критерия (8.43) при ограничениях вида (8.36), (8.45) для ресурсов, которые можно динамически перераспределять между виртуальными серверами, и вида (8.37), (8.38), (8.45) для ресурсов, которые нельзя динамически перераспределять между виртуальными серверами.

После того, как оптимальные значения переменных x_p найдены, они передаются диспетчеру запросов для управления нагрузкой. Далее на основе известных значений переменных x_p и системных требований приложений по формулам

(8.45), (8.36) и (8.45), (8.37), (8.38) рассчитываются ограничения на использование ресурсов для каждого приложения и виртуального сервера, которые устанавливаются средствами агентов СУИ для приложений и средствами гипервизора для виртуальных серверов.

Задача управления 2. Пусть существует возможность останавливать приложения, но нет возможности (вследствие технических или административных ограничений) останавливать виртуальные серверы с целью высвобождения ресурсов.

В этом случае при недостатке ресурсов имеет смысл держать приложение в памяти только в том случае, если оно необходимо для поддержания хотя бы одного из бизнес-процессов, которые выбраны для поддержания. Иными словами, вместо выражения (8.45) следует использовать такое выражение для оценки количества ресурсов, потребляемых приложением A_{ij}:

$$r_{kij} = \sum_{p=1}^{z} a_{ijp} x_p b_{kij} + C_{kij} \ , \tag{8.46}$$

где

$$C_{lij} = \begin{cases} c_{kij}, \text{ если } \sum_{p=1}^{z} a_{ijp} x_p > 0; \\ 0, \text{ если } \sum_{p=1}^{z} a_{ijp} x_p = 0. \end{cases} \tag{8.47}$$

Тогда задачу можно сформулировать следующим образом.

Найти наилучшее решение согласно критерию (8.43) при ограничениях вида (8.36), (8.46), (8.47) для ресурсов, которые можно динамически перераспределять между виртуальными серверами, и вида (8.37), (8.38), (8.46), (8.47) для ресурсов, которые нельзя перераспределять между виртуальными серверами.

После того, как оптимальные значения переменных x_p найдены, они передаются диспетчеру запросов для управления нагрузкой. Далее на основе известных значений переменных x_p и системных требований приложений по формулам (8.46), (8.47), (8.36) и (8.46), (8.47), (8.37), (8.38) рассчитываются ограничения на использование ресурсов для каждого приложения и виртуального сервера, которые устанавливаются средствами агентов СУИ для приложений и средствами гипервизора для виртуальных серверов. Приложения, для которых выполняется условие $\sum_{p=1}^{z} a_{ijp} x_p = 0$, средствами агента СУИ приостанавливаются с целью высвобождения их ресурсов.

Задача управления 3. Пусть существует возможность останавливать приложения и виртуальные серверы с целью высвобождения их ресурсов.

В этом случае при недостатке ресурсов имеет смысл держать приложение в памяти только в том случае, если оно необходимо для поддержания хотя бы одного из выбранных для поддержания бизнес-процессов. Виртуальный сервер имеет смысл держать в памяти только в том случае, если в нем выполняется хотя

АТС КП

бы одно приложение, необходимое для поддержания выбранных для поддержания бизнес-процессов. Тогда для ресурсов, которые можно перераспределять между виртуальными серверами, выражение (8.36) примет следующий вид:

$$\sum_{i=1}^{n}\sum_{j=2}^{m_i} r_{kij} + \sum_{i=1}^{n} r_{ki1} I_i \le r_k, \quad k = \overline{1,l}, \tag{8.48}$$

где

$$I_i = \begin{cases} 0, \text{ если } \sum\limits_{p=1}^{z}\sum\limits_{j=2}^{m_i} a_{ijp} x_p = 0; \\ 1, \text{ если } \sum\limits_{p=1}^{z}\sum\limits_{j=2}^{m_i} a_{ijp} x_p > 0. \end{cases} \tag{8.49}$$

В выражениях (8.48), (8.49) индекс j принимает значения, начиная с 2. Это означает, что функционирование виртуального сервера и его ОС является не целью, а лишь средством для функционирования приложений, участвующих в поддержке избранных бизнес-процессов. Наличие второго слагаемого в формуле (8.48) предусматривает возможность перераспределения ресурсов, которые используются определенными виртуальными серверами при отсутствии необходимости их функционирования.

Тогда задачу можно сформулировать следующим образом.

Найти наилучшее решение согласно критерию (8.43) при ограничениях (8.37), (8.38), (8.46), (8.49) для ресурсов, которые нельзя перераспределять между виртуальными серверами, и (8.46), (8.47), (8.48), (8.49) для ресурсов, которые можно перераспределять между виртуальными серверами.

После того, как оптимальные значения переменных x_p найдены, они передаются диспетчеру запросов для управления нагрузкой. Далее на основе известных значений переменных x_p и системных требований приложений по формулам (8.46), (8.47), (8.36) и (8.46), (8.47), (8.37), (8.38) рассчитываются ограничения на использование ресурсов для каждого приложения и виртуального сервера, которые устанавливаются средствами агентов СУИ для приложений и средствами гипервизора для виртуальных серверов. Приложения, для которых выполняется условие $\sum\limits_{p=1}^{z} a_{ijp} x_p = 0$, средствами агента СУИ останавливаются с целью освобождения их ресурсов.

Виртуальные серверы, для которых выполняется условие $\sum\limits_{p=1}^{z}\sum\limits_{j=2}^{m_i} a_{ijp} x_p = 0$, средствами гипервизора останавливаются с целью освобождения их ресурсов.

Задача управления 4. Пусть переменная x_p – непрерывная, а вследствие технических или административных ограничений не имеется возможности останавливать приложения и виртуальные сервера с целью освобождения их ресурсов.

Тогда задачу можно сформулировать следующим образом.

Архив ВШ

Найти наилучшее решение согласно критерию (8.43) при ограничениях вида (8.36), (8.45) для ресурсов, которые можно перераспределять между виртуальными серверами, и вида (8.37), (8.38), (8.45) для ресурсов, которые нельзя перераспределять между виртуальными серверами.

После того, как оптимальные значения переменных x_p найдены, они передаются диспетчеру запросов для управления нагрузкой. Далее на основе известных значений переменных x_p и системных требований приложений по формулам (8.45), (8.36) и (8.45), (8.37), (8.38) рассчитываются ограничения на использование ресурсов для каждого приложения и виртуального сервера, которые устанавливаются средствами агентов СУИ для приложений и средствами гипервизора для виртуальных серверов.

8.4.4. Методы решения задач планирования и управления

Задача планирования 1 является тривиальной и ее решение сводится к выполнению последовательности расчетов по приведенным выше формулам.

Задача планирования 2.1, а также задачи управления 2 и 3 являются нелинейными задачами булевого программирования и их решение при небольшом количестве z бизнес-процессов Z_p может осуществляться полным перебором возможных комбинаций (2^z), а при значительных z требует использования эвристических методов или методов искусственного интеллекта, например генетических алгоритмов.

Задачи планирования 2.2.1 и 2.2.2, а также задача управления 4 являются задачами линейного программирования и могут быть решены известными методами для данного класса задач, например, модифицированным симплекс-методом, аффинно-масштабирующими алгоритмами или методами на основе декомпозиции [30, 34].

Задача управления 1 является задачей линейного булевого программирования и для ее решения могут использоваться значительно более эффективные методы по сравнению с задачами управления 2 и 3. Здесь можно применять разнообразные приемы сокращения перебора на основе схемы Балаша и комбинированные алгоритмы, построенные на применении направленного перебора для решения соответствующей задачи линейного программирования.

Задача диспетчеризации. Задача диспетчеризации состоит в распределении клиентских запросов между виртуальными серверами пропорционально значениям переменных x_p, которые были определены на этапах планирования и управления. Эта задача решается с помощью системы правил диспетчером запросов СУИ. Данная система правил формируется модулем управления.

Итак, предложен подход к планированию и управлению ресурсами физического сервера для случая серверной виртуализации, когда на одном физическом сервере размещено несколько виртуальных машин. Разработан комплекс моделей планирования и управления ресурсами сервера с учетом различий в возможностях управления ресурсами и нагрузкой, которые зависят от особенностей реализации конкретных технологий виртуализации и приложений. Предложены алгоритмы решения задач планирования и управления распределением ресурсов с учетом важности приложений. Предложенные методы управления можно использовать при управлении ресурсами корпоративных ЦОД, когда СУИ известна значимость бизнес-процессов и соответственно значимость поддерживающих их приложений. Разработанные модели и методы были использованы при создании СУИ.

8.5. МОДЕЛИ УПРАВЛЕНИЯ ВИРТУАЛЬНЫМИ МАШИНАМИ ПРИ СЕРВЕРНОЙ ВИРТУАЛИЗАЦИИ И ЖИВОЙ МИГРАЦИИ ВИРТУАЛЬНЫХ МАШИН

8.5.1. Актуальность исследования

Проектирование ИТ-инфраструктуры осуществляется с учетом необходимости обеспечения качества предоставления и надежности функционирования ИТ-сервисов, как основы ведения бизнеса. В течение длительного времени решением этого вопроса было размещение наиболее критических приложений на выделенных серверах или кластерах [159]. Это существенно повышало надежность функционирования ИУС, но значительно увеличивало совокупную стоимость владения серверной составляющей ИТ-инфраструктуры за счет стоимости дополнительных серверов и лицензий на коммерческое ПО, расходов на электропитание, кондиционирование, администрирование, обслуживание и т. п.

По данным компании VMware, обычно загрузка выделенных под отдельные задачи серверов не превышает 15–20% от их полной мощности. Поэтому стало популярным сокращение количества физических серверов для поддержки различных сервисов путем применения технологии виртуализации серверов. Такой подход за счет большей загрузки серверов позволяет не только уменьшить их количество, но и сократить расходы на их содержание [288]. К тому же наличие функции миграции ВМ между физическими серверами без их перезагрузки позволяет получить показатели готовности сервисов, близкие к кластерным решениям, при значительно меньших затратах.

При этом актуальной становится задача управления распределением ресурсов между ВМ и расположения ВМ по серверам таким образом, чтобы при ограниченных вычислительных ресурсах, в первую очередь, обеспечить функционирование приложений, связанных с наиболее важными бизнес-процессами, даже за счет менее важных [148], а при избытке ресурсов расположить ВМ таким образом, чтобы высвободить серверы с целью экономии энергопотребления. Реализация подобных систем управления распределением ресурсов требует разработки соответствующих моделей и методов или адаптации уже известных [144] с учетом виртуализации серверов и миграции ВМ.

В работе [148] предложен ресурсный подход к управлению ресурсами ИТ-инфраструктуры. Предлагается рассматривать деятельность предприятия как совокупность бизнес-процессов, потребляющих определенное количество ресурсов определенных типов. Рассмотрены дискретный случай, когда отдельные бизнес-процессы поддерживаются в полном объеме или не поддерживаются вовсе, и непрерывный случай, когда возможна частичная поддержка бизнес-процессов.

Такой подход развит в работе [144], где кроме ресурсных ограничений добавлялись ограничения по надежности поддержки бизнес-процессов.

В работе [156] предложены модели и методы управления ИТ-инфраструктурой на основе ресурсного подхода на всех этапах жизненного цикла – от планирования при создании до эксплуатации и дальнейшего развития.

В работе [146] рассмотрены вопросы управления ресурсами, приложениями и нагрузкой для случая виртуализации серверов, но без возможности миграции ВМ.

В настоящее время возможность миграции ВМ без перерывов в предоставлении услуг повсеместно используется в ЦОД. Поэтому разработка моделей и ме-

тодов управления распределением ресурсов ИТ-инфраструктуры с использованием живой миграции, которая обеспечивает повышение доступности важнейших сервисов в условиях недостатка вычислительных ресурсов и экономию энергопотребления в условиях их избытка, является актуальной.

8.5.2. Модели и методы управления ресурсами при живой миграции

Пусть имеется несколько физических серверов S_i, $i=\overline{1,n}$, на которых под управлением гипервизора функционируют ВМ V_j, $j=\overline{1,m}$ (рис. 8.9).

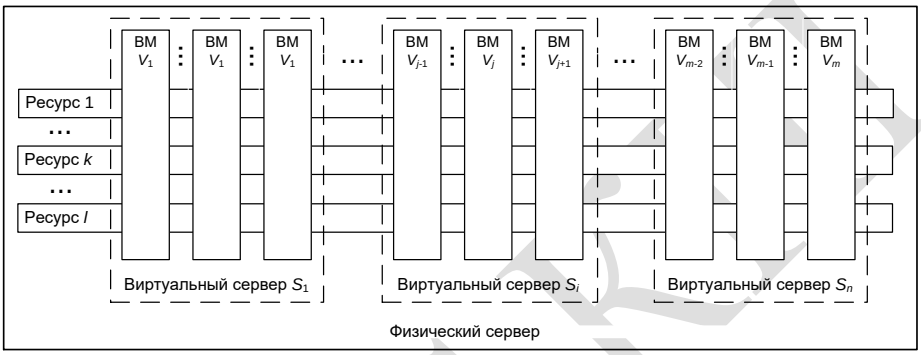


Рис. 8.9. Распределение ресурсов серверов между ВМ

Каждая из ВМ в зависимости от потока клиентских запросов использует определенное количество ресурсов типа R_k, $k=\overline{1,l}$ (память, процессорное время, дисковое пространство, пропускная способность подсистемы ввода-вывода внешних каналов связи и т. п.).

Введем необходимые для построения моделей обозначения: r_{ki} – количество ресурсов типа R_k, установленное на сервере S_i; p_{kj} – потребности ВМ V_j в ресурсах типа R_k для обеспечения заданного уровня SLA; x_{ij} – булева переменная, которая определяет, установлена ли ВМ V_j на сервере S_i.

Поскольку каждая ВМ одновременно находится не более чем на одном сервере, должно выполняться условие:

$$\sum_{i=1}^{n} x_{ij} \le 1, \quad j=\overline{1,m}. \tag{8.50}$$

К тому же, для нормального функционирования ВМ должны быть обеспечены достаточным объемом ресурсов серверов, на которых они расположены:

$$\sum_{j=1}^{m} x_{ij} p_{kj} \le r_{ki}, \quad k=\overline{1,l}, \; i=\overline{1,n}. \tag{8.51}$$

Рассмотрим две задачи расположения ВМ для случаев недостатка и избытка ресурсов.

Задача 1. В случае, если в результате увеличения клиентских запросов потребность отдельных ВМ в ресурсах определенного типа увеличилась настолько,

что становится невозможным обеспечить все ВМ необходимым количеством ресурсов, естественным выходом становится задача поддержки наиболее важных бизнес-процессов путем обеспечения ресурсами тех ВМ, на которых работают связанные с ними сервисы, за счет менее важных.

Обозначим через w_j, $j=\overline{1,m}$ важность приложений, установленных на ВМ V_j.

Тогда задачу можно сформулировать следующим образом: найти наилучшее распределение VM по серверам согласно критерию

$$\sum_{j=1}^{m}\sum_{i=1}^{n} x_{ij} w_j \to \max \tag{8.52}$$

при ограничениях (8.50), (8.51).

Эта задача представляет собой линейную задачу булевого программирования. Для ее решения воспользуемся методом ветвей и границ. Но сначала покажем, как можно, учитывая специфику задачи, снизить начальное количество возможных комбинаций, которое составляет 2^{mn}.

Во-первых, согласно ограничениям (8.50), каждая ВМ может быть расположена не более чем на одном сервере, т. е. вместо перебора 2^n комбинаций для каждой ВМ V_j, можно ограничиться рассмотрением только $n + 1$ вариантов: $y_j = 0,...,n$, где y_j – номер сервера, на котором установлена ВМ V_j ($y_j = 0$ в случае, если в результате нехватки ресурсов ВМ V_j не установлена на одном сервере или ресурсы для нее выделяются по остаточному принципу).

Во-вторых, учитывая, что требования ВМ в ресурсах p_{ij} являются неотъемлемыми, если на каком-то из этапов построения дерева вариантов размещения ВМ по серверам одно из ограничений (8.51) перестает выполняться, продолжать построение этой ветви нет смысла, поскольку для всех ее узлов это ограничение также не будет выполняться.

В-третьих, следует учесть, очень часто при построении серверных пулов используют серверы с одинаковой конфигурацией. Если, например, есть два идентичных сервера, то количество возможных вариантов размещения ВМ можно сократить в 2 раза, поскольку перенесение всех ВМ с первого сервера на другой, а со второго на первый никак не повлияет на объем ресурсов, которые используются. Иными словами, перестановка строк матрицы $X = ||x_{ij}||$, соответствующих одинаковым столбцам в матрице $R = ||r_{ki}||$, никак не влияет ни на выполнение ограничений (8.50), (8.51), ни на значение критерия (8.52). Иными словами, для серверов с одинаковой конфигурацией важно не абсолютное размещение ВМ на этих серверах, а их расположение друг относительно друга. При увеличении количества серверов с одинаковой конфигурацией количество комбинаций, которые можно не рассматривать, значительно возрастает от $(n-1)$ для случая размещения всех ВМ на одном сервере до $(n!-1)$ при размещении ВМ на разных серверах.

Например, для случая трех ВМ и трех идентичных серверов из возможных 27 комбинаций расположения ВМ лишь 5 являются уникальными, а остальные 22 комбинаций альтернативными, т. е. даже для таких незначительных значений m и n начальное множество вариантов можно сократить более чем в 5 раз. Алгоритм перебора и его результаты для этого случая проиллюстрированы в табл. 8.1. Для наглядности буквами «A», «B», «C» обозначены ВМ, а символы S_1, S_2, S_3, указанные в скобках, означают серверы, на которых размещены ВМ.

Таблица 8.1. Рабочие и альтернативные комбинации размещения ВМ на серверах

Рабочие комбинации размещения ВМ	Альтернативные комбинации размещения ВМ
A(S_1)-B(S_1)-C(S_1)	A(S_2)-B(S_2)-C(S_2); A(S_3)-B(S_3)-C(S_3)
A(S_1)-B(S_1)-C(S_2)	A(S_2)-B(S_2)-C(S_1); A(S_3)-B(S_3)-C(S_1); A(S_1)-B(S_1)-C(S_3); A(S_2)-B(S_2)-C(S_3); A(S_3)-B(S_3)-C(S_3)
A(S_1)-B(S_2)-C(S_1)	A(S_2)-B(S_1)-C(S_1); A(S_3)-B(S_1)-C(S_3); A(S_1)-B(S_3)-C(S_1); A(S_2)-B(S_3)-C(S_2); A(S_3)-B(S_2)-C(S_3)
A(S_1)-B(S_2)-C(S_2)	A(S_2)-B(S_1)-C(S_1); A(S_3)-B(S_1)-C(S_1); A(S_1)-B(S_3)-C(S_1); A(S_2)-B(S_3)-C(S_2); A(S_3)-B(S_2)-C(S_2)
A(S_1)-B(S_2)-C(S_3)	A(S_1)-B(S_3)-C(S_2); A(S_2)-B(S_3)-C(S_1); A(S_3)-B(S_2)-C(S_1); A(S_2)-B(S_1)-C(S_3); A(S_3)-B(S_1)-C(S_2)

Рядом с рабочими комбинациями в табл. 8.1. перечислены комбинации, которые являются альтернативными, т. е. приводят к таким же результатам, что и рабочие, и могут быть исключены из перебора. Алгоритм перебора при построении дерева расположения ВМ заключается в следующем: на очередном уровне очередной ветки количество серверов, рассматриваемых для перебора, принимается на 1 больше, чем максимальное для предыдущих уровней этой же ветви, но не более чем n.

Рассматривается только случай расположения ВМ «А» на сервере S_1, а на серверах S_2 и S_3 не рассматривается, поскольку серверы S_1–S_3 являются полностью идентичными и расположение ВМ «А» на них приведет к таким же результатам. При решении задачи в реальных условиях, с целью уменьшения количества миграций ВМ, при выборе произвольного сервера выбирается тот, на котором эта ВМ уже установлена. При размещении ВМ «В» рассматриваются два варианта: B(S_1) – размещение ВМ «В» на том же сервере, что и ВМ «А», и B(S_2) – размещение ВМ «В» на сервере S_1. Вариант B(S_3) не рассматривается, поскольку он принципиально ничем не отличается от варианта B(S_2) и приведет к таким же результатам (аналогично, если ВМ «В» уже установлена на сервере S_3, следует использовать именно его, а не сервер S_2).

Если имеются серверы с отличающейся конфигурацией, к схеме перебора добавляются соответствующие узлы на каждом уровне. Сюда также относятся случаи, когда вследствие нехватки ресурсов разрешается низкоприоритетные ВМ не размещать на одном сервере, выделять ресурсы для них по остаточному принципу. Формально это описывается путем размещения таких ВМ на несуществующем сервере S_0 (это возможно в задаче 1 и невозможно в задаче 2). В этом случае на первом уровне будет уже несколько узлов (рис. 8.10).

Кроме определения процедуры ветвления метод ветвей и границ предусматривает определение процедур оценки верхней и нижней границ с помощью приближенных, но быстрых алгоритмов.

При решении задачи 1 (задача максимизации) для определения оценки сверху можно использовать оптимистичный прогноз, который заключается в предположении того, что на очередном шаге алгоритма все ВМ, не распределенные между серверами, могут быть распределены таким образом, что ни одна из них не получает отказ в обслуживании (не будут размещены на фиктивном сервере S_0) при условии выполнения ресурсных ограничений (8.51).

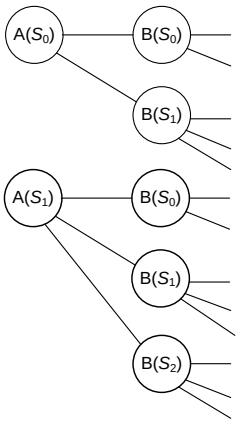


Рис. 8.10. Схема перебора комбинаций для неидентичных серверов

Чтобы получить оценку снизу, предлагается использовать «жадный» алгоритм, упорядочив ВМ по убыванию требований к тому ресурсу, недостаток которого стал причиной перехода от старого к новому плану размещения ВМ.

Процедура отсечения ветвей заключается в том, что ветвь, для которой оценка верхней границы меньше, чем оценка нижней границы хотя бы одной из других ветвей, рассматривается как бесперспективная и исключается из дерева решений. Учитывая, что требования ВМ к ресурсам являются неотъемлемыми, эта процедура может быть дополнена отсечением тех ветвей, для которых прекращают выполняться ограничения (8.51).

В случае, если на очередном шаге будет найдено решение, при котором все ВМ успешно размещены, т. е. ни одна ВМ не расположена на сервере S_0, поиск можно прекратить, поскольку глобальный оптимум найден и критерий (8.52) принимает максимальное значение. Но при наличии времени продолжение поиска может помочь найти решение, которое обеспечит меньшее количество миграций ВМ при реализации нового плана расположения, чем уже найденное решение и его альтернативные решения.

Задача 2. В случае, если количество имеющихся вычислительных ресурсов существенно превышает потребности ВМ, естественной становится задача распределить ВМ между серверами таким образом, чтобы уменьшить энергопотребление за счет отключения питания серверов, которые не используются в настоящее время.

Поскольку отключать серверы для экономии электроэнергии имеет смысл только в том случае, когда в наличии есть свободные ресурсы, модифицируем ограничения (8.50), чтобы гарантировать, что каждая ВМ будет обеспечена ресурсами одного и только одного сервера:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = \overline{1,m}. \tag{8.53}$$

Обозначим через e_i энергопотребление сервера S_i, когда на нем не работает ни одна ВМ.

Признак того, что на сервере S_i не работает ни одна ВМ, выразим следующим образом:

$$d_i = \prod_{j=1}^{m} \overline{x_{ij}}, \quad j = \overline{1,m}. \tag{8.54}$$

Тогда задачу минимизации энергопотребления можно сформулировать как задачу нахождения наилучшего набора серверов согласно критерию

$$\sum_{i=1}^{n} d\ e_i \to \max \tag{8.55}$$

при ограничениях (8.51), (8.53), (8.54).

Для решения задачи 2 предлагается тот же алгоритм, но с упрощенной процедурой перебора, поскольку согласно ограничению (8.53) невозможны случаи, когда какая-то ВМ не размещена на одном из серверов.

Реализация метода. Разработанные модели были апробированы для управления ВМ, работающих под управлением гипервизора VMware ESX на двух серверах.

Схема макета приведена на рис. 8.11.

Каждый из серверов имеет два сетевых интерфейса: один – для взаимодействия через сеть, второй – для передачи состояния памяти ВМ в процессе миграции ВМ. Миграция осуществлялась с помощью утилиты VMware vMotion, которая является составной пакета VMware vSphere.

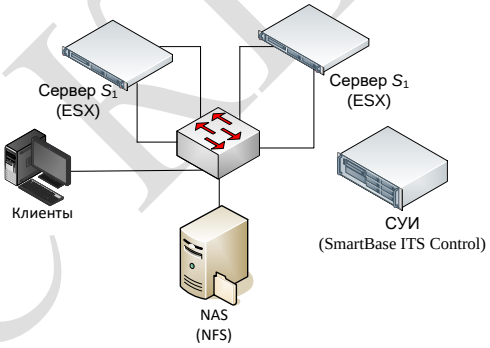


Рис. 8.11. Схема макета

Управляющая программа, которая является частью СУИ, написана на языке C#. Для взаимодействия с ПО VMware (мониторинг потребления ресурсов, управления распределением ресурсов и миграцией ВМ) использовалась библиотека vSphere SDK for. NET. На всех ВМ установлена ОС Linux. На ВМ «А» установлен веб-сервер Apache, для которого на языке PHP создано ПО, которое генерировало динамический контент в ответ на запросы, поступавшие от клиентов. На ВМ «В» и «С» запущены вычислительные процессы для создания постоянной фоновой загрузки ВМ.

В результате проведенных экспериментов обнаружено, что при заданных условиях из ресурсов (процессор, память, пропускная способность сети, дисковая подсистема) узким местом оказался процессор. Поэтому на графиках изображены только загрузки процессоров, мониторинг которых осуществлялся средствами VMware, а статистика обработки клиентских запросов собиралась и обрабатывалась с помощью написанного ПО.

Поток клиентских запросов постепенно увеличивался. На графиках (рис. 8.12–8.14) видно, что после того, как один из параметров (в этом эксперименте – загрузка процессора на сервере 1), в течение заданного времени превышал заданный порог, СУИ приняла решение о необходимости переноса ВМ «А» на сервер 2, что позволило ликвидировать узкое место и восстановить динамику обработки запросов. Пороговые значения и максимальная продолжительность их превышения задаются в конфигурации СУИ отдельно для каждого из параметров, исходя

из требований, оговоренных в SLA. В этом эксперименте решение о необходимости миграции ВМ принималось в случае, если средняя загрузка процессора превышала заданный уровень в течение трех минут.

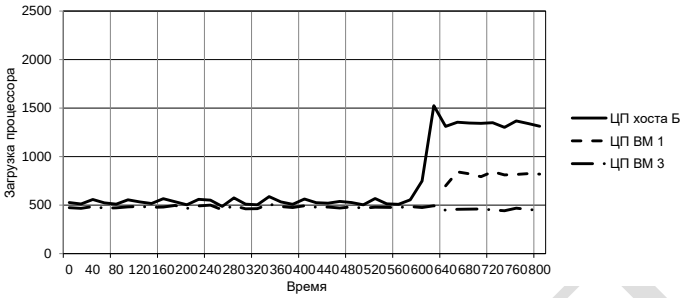


Рис. 8.12. Загрузка ЦП сервера 1

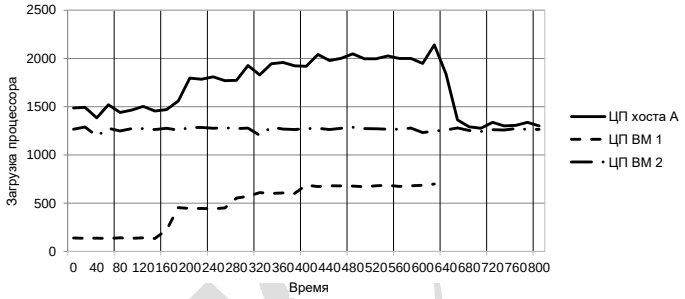


Рис. 8.13. Загрузка ЦП сервера 2

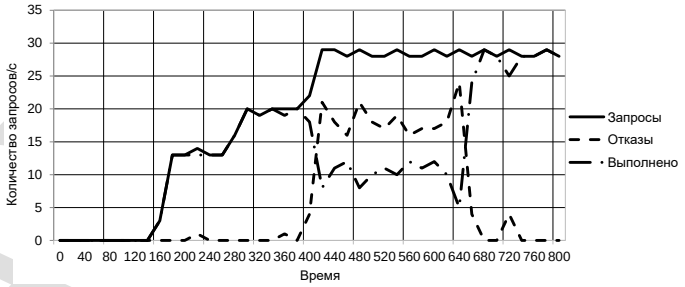


Рис. 8.14. Поступление и обработка запросов

Таким образом, рассмотрены вопросы управления размещением ВМ при серверной виртуализации с возможностью осуществления миграции ВМ без остановки процессов. Предложены модели управления миграцией ВМ и распределением ресурсов между ними для случаев недостатка и избытка вычислительных ресурсов. Эти модели могут быть применены для решения задач балансировки нагрузки между серверами путем миграции ВМ, для обеспечения высокого уровня доступности наиболее важных сервисов и уменьшения эксплуатационных расходов, а именно экономии электроэнергии путем отключения электропитания отдельных серверов с предварительным переносом ВМ с них на другие серверы. Для решения указанных задач предложена модификация алгоритма ветвей и границ.

8.6. МЕТОД УПРАВЛЕНИЯ РЕЗЕРВИРОВАНИЕМ РЕСУРСОВ ЦОД ДЛЯ ПОДДЕРЖАНИЯ ЗАДАННОГО КАЧЕСТВА ОБСЛУЖИВАНИЯ ЗАПРОСОВ ПОЛЬЗОВАТЕЛЕЙ

8.6.1. Необходимость управления резервированием ресурсов ЦОД

Развитие интернета и доступность высокоскоростных каналов связи позволили объединить рассредоточенные информационные и вычислительные системы предприятий, организаций и государственных учреждений в интегрированные ЦОД, услугами которых пользуются как телекоммуникационные компании, представители банковского и финансового секторов, так и корпорации, крупные организации со значительной территориально распределенной ИТ-инфраструктурой, а также общегосударственные и региональные органы власти.

Существует два пути создания ЦОД: обеспечение взаимодействия рассредоточенных средств хранения и обработки информации или создание специализированных централизованных ЦОД. Централизованные ЦОД имеют преимущества по сравнению ЦОД с рассредоточенной структурой, которые проявляются в использовании ограниченного количества ИТ, тогда как рассредоточенные ЦОД зачастую создают, интегрируя, вычислительные ресурсы и ИТ территориально распределенных автоматизированных систем. Эти системы созданы различными производителями, в разное время, на различных аппаратно-программных платформах и не допускают простого взаимодействия между собой из-за различия ОС и закрытости специального ПО.

Существуют корпоративные и коммерческие специализированные ЦОД. Несмотря на то, что, в первую очередь, особый интерес к применению коммерческих ЦОД проявляет малый и средний бизнес, для крупных корпораций с собственными ЦОД коммерческие центры могут применяться в качестве резерва мощностей. Услугами коммерческих централизованных ЦОД пользуются компании, занимающиеся электронной коммерцией, предоставляющие информационные услуги, финансовые организации, операторы связи и телекоммуникаций и др., а также физические лица. Качество услуг, предоставляемых пользователям со стороны ЦОД, обычно оговаривается в SLA, содержащем описание услуг, тарифы на услуги, уровень сервиса, спецификацию показателей качества сервиса, параметры уровня обслуживания и т. д. Готовность, доступность, непрерывность и целостность вычислительных ресурсов ЦОД или предоставляемых сервисов – одни из важнейших показателей надежности. Чтобы обеспечить поддержание параметров надежности системы на заданном в SLA уровне, а также для обслуживания запросов пользователей с параметрами времени, не превышающих оговоренных SLA значений, при существенной динамике поступления запросов, необходимо иметь избыток вычислительных ресурсов. В тоже время, современные ЦОД характеризуются высокой стоимостью не только создания, но и обслуживания, а следовательно, значительное количество незадействованных или излишне зарезервированных ресурсов является неоправданным с экономической точки зрения. В связи с этим задача управления зарезервированными ресурсами, необходимыми для предоставления услуг с показателями качества и надежности согласно SLA, является актуальной.

Одной из основных задач управления современными ЦОД можно считать повышение эффективности их работы. Это объясняется не только высоким уровнем капиталовложений в проектирование, строительство и оснащение

ЦОД, но и значительными текущими затратами на поддержание их работоспособности, которые включают не только заработную плату высококвалифицированных администраторов, но и существенные расходы, связанные с поддержкой соответствующей инфраструктуры обеспечения. Например, существенной статьей расходов на поддержку ЦОД в рабочем состоянии является оплата счетов за электроэнергию.

Одним из направлений повышения эффективности работы ЦОД считается применение технологии виртуальных серверов [146], которая одновременно с уменьшением количества физических серверов и повышением эффективности использования вычислительных ресурсов позволяет сократить расходы на администрирование, а также оптимизировать энергопотребление благодаря временному отключению незадействованных физических серверов и соответствующему снижению затрат на системы энергоснабжения и кондиционирования [159].

Однако применение виртуализации порождает проблемы, связанные как с необходимостью управления распределением ресурсов ЦОД [159], так и с решением задачи оптимизации количества зарезервированных ресурсов, необходимых для обеспечения требований к уровню обслуживания запросов пользователей, заданных в SLA. Последняя задача тесно связана с задачей планирования расширения ЦОД, поскольку в последнее время преобладает тенденция их поэтапного развития, предусматривающая постепенное наращивание суммарной мощности вычислительных систем и объема систем хранения данных в соответствии с требованиями заказчиков. Хотя при таком подходе необходимо учитывать тренды увеличения количества пользователей, он позволяет сократить как первоочередные инвестиции, так и стоимость эксплуатации.

Задача управления распределением ограниченных ресурсов ИУС рассматривается в работах [114, 148, 149, 156, 157], а управлению распределением ресурсов ЦОД посвящены работы [146, 158, 159, 228, 363, 364, 365, 366, 368, 369]. Однако указанные работы не уделяют внимания вопросам определения необходимого объема ресурсов горячего резерва для обеспечения необходимого качества обслуживания запросов пользователей.

В работе [144] рассмотрены вопросы управления распределением вычислительных ресурсов с учетом надежности аппаратно-программных средств, но не раскрываются вопросы качества обслуживания запросов пользователей и необходимого резервирования ресурсов.

Для управления ресурсами ЦОД используют СУИ [120, 154, 155], которые предназначены не только для обеспечения управления распределением вычислительных и коммуникационных ресурсов [149, 157], но и для решения задачи оптимизации объема зарезервированных ресурсов с целью поддержания показателей качества и надежности на указанном в SLA уровне, динамически изменяя объем зарезервированных для отдельного заказчика ресурсов в зависимости от текущего количества пользователей. Для решения этой задачи СУИ должна иметь математические модели и алгоритмы определения необходимого количества зарезервированных ресурсов, а также методы и средства управления их выделением и сокращением.

Целью параграфа является разработка метода управления резервированием вычислительных ресурсов ЦОД, необходимых для поддержания заданного качества обслуживания запросов пользователей с учетом динамики нахождения этих запросов.

8.6.2. Постановка задачи управления резервированием ресурсов ЦОД

Увеличение объемов обрабатываемой информации и существенное повышение скорости доступа к интернету, который повсеместно используется для организации частных виртуальных сетей, привели к значительному увеличению числа корпоративных и коммерческих ЦОД. Современные ЦОД позволяют не только консолидировано обрабатывать всю корпоративную информацию, но и обеспечивают работу социальных сетей, веб-конференций, интернет-магазинов и других современных объектов, предоставляя виртуальный хостинг, инфраструктуру, другие сервисы. Технологии, применяемые в ЦОД, достаточно эффективны для решения аналитических задач с целью поддержания бизнес-процессов, ERP и биллинговых систем, систем мобильных платежей и иных подобных систем.

Обычно заказчика не интересует, на каких типах серверов решаются его задачи, какие объемы систем хранения данных, пропускная способность каналов связи и другие параметры. Заказчику важно, чтобы пользователи его услуг получали результаты в виде сведений или отчетов в течение интервала времени, не превышающего значения, зафиксированные в SLA. Несоблюдение значений показателей SLA приводит к значительным штрафным санкциям. Обеспечение качества услуг ЦОД на должном уровне требует поддержания вычислительной системы, системы хранения данных и других систем в рабочем состоянии с заданным уровнем надежности. Для этого необходимо обеспечить постоянное наличие свободных ресурсов, которые оперативно задействуются для наращивания производительности сервиса при увеличении количества одновременных запросов. С одной стороны, качество, своевременность и объем предоставляемых услуг, с другой – необходимость минимизации затрат обуславливают актуальность задачи рационального управления ресурсами ЦОД, когда количество зарезервированных ресурсов динамически изменяется в зависимости от текущей нагрузки на сервис.

Для поддержания согласованного уровня услуг в условиях избытка ресурсов при существенной динамике поступления запросов необходимо оперативно задействовать резерв ресурсов. При этом становится актуальной задача рационального управления объемом зарезервированных ресурсов [125].

8.6.3. Суть предлагаемого метода управления объемами зарезервированных ресурсов

В SLA провайдер и заказчики фиксируют требования к качеству услуг. Для предоставления этих услуг в ЦОД выделяются вычислительные и иные ресурсы. Если выделенных ресурсов недостаточно для решения всех задач пользователей, запросы пользователей не выполняются с заданными ограничениями во времени. Происходит функциональный отказ, который приводит к существенному снижению качества услуг или недоступности сервиса. Устойчивость ресурсов к таким отказам будем описывать расчетным показателем надежности – коэффициентом ресурсной готовности $K_{\text{рг}}$, который определяет вероятность того, что в произвольный момент времени отдельному заказчику выделены достаточные объемы ресурсов для выполнения запросов пользователей, которые исключают появление функциональных отказов конкретного сервиса. ИТ управления ИТ-инфраструктурой должна обеспечивать устойчивость услуг к функциональным отка-

зам, выделяя и резервируя ресурсы для сервисов с последующим перераспределением незадействованных ресурсов при изменении количества запросов пользователей таким образом, чтобы поддерживать коэффициент ресурсной готовности на заданном уровне.

Поток запросов пользователей носит статистический характер, поэтому для выполнения требований SLA необходимо управлять динамическим выделением и перераспределением ресурсов. В этом случае СУИ, используя информацию о заказанных услугах, качестве услуг и предельных показателях качества их предоставления, например, в виде максимального времени обслуживания отдельного запроса, отслеживает динамику поступления запросов дифференцированно по каждой услуге. Для усиления вычислительных мощностей, поддерживающих услуги, коэффициент готовности $K_{\text{рг}}$ которых приближается к предельным значениям, СУИ выделяет дополнительные ресурсы. Для этого часть общих для всех услуг ресурсов, являющихся резервом ЦОД и находящихся в выключенном состоянии или в состоянии ожидания, включается по команде СУИ, проходит подготовку к работе и дополняет задействованные в настоящее время ресурсы. Выделение дополнительных ресурсов может происходить также путем перераспределения задействованных ресурсов с учетом приоритета услуг с применением моделей, предложенных в [114, 144, 146, 148, 149, 156, 157, 158, 159].

В отличие от традиционного подхода, когда сначала определяется пиковая нагрузка, а затем резервируются ресурсы в объеме, достаточном для поддержания уровня услуг при пиковой нагрузке, в предлагаемом методе объем зарезервированных ресурсов динамически изменяется в зависимости от интенсивности поступления запросов и увеличивается соответственно возрастанию интенсивности. При снижении интенсивности объем зарезервированных ресурсов сокращается, а освободившиеся ресурсы могут быть переданы для решения других задач или временно отключены в целях экономии электроэнергии и уменьшения затрат на обслуживание.

В качестве ресурсов ЦОД рассматривают процессорную емкость сервера, объем ОЗУ и дискового пространства сервера, пропускную способность каналов связи, емкость системы хранения данных, каналы запросов к СУБД и др. [158]. Часто принимают во внимание лишь часть процессорной емкости и объема памяти, которые выделяются отдельной услуге или отдельному заказчику. В качестве ресурса выступают и виртуальные машины или так называемые ноды – кванты ресурсов, в случае, если заказчикам выделяется некоторое количество ВМ или нодов, необходимых для поддержки предоставления услуг пользователям.

8.6.4. Разработка математической модели

Пусть кластер ЦОД имеет L физических серверов, образующих множество $N = \{N_1,...,N_L\}$. Каждый физический сервер N_l, $l = \overline{1,L}$ может быть представлен совокупностью ресурсов, характеризующие его мощность. На каждом физическом сервере N_l может базироваться G_l ВМ, причем каждая ВМ g_l, $l = \overline{1,G_l}$ может обслуживать максимум K_{gl} пользователей, обеспечивая гарантированное качество обслуживания в соответствии с требованиями SLA.

Пусть существует n видов ресурсов, критических относительно функциональных отказов.

Динамику использования *i*-го ресурса, $i = \overline{1,n}$ обусловленную выполнением некоторого переменного числа запросов сервером приложений, работающим на отдельной ВМ, представим как задействование и освобождение некоторого объема *i*-го ресурса.

Считаем, что *i*-й ресурс задействуется фиксированными порциями – квантами фиксированного объема q_i, где q_i, $i = \overline{1,n}$ – объем одного кванта *i*-го ресурса.

При увеличении количества запросов для поддержания уровня услуг задействуются дополнительные кванты ресурса. При этом объем Q_{ik} *i*-го ресурса после *k*-го шага увеличения объема задействованного ресурса определяется так:

$$Q_{ik} = \sum_{j=1}^{k} q_i \cdot r_{ij} \ , \tag{8.56}$$

где r_{ij} – количество квантов *i*-го ресурса, задействованных СУИ для удовлетворения запросов пользователей на *j*-м шаге увеличения объема задействованных ресурсов.

Такой подход позволяет абстрагироваться от количества запросов и оперировать числом задействованных ресурсных квантов. При этом не учитывается возможное наличие зависимостей между интенсивностями задействования и освобождения различных видов ресурсов, обусловленное природой запросов, особенностями выделения ресурсов приложениями и другими особенностями.

Интенсивность λ_i задействования *i*-го ресурса определяется как

$$\lambda_i = \lim_{\Delta t \to 0} \frac{M\left[X_i\left(t, t+\Delta t\right)\right]}{\Delta t}, \tag{8.57}$$

где $M\left[X_i\left(t, t+\Delta t\right)\right]$ – математическое ожидание количества задействованных квантов *i*-го ресурса на элементарном промежутке времени $\left(t, t+\Delta t\right)$.

Освобождение ресурсов происходит с интенсивностью $1/T_i$, где T_i – среднее время задействования ресурсного кванта.

Важная оценка задействованной *i*-го ресурса – его математическое ожидание m_i – может быть вычислено как за все время работы ВМ, так и на определенном актуальном временном интервале $T_{акт}$, который задается с учетом скорости изменения характеристик потока запросов, подключений пользователей, сезонных или суточных трендов использования ресурсов. Интервал $T_{акт}$ вводится для обеспечения необходимой динамики управления качеством предоставления услуг, а его изменение предусматривает перерасчет значения m_i с последующей коррекцией результатов расчета надежности.

Считая поток задействования-высвобождения квантов ресурсов пуассоновским, процесс задействования-высвобождения ресурса можно свести к частному случаю марковских цепей – процессу размножения-гибели с непрерывным временем [52].

Введем понятие дискретного *n*-мерного ресурсного пространства, на точках которого определяются вершины графа процесса размножения-гибели с непрерывным временем. При этом предполагается теоретически непрерывное поступление запросов и отсутствие ограничений на число состояний. Каждая рабочая

точка отражает возможное состояние текущего задействования ресурсов, а переходам между состояниями соответствуют ребра графа. Для упрощения расчетов, но без потери общности, воспользуемся только переходами, параллельными координатным осям. Пример графа процесса задействования-высвобождения отдельного ресурса представлен на рис. 8.15:

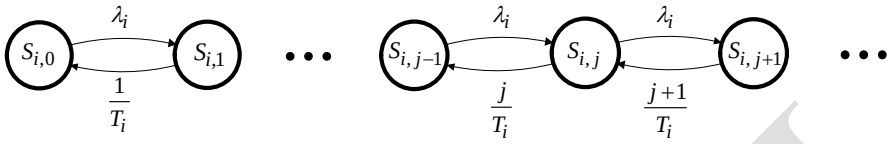


Рис. 8.15. Пример графа процесса задействования-высвобождения *i*-го ресурса

На рис. 8.15 $S_{i,j}$ – состояние *i*-го ресурса, характеризуемого использованием *j* квантов. Вероятность $p_{i,j}$ нахождения в состоянии $S_{i,j}$ вычисляется по формуле

$$p_{i,j} = \frac{[m_i]}{j!} \cdot e^{-m_i}. \tag{8.58}$$

Гарантированная вероятность обслуживания определяется вероятностью нахождения рабочей точки в пределах области доступных ресурсов, очерченной гиперповерхностью ресурсных ограничений в положительном полупространстве. Гиперповерхности выражают зависимости между доступными объемами ресурсов и в простейшем случае – при отсутствии последних являются ортогональными координатным осям гиперплоскости. Тогда оценка коэффициента ресурсной готовности $K_{\text{рг}}$ может быть проведена отдельно для каждого вида ресурсов по формуле [18]:

$$K_{\text{рг}} = P\{X_i \le v\} = \sum_{j=0}^{v} \frac{[m_i]^j}{j!} \cdot e^{-m_i} \approx \Phi\left(\frac{v - m_i}{\sqrt{m_i}}\right) + 0{,}5, \tag{8.59}$$

где $\Phi(x) = \frac{1}{\sqrt{2 \cdot \pi}} \int_0^x e^{-\frac{t^2}{2}} dt$ – функция Лапласа, v – количественное ограничение, выраженное в квантах ресурсов. Величина v определяет количество выделенных ресурсов, а коэффициент ресурсной готовности $K_{\text{рг}}$ соответствует вероятности того, что объем задействованных ресурсов не выходит за пределы выделенных ресурсов.

Если ограничения заданы более сложными гиперповерхностями, коэффициент ресурсной готовности определяется пространственной суммой вероятностей состояний задействования ресурсов, рассчитанных по формуле (8.58), которые принадлежат части положительного полупространства, ограниченного гиперповерхностью согласно зоне выделенных ресурсов. Актуальность оценки (8.59) сохраняется благодаря уменьшению объемов расчетов на одну ось, учитывая дискретность ресурсного пространства, которая позволяет четко определять принадлежность состояния (8.58) зоне выделенных ресурсов.

АУТСКИ

СУИ циклически оценивает значение m_i или пары λ_i и T_i, за определенный промежуток времени $T_{акт}$ по каждому из ресурсов с периодичностью $\Delta T_{i,акт}$. Периодичность можно принять одинаковой для разных видов ресурсов или определять для каждого ресурса отдельно на основе градиента $\mathrm{grad}\{m_i\}$.

На основе полученных значений m_i согласно выражению (8.58) или (8.59) рассчитываются вероятности $p_{i,j}$ и выполняется проверка принадлежности величин $p_{i,j}$ подпространству ресурсных ограничений. Сумма вероятностей $p_{i,j}$ соответствующих рабочих точек представляет собой текущее значение коэффициента ресурсной готовности $K_{\text{рг}}$.

Для управления резервированием ресурсов необходимо определить такое количество выделенных ресурсов, при котором коэффициент ресурсной готовности будет превышать заданное значение на допуск. Его можно определить эмпирическим путем и ввести с целью компенсации инертности вычислительной системы. Инертность СУИ также компенсируется выделением дополнительной доли ресурсов. Для ускорения процесса определения объема ресурсов можно воспользоваться таблицами зависимости $K_{\text{рг}}$ от количества выделенных ВМ $n_{\text{ВМ}}$ для соответствующего значения m_i, которые целесообразно предварительно подготовить и сохранять как результаты текущих расчетов.

Определенный таким образом объем ресурсов является минимально необходимым для поддержания качества обслуживания запросов пользователей ЦОД на заданном уровне. В случае, если выделенных ресурсов после очередного пересчета не достаточно, необходимо выделить ресурсы или перераспределить задачи таким образом, чтобы обеспечить соответствующий избыток свободных ресурсов. В случае превышения текущего объема выделенных ресурсов на минимально необходимую разницу избыток ресурсов можно задействовать для решения второстепенных задач, или отключить соответствующие серверы.

8.6.5. Результаты исследований

Рассмотрим два случая обеспечения гарантированной вероятности обслуживания запросов при доступе к ресурсам ЦОД. В первом случае часть ресурсов зарезервирована и находится в состоянии постоянной готовности («горячий» резерв), во втором – создается определенный резерв, который требует некоторого времени для подготовки к работе («холодный» резерв).

В качестве кванта ресурса рассматривается одна ВМ с фиксированными объемами ресурсов, для которой определяются интенсивности задействования и высвобождения, а также математическое ожидание $m_{\text{ВМ}}$ количества задействованных ВМ на промежутке времени $T_{акт}$.

Процесс функционирования системы с постоянной готовностью всех ресурсов описывается зависимостью коэффициента ресурсной готовности $K_{\text{рг}}$ от количества выделенных ВМ $n_{\text{ВМ}}$ при заданном математическом ожидании задействованных ВМ $m_{\text{ВМ}}$. Полученную на основе расчетов по формуле (8.59) эту зависимость показано на рис. 8.16.

Зависимость коэффициента ресурсной готовности $K_{\text{рг}}$ от математического ожидания задействования ВМ $m_{\text{ВМ}}$ при заданном количестве выделенных ВМ $n_{\text{ВМ}}$, рассчитанная по (8.59), приведена на рис. 8.17.

Поскольку одной из важнейших расходных статей обслуживание ЦОД считаются затраты на энергообеспечение, то организация резерва, когда часть зарезервированных вычислительных ресурсов находится в выключенном состоянии или в состоянии пониженного энергопотребления, имеет преимущества с экономической точки зрения. Сюда же можно отнести случай, когда с целью уменьшения простоя зарезервированных ресурсов их временно используют для решения второстепенных задач.

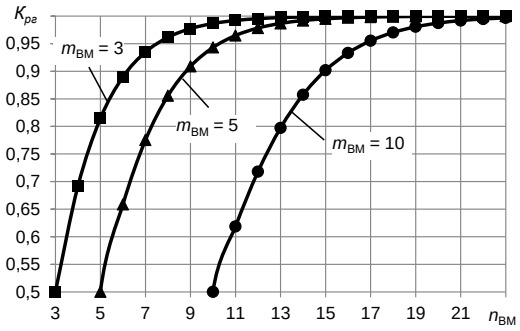


Рис. 8.16. Зависимость $K_{рг}$ от количества выделенных ВМ $n_{ВМ}$ при различных математических ожиданиях их задействования $m_{ВМ}$

При использовании «холодного» резерва показатели качества обслуживания запросов пользователей ухудшаются в связи со временем, необходимым для подготовки ресурса к работе (включение сервера, перенос/гибернация выполняемых на нем второстепенных задач и других операций). Степень ухудшения можно оценить, умножив коэффициент ресурсной готовности $K_{рг}$ на коэффициент P_{cs}, который определяется следующим образом:

$$P_{cs} = \prod_{j=\nu}^{\nu_{cs}-1} p_j \cdot \left(1 - e^{-\frac{j \cdot Ts_{j+1}}{T}} + e^{-\lambda \cdot Ts_{j+1}}\right), \qquad (8.60)$$

где p_j – вероятность пребывания в состоянии, которое характеризуется использованием j ВМ, рассчитывается по формуле (8.58); Ts_{j+1} – время, необходимое для подготовки $(j + 1)$-й ВМ к работе; ν – количество ВМ, постоянно находящихся в рабочем состоянии; ν_{cs} – количество ВМ с учетом незадействованных; λ и T вычисляются на основании (8.57).

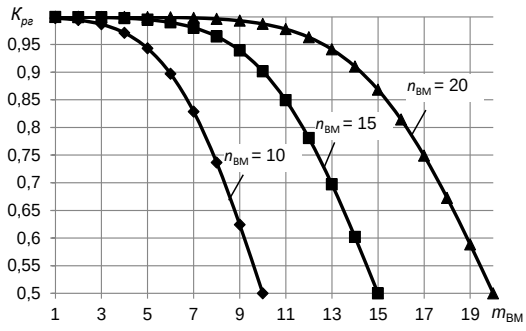


Рис. 8.17. Зависимость $K_{рг}$ от математического ожидания задействованных ВМ $m_{ВМ}$ при различных количествах выделенных ВМ $n_{ВМ}$

Таким образом, предложен метод управления резервированием ресурсов, обеспечивающий поддержание согласованного уровня услуг за счет динамического изменения объемов зарезервированных ресурсов, обусловленных стохастической природой запросов. Применение в СУИ предложенного метода управления резервированием ресурсов ЦОД для поддержания заданного в SLA качества обслуживания запросов пользователей позволит операторам услуг ЦОД не только поддерживать гарантированное качество услуг, но и сократить затраты на электроэнергию и обслуживание вычислительных ресурсов путем отключения части ресурсов при снижении интенсивности запросов.

ВЫВОДЫ К ГЛАВЕ 8

1. Выполнен анализ проблем управления ресурсами и нагрузкой ЦОД, в результате чего предложен подход к управлению ресурсами и нагрузкой ЦОД на основе комплекса математических моделей и методов решения оптимизационных задач. Систематизирована совокупность признаков, влияющих на постановки задач управления ресурсами и нагрузкой ЦОД, на основе чего выделены классы линейных и нелинейных задач с непрерывными, целыми и булевыми переменными, обеспечивающие широкий выбор критериев и ограничений для различных условий функционирования ЦОД.

2. С учетом моделей хостинга разработаны схемы доступа пользователей к ресурсам и архитектура подсистемы управления ресурсами и нагрузкой ЦОД. Управление нагрузкой и ресурсами ЦОД предложено рассматривать на трех уровнях – планирования, управления, диспетчирования. Это позволило структурировать комплексную задачу управления ресурсами и нагрузкой ЦОД и обеспечило гибкость и масштабируемость созданной системы управления.

3. Разработан комплекс моделей и методов управления распределением ресурсов и нагрузкой корпоративной ИТ-инфраструктуры в условиях избыточности ресурсов, характерной для ЦОД. Модели и методы ориентированы на их применение провайдером хостинговых услуг и отличаются возможностью учета текущей политики провайдера. Применение моделей и методов позволяет повысить эффективность использования дорогостоящих ресурсов ЦОД, перераспределяя их в соответствии с изменением политики провайдера.

4. Предложен метод управления резервированием вычислительных ресурсов ЦОД, необходимых для поддержания заданного качества обслуживания запросов пользователей с учетом динамики поступления запросов. В отличие от традиционного подхода, когда сначала определяется пиковая нагрузка, а затем резервируются ресурсы в объеме, достаточном для преодоления пиковой нагрузки, в предлагаемом методе объем зарезервированных ресурсов динамически изменяется в зависимости от интенсивности поступления запросов. Применение предложенного метода определения объема зарезервированных ресурсов для обеспечения выполнения запросов пользователей с гарантированным качеством обслуживания позволит операторам услуг ЦОД выбирать рациональную политику использования ресурсов.

5. Показано, что в условиях консолидации и виртуализации ресурсов ЦОД большое значение приобретает виртуальный хостинг с возможностью миграции машин. Разработан трехуровневый комплекс моделей и методов планирования, управления и диспетчировании ресурсов ЦОД для различных критериев и ограничений, учитывающих особенности политики провайдеров услуг и технологий виртуализации.

6. Экспериментальное исследование предложенных моделей, методов и средств показало, что их применение позволит повысить эффективность использования ресурсов ЦОД при одновременном снижении влияния рисков, связанных с нарушением провайдерами условий предоставления услуг, зафиксированных в SLA.

РЕПОЗИТОРИЙ

ГЛАВА 9

МОДЕЛИ И МЕТОДЫ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ПОТОКАМИ В КОРПОРАТИВНОЙ ИТ-ИНФРАСТРУКТУРЕ

Глава посвящена разработке методов управления информационными потоками в корпоративной информационно-телекоммуникационной системе.

9.1. АНАЛИЗ ПРОБЛЕМ УПРАВЛЕНИЯ СЕТЕВЫМ ТРАФИКОМ

9.1.1. Актуальность проблемы управления сетевым трафиком корпоративной ИТ-инфраструктуры

Повышение качества предоставления ИТ-сервисов можно осуществлять двумя основными путями. В первом случае ИТ-сервисам, качество которых необходимо повысить, СУИ предоставляет дополнительные вычислительные ресурсы ЦОД и управляет нагрузкой на эти ресурсы. Модели, методы и алгоритмы решения этого класса задач предложены в главах 7 и 8.

Второй путь повышения качества предоставления ИТ-сервисов основан на увеличении скорости прохождения транзакций пользователя. Это достигается выделением дополнительных коммуникационных ресурсов и сводится, в итоге, к выделению дополнительной гарантированной полосы пропускания на первом уровне ИТ-инфраструктуры – уровне сетевого взаимодействия.

Задачи, связанные с выделением ресурсов на уровне сетевого взаимодействия, относятся к классу задач управления сетевым трафиком или traffic engineering.

Повышение скорости прохождения транзакций пользователя через сетевую инфраструктуру может быть обеспечено:

- выделением дополнительной полосы пропускания наиболее критичным ИТ-сервисам за счет уменьшения полосы менее важным ИТ-сервисам;
- ограничением полосы пропускания, выделяемой распределенным приложениям;
- сокращением накладного трафика, иными словами, повышением продуктивности использования сетевой инфраструктуры.

В данном параграфе предложен подход к управлению сетевыми ресурсами с учетом значимости бизнес-процессов, генерирующих трафик. Применение такого подхода в СУИ позволяет:

- повысить эффективность использования сетевых ресурсов;
- оптимизировать распределение сетевых ресурсов между бизнес-процессами, распределенными приложениями и пользователями;
- сбалансировать соотношение между входящей нагрузкой и производительностью сети с целью предотвращения снижения эффективности утилизации сетевых ресурсов и возможности потери работоспособности в результате перегрузок.

Необходимость нового подхода к управлению сетевыми ресурсами обусловлена тем, что СУИ должна максимально быстро реагировать на изменение состояния и загруженности сети. Реализация такого подхода требует анализа факторов, влияющих на эти параметры сети. Скорость реакции СУИ на перегрузку в сети определяется периодом сбора данных о загруженности каналов связи, временем принятия решения в СУИ и временем реализации команд управления.

Кроме того, длительность цикла управления зависит от объема контрольной информации о состоянии элементов сети, сложности алгоритмов обработки информации в СУИ и производительности серверов СУИ и объема управляющей информации и времени передачи его по сети.

Величина цикла управления определяет способность СУИ адаптироваться к изменяющимся условиям в сети.

9.1.2. Анализ проблем сетевого управления

Сегодня аналитики с проектированием сетей чаще всего связывают два недочета. Во-первых, недостаточно внимания уделяется будущим проблемам их управления. Во-вторых, разработчики практически не обращают внимания на реальные и перспективные потребности бизнес-процессов, сосредотачиваясь в основном на пропускной способности и ограничениях по финансированию. Как правило, при проектировании корпоративных локальных сетей учитывается только характер бизнеса заказчика и количество подключаемых к сети компьютеров.

Такой подход приводит к тому, что развитие инфраструктуры вызывает сложность и неэффективность управления сетевыми и информационными ресурсами. Проектирование ТКС без учета требований со стороны бизнес-процессов, распределенных приложений и конкретных пользователей, в итоге, приводит к нехватке пропускной способности и, как следствие, к снижению эффективности выполнения бизнес-процессов. При этом основной и единственный выход из данной ситуации разработчики ТКС видят в увеличении пропускной способности каналов связи. Безусловно, существенное повышение скорости передачи данных способно решить эту проблему, но зачастую это требует значительных финансовых вложений.

Более перспективным представляется подход, при котором проектирование ТКС осуществляется с учетом индивидуальных потребностей бизнес-процессов, планов развития бизнеса, а выбираемые протоколы и оборудование должны обеспечивать приоритетную передачу и обслуживание сообщений бизнес-процессов с учетом их значимости. При этом должны быть определены количество категорий процессов или ИТ-сервисов и количество приоритетов, а также разработан механизм, сопоставляющий категорийность с приоритетами. Кроме того, необходимо, чтобы СУИ обеспечивала эффективное интегрированное управление приложениями и информационно-коммуникационными ресурсами.

Одной из наиболее актуальных проблем сетевого управления является управление ресурсами сети с целью оптимизации сетевого трафика. Острота проблемы определяется тем, что локальные перегрузки в сети приводят к функциональным отказам, которые в отличие от неисправности оборудования могут носить временный характер, но ухудшают характеристики сети, например, увеличивают задержку передачи сообщений, количество потерянных пакетов. Тем самым они

увеличивают время реакции сети на управляющие воздействия. Это, в итоге, приводят к ухудшению качества предоставления ИТ-услуг. Кроме того, когда нагрузка в сети превышает ее потенциальную производительность, может произойти потеря работоспособности сети. Причиной этому является чрезмерная входящая нагрузка, приводящая к локальным или общим перегрузкам сети, а также к увеличению транзитной нагрузки в узлах коммутации (УК).

Перегрузки могут быть вызваны увеличением количества пользователей, генерирующих трафик, новыми типами трафика, генерируемого пользователями, увеличением скорости входящих потоков, проблемами с адресацией, малым объемом буферов в УК, низкой производительностью УК, отказами в сети и другими причинами.

Это и превращает управление сетевыми ресурсами в важную задачу сетевого управления. При этом, в первую очередь, речь должна идти об оптимизации маршрутизации, обеспечивающей распределение и перераспределение потоков информации, и управлении потоками (трафиком), базирующемся на ограничении входящей и транзитной нагрузок в сети [33, 66, 100, 119, 172].

Решение этих задач осуществляется в предположении, что особенностью сетей передачи данных является низкая динамика характеристик трафика. Это позволяет определять дискретные интервалы времени (интервалы управления T_y), в течение которых параметры сетевого трафика изменяются столь незначительно, что состояние сети рассматривается как стационарное. В этом случае в системе сетевого управления с периодом T_y решаются простые задачи математического программирования, не содержащие время в явном виде [33].

Формулировка целевой функции и функции ограничений при формализации задач сетевого управления является достаточно сложной проблемой, поскольку зависит от используемых методов управления сетевыми ресурсами, алгоритмов маршрутизации в УК, применяемых сетевых технологий и других аспектов.

Решение задач управления сетевыми ресурсами сопряжено с рядом сложностей, обусловленных разнородностью сетевого трафика, носящего взрывной характер, большим количеством пользователей и множеством требований со стороны пользователей и распределенных приложений к качеству сетевых услуг, высокими скоростями доступа и информационного обмена, способными очень быстро создать перегрузки в сети, потерями информации в результате сетевых перегрузок, что может привести к экономическим санкциям, необходимостью быстрой реакции СУИ на изменяющиеся условия передачи данных. Это требует применения быстродействующих алгоритмов управления сетевыми ресурсами, поиска компромиссных решений при управлении доступом к ограниченным сетевым ресурсам. В последнем случае речь идет как о распределении ресурсов между множеством приложений и пользователей, так и о необходимости выполнения противоречивых требований к характеристикам информационного обмена. Например, требования обеспечения максимальной загруженности каналов связи, как общесистемное требование, противоречит требованию минимизации задержек при передаче сообщений со стороны множества критичных к величине задержки приложений.

При решении этих задач рассматриваются вопросы транспортировки данных, выполняемых транспортной сетью, под которой подразумевается совокупность аппаратно-программных средств сети, реализующих функции и протоколы транспортного и всех ниже расположенных уровней модели OSI/ISO.

Поэтому перспективные СУИ для управления сетевыми ресурсами должны обеспечивать:

- выделение сетевых ресурсов ИТ-сервисам, бизнес-процессам и операциям, распределенным приложениям и пользователям с учетом приоритезации;
- перераспределение ресурсов при изменении значимости ИТ-сервисов, бизнес-процессов или возникновении нештатных ситуаций;
- исключение перегрузок сети путем своевременного изменения маршрутов (оптимизация маршрутизации к перегрузкам и отказам) и управления входящими потоками информации;
- управление устранением неисправностями; оптимизация настройки сетевых элементов и др.

При этом способность СУИ кардинально решить проблему поддержания эффективности ИТ-сервисов связана, в первую очередь, с управлением трафиком данных.

В целом, существуют два основных подхода к управлению сетевой нагрузкой – использование методов маршрутизации и управление сетевыми потоками. Рассмотрим возможности каждого из этих подходов для повышения эффективности предоставления ИТ-услуг в ИУС.

9.1.3. Управление сетевой маршрутизацией

В крупномасштабных сетях обычно имеется множество путей передачи информации между каждым отправителем и получателем сообщений. Выбор определенных путей для каждых двух УК составляет сущность задачи маршрутизации.

Конкретизация пути в процессе функционирования сети осуществляется с помощью хранимой в каждом *i*-м, $i=\overline{1,N}$ УК или маршрутизаторе таблицы или матрицы маршрутов $M_i=\left\|m_{ijr}\right\|$, $i=\overline{1,N}$, $j=\overline{1,G}$, $r=\overline{1,N}$, где G – количество ветвей, исходящих из *i*-го узла, а элемент m_{ijr} определяет порядковый номер *j*-й ветви при передаче информации из *i*-го узла в *r*-й, либо вероятность выбора *j*-й ветви [40].

Решение задачи маршрутизации состоит в том, чтобы для всей сети определить план распределения потоков. Для этого необходимо задать множество $\{M_i\}$, $i=\overline{1,N}$ маршрутных таблиц [33].

Для каждой ветви в соответствии с выбранным критерием оптимизации задается некоторый вес выбранной метрики. Оптимальным по заданному критерию является маршрут с минимальным либо максимальным суммарным значением метрики. Использование различных критериев оптимальности приводит к получению в общем существенно различающихся планов маршрутизации трафика. Поскольку различные метрики могут противоречить друг другу, отсутствует принципиальная возможность решения универсальной задачи оптимизации планов маршрутов или распределения потоков. Следствием этого является широкое разнообразие часто несовместимых критериев оптимизации, что и объясняет невозможность создания алгоритмов универсальной маршрутизации.

Эффективность использования сетевых ресурсов и способность сети противостоять перегрузкам во многом зависит от используемых методов маршрутизации.

Статическая маршрутизация предполагает однократное определение путей передачи для каждого информационного потока, а переопределение маршрутов

проводится при изменении условий передачи информации, вызванных неисправностями или изменениями конфигурации сети. Статические маршруты часто устанавливаются администратором сети вместо механизмов автоматической маршрутизации и используются для обслуживания исключительных ситуаций. Статические маршруты являются потенциальным источником ошибок при изменении конфигурации или при возникновении неисправностей в сети.

Эффективное использование ресурсов сети может быть обеспечено при использовании алгоритмов динамической адаптивной маршрутизации. В этом случае маршрутные таблицы из $\{M_i\}$, $i=\overline{1,N}$ переопределяются при изменении условий передачи данных в сети, вызванных отказами в УК, перегрузками, неисправностями линий связи и другими факторами. Динамическое перераспределение потоков данных используется в подавляющем большинстве современных протоколов маршрутизации. Адаптация чаще всего привязана к нагрузке в линиях или отказам каналов и оборудования.

При адаптивной маршрутизации необходимо определить периодичность обновления маршрутных таблиц. В зависимости от характера периода обновления алгоритмы адаптивной маршрутизации делятся на синхронные, когда управляющая информация и таблицы обновляются через постоянные интервалы времени, и асинхронные. В последнем случае обмен управляющей информации для обновления таблиц проводится в моменты превышения контролируемых параметров пороговых значений.

Порядок выбора линий в УК, использующих только локальную информацию, может проводиться на основе сведений о кратчайшем пути передачи информации из *i*-го узла в *r*-й, стоимости передачи по линии, длины линии, задержке передачи пакетов, вероятности потери пакетов, длины очереди (загруженности линии), сведений об отказах каналов либо необходимости равномерной загрузки линий.

Если в сети используются алгоритмы распределенной адаптивной маршрутизации, то предпочтения при выборе путей определяются на основе информации о состоянии всей сети и рассылки управляющей информации по УК.

При решении задач маршрутизации, как правило, не учитываются требования со стороны бизнес-процессов к информационному обмену.

Отметим, что маршрутизация может лишь частично снизить потери из-за перегрузки в сети. Больший эффект можно получить, осуществляя управление нагрузкой на сеть путем управления потоками.

9.1.4. Управление потоками информации в сети

Использование динамической маршрутизации позволяет поддержать работоспособность сети только при незначительных локальных и общих перегрузках сети. При превышении нагрузки на сеть выше некоторого значения происходит столь существенное снижение скорости передачи, что такую ситуацию можно рассматривать как потерю работоспособности сети, и маршрутизация в принципе не может решить эту проблему.

Кардинальным выходом из такой ситуации без модификации сети может быть межузловое управление потоками, которое предполагает ограничение входящей в сеть нагрузки и транзитной нагрузки в УК. Эти аспекты управления обычно рассматриваются как управление потоками в сети [33].

В этом случае СУИ должна обеспечивать динамическую реконфигурацию установленных соединений, а также ограничивать входящую и транзитную нагрузку с учетом особенностей неоднородного трафика различного происхождения с разнообразными статистическими характеристиками и требованиями к показателям качества обслуживания.

В конвергентных сетях для борьбы с перегрузками могут быть использованы следующие механизмы управления потоками:

- регулирование возможности установления логических соединений (доступа в сеть) для пользователей, процессов и приложений, количества одновременно устанавливаемых соединений пользователем, процессом и приложением;
- блокирование работы пользователей, процессов и приложений;
- управление скоростью передачи или управление использованием полосы пропускания (сетевого интерфейса рабочей станции пользователя, запускаемых им приложений, процессов, распределенных приложений, сетевых и иных сервисов);
- механизм формирования трафика, обеспечивающий сглаживание неравномерного входного потока за счет буферизации, разделения по очередям, ограничения скорости передачи и использования других приемов;
- управление приоритетами и др.

В соответствии с моделью ISO/OSI [99] различают два механизма управления потоками – одноуровневый и интерфейсный.

При одноуровневом управлении потоками регулируется скорость передачи данных между логическими объектами, находящимися на одном уровне модели в различных открытых системах. Протоколы различных уровней сетевой архитектуры осуществляют управление потоками в виртуальном канале между абонентами сети, на участке доступа абонентов в сеть и между УК.

Интерфейсное управление потоками регулирует скорость передачи между логическими объектами, находящимися на смежных уровнях в одной открытой системе. К интерфейсному управлению можно отнести регулирование скорости поступления в сеть данных от абонентов и управление доступом к ресурсам ИУС. При существенных перегрузках сети может ограничиваться доступ низкоприоритетных бизнес-процессов к ИТ-ресурсам [113, 119, 126] или производиться временное отключение части абонентов от сетевых услуг.

Управление потоками между УК, как правило, осуществляется децентрализовано в каждом узле путем регулирования очередей пакетов на входе или выходе УК. При этом рассматриваются разнообразные классические модели работы с очередями в каждой исходящей линии связи [51] и различные схемы разделения и управления буферами УК [66].

Благодаря алгоритмам управления потоками можно эффективно поддерживать такое соотношение между входящей нагрузкой и пропускной способностью сети, при котором не возникают существенные и длительные перегрузки в сети.

Методы управления потоками входящей нагрузки [33] построены на ограничении входящей на УК нагрузки при переполнении входных буферов или буферов для транзитной нагрузки. При этом ограничивается либо вся нагрузка на УК, либо ограничение касается только низкоприоритетных пакетов входящих и транзитных потоков. Более сложные алгоритмы управления ограничивают входящую в сеть нагрузку при перегрузке в любой из ветви оптимального маршрута передачи пакетов от отправителя к получателю.

Эффективность алгоритмов управления потоками можно оценить по таким усредненным по всей сети параметрам: уменьшение задержки при передаче пакетов; уменьшение потерь при передаче пакетов; эффективность использования пропускной способности сети и др.

В последнем случае в рассмотрение вводится коэффициент использования сетевых ресурсов:

$$K_{c_i} = \frac{R'_{c_i}}{R_{c_i}}, \tag{9.1}$$

определяемый как отношение фактической загруженности сети R'_C к максимально достижимой R_c загруженности сети. При этом, R'_C соответствует суммарному количеству успешно переданных пакетов через сеть за определенный интервал времени, а R_c определяется как суммарная максимально возможная загруженность всех линий связи.

При определении R_c учитываются естественные ограничения, накладываемые структурой сети. Например, для фрагмента сети, приведенного на рис. 9.1, R'_C будет определяться следующим образом:

$$R_c = \min\{(S_1 + S_2 + S_3), S_4, S_5, (S_6 + S_7)\}, \tag{9.2}$$

где S_i, $i = \overline{1,7}$ – пропускная способность i-го канала связи.

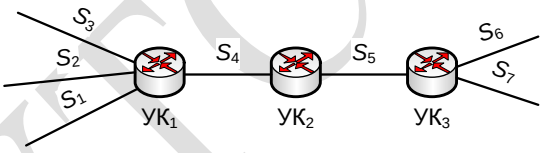


Рис. 9.1. Пример фрагмента сети для определения R'_C

Теоретически коэффициент K_c будет возрастать от 0 до 1 при увеличении загруженности сети, а при перегрузке сети будет уменьшаться от 1 до 0 за счет снижения количества успешно переданных пакетов, при этом увеличивается задержка передачи блоков данных. В идеальном случае, когда фактическая загруженность ресурсов сети соответствует максимально возможной загруженности всех линий связи, выполняется условие $K_c = 1$.

Аналогично можно ввести коэффициент K_{c_i} использования ресурса i-го канала связи:

$$K_{c_i} = \frac{R'_{c_i}}{R_{c_i}}, \tag{9.3}$$

где R'_{c_i} – загруженность i-го канала связи, R_{c_i} – максимально возможная загруженность канала с учетом ограничений типа (9.2).

Хотя маршрут между двумя узлами может динамически изменяться, характеристики пути передачи, например, максимальный размер нефрагментируемой

дейтаграммы или средняя задержка, измеренная транспортным протоколом, остаются практически неизменными в течение продолжительного времени для каждого транспортного соединения между произвольной парой хостов. Информация о свойствах путей хранится в маршрутизаторах. Эти данные могут использоваться СУИ при определении оптимальных путей передачи информации.

В настоящее время для управления входными потоками и для борьбы с перегрузками используются механизмы оконного типа и управление скоростями передачи по отдельным логическим соединениям [29, 33, 66, 100]. Окна являются средствами ограничения нагрузки на логическое соединение. Размер окна определяет количество пакетов отдельного логического соединения, которые могут быть переданы отправителем сообщений, до получения подтверждений об их приеме узлом-получателем. Нулевое значение ширины окна, установленное для отдельного логического соединения, указывает на то, что передача данных по этому соединению блокируется. Установка нулевых окон может применяться при разработке адаптивных алгоритмов управления потоками. Окна реализуются как на входе в сеть, так и внутри сети в УК. Недостатком метода являются существенные затраты на реализацию процедур подтверждения и переспросов.

Метод управления потоками, основанный на регулировании скорости передачи по логическому соединению, является более эффективными. Но и ему свойственны недостатки. В частности он не учитывает требования со стороны бизнес-процессов или ИТ-сервисов к информационному обмену.

Естественно, наибольший положительный эффект при управлении сетевыми ресурсами может быть достигнут при комбинированном использовании в СУИ алгоритмов маршрутизации и управления потоками.

В ИТ УИТИ выбор критерия оптимальности для решения задачи управления распределением сетевых ресурсов определяется на основе требований, предъявляемых со стороны бизнес-процессов или ИТ-сервисов к характеристикам информационного обмена. Так, перед СУИ может быть поставлена задача в течение определенного времени обеспечить наиболее важные бизнес-процессы сетевыми ресурсами в запрашиваемом ими объеме с максимальным качеством услуг. Такая задача может возникнуть, например, при организации важной для корпорации видеоконференции.

При выборе критериев оптимальности на основе требований со стороны бизнес-процессов наиболее важные распределенные приложения могут получить максимально комфортные условия для осуществления информационного обмена. Таким критерием может быть, например, минимизация задержки передачи данных.

Если со стороны бизнес-процессов требований к параметрам информационного обмена или характеристикам сетевых сервисов не предъявляется, то в качестве критериев оптимальности для решения задач динамической маршрутизации можно использовать общесетевые критерии, предполагающие максимально эффективное использование пропускной способности всей сети при выполнении требований к качеству информационного обмена. В этом случае все бизнес-процессы, использующие в своей работе сетевое взаимодействие, могут получить приблизительно равные условия для осуществления информационного обмена.

В СУИ при управлении сетевым взаимодействием бизнес-процессов целесообразно реализовать четыре механизма управления потоками:

Архитект

• предоставление доступа к сетевым ресурсам в запрашиваемом объеме с требуемым качеством обслуживания или с максимальным качеством, обеспечиваемым сетью в принципе;

• предоставление доступа к сетевым ресурсам в запрашиваемом объеме, но с пониженным качеством сетевых услуг по транспорту данных;

• ограничение доступа к сетевым ресурсам в виде ограничения запрашиваемой полосы пропускания, ограничения количества одновременно запрашиваемых сеансов связи и других параметров;

• отказ в доступе бизнес-процесса к сетевым ресурсам, отказ в обслуживании.

Для пользователей такие действия СУИ будут восприниматься следующим образом:

• доступ к удаленным ресурсам и работа распределенных приложений происходит в штатном режиме;

• работа приложений и доступ к ресурсам существенно замедляются;

• невозможно получить доступ к удаленным ресурсам, некоторые распределенные приложения, сетевые службы и сервисы не работоспособны.

При этом СУИ при управлении распределением сетевых ресурсов относительно потоков или логических соединений может предпринимать следующие действия:

• не ограничивать скорость передачи и не использовать механизмы управления потоками (равноправная работа приложений, пользователей и видов трафика);

• выделить и зафиксировать конкретную скорость;

• установить максимальную скорость передачи, которая может только уменьшаться;

• установить минимальную скорость, которая может только увеличиваться;

• установить диапазон возможных изменений скоростей от минимальной до максимальной;

• не выделять полосу пропускания, т. е. полностью запретить передачу отдельного потока, установить логические соединения отдельного приложения, т. е. заблокировать работу распределенного приложения (приложений) или только отдельных пользователей отдельных приложений.

9.1.5. Постановка проблемы управления сетевым трафиком в корпоративной ИТ-инфраструктуре

Повышение уровня ИТ-услуг может быть достигнуто выделением отдельным приложениям A_l, $l=\overline{1,m}$ дополнительных ресурсов. Эту задачу можно решить двумя путями. В первом случае приложениям A_l, $l=\overline{1,m}$, предоставляющим ИТ-услуги, уровень которых нужно повысить, выделяются дополнительные вычислительные ресурсы. Второй путь, рассматриваемый в данном параграфе, основан на повышении скорости прохождения транзакций через сетевую инфраструктуру, что достигается выделением дополнительной полосы пропускания наиболее важным приложениям за счет менее важных приложений.

АУТСКИЙ

Управление потоками активно используется для борьбы с сетевыми перегрузками, возникающими при превышении входящей нагрузкой пропускной способности сети [33]. Причинами перегрузки в УК чаще всего являются увеличение количества пользователей и объемов передаваемых данных.

Управление на уровне сетевого взаимодействия включает в себя оптимальную маршрутизацию и управление потоками [29, 33, 66, 100, 172]. Маршрутизация обладает ограниченными возможностями, а кардинальным решением является ограничение входящей в сеть нагрузки [33].

Эффективность управления потоками можно оценивать по: уменьшению задержки $\tau_{ср}$ при передаче пакетов; уменьшению потерь $\tau_{ср}$ при передаче пакетов; эффективности использования пропускной способности сети. В последнем случае в рассмотрение вводится коэффициент использования сетевых ресурсов (9.1).

Задача ИТ УИТИ при управлении сетевыми ресурсами заключается в том, чтобы поддерживать значение R'_C максимально близким к R_C, не допуская перегрузок в сети, т. е. осуществлять управление входящей и транзитной нагрузками так, чтобы выполнялось условие $K_c \to 1$.

В ИТ УИТИ целесообразно реализовать такие политики управления потоками:

- предоставление доступа к сетевым ресурсам в запрашиваемом объеме;
- ограничение доступа к сетевым ресурсам;
- отказ в доступе бизнес-процесса к сетевым ресурсам.

В работе [33] рассматриваются вопросы управления потоками информации с целью предотвращения потери работоспособности сети, вызванной запредельными входящей и транзитной нагрузками. Однако управление потоками в сетях рассматривается без учета их приоритетов.

Множество работ, например [184, 255, 378], связанных с агрегацией трафика, оптимизацией сетевых потоков, достижением максимальной загруженности сети и улучшением других ее показателей, посвящены решению различных задач сетевого управления, но при этом не учитывают требования к характеристикам информационных потоков, предъявляемые со стороны бизнес-процессов.

В работе [95] рассматриваются вопросы управления сетевыми ресурсами, но не решается задача управления трафиком. В работе [5] рассматриваются вопросы проверки эффективности управления мультимедийным трафиком. В работах [387, 388] применяются механизмы QoS, предоставляющие приложениям требуемую полосу без учета важности работающих приложений.

Решение всего комплекса задач адаптивного автоматического управления потоками при изменении значимости бизнес-процессов сложно реализовать на основе управления с помощью правил (Policy-Based Management, PBM) [195], поскольку изменение критериев оптимальности управляющих решений требует постоянного экспертного вмешательства. Кроме того, основным недостатком таких систем является их привязанность к нижним уровням модели OSI, не связанным напрямую с поддержкой бизнес-процессов. СУ по заранее определенным политикам позволяют улучшить показатели качества работы сети, например, сократить время неработоспособности, увеличить загрузку сети до предельных значений [323, 375], но при этом не учитывают приоритеты информационных потоков, задаваемые прикладным уровнем.

Управление потоками информации целесообразно осуществлять с учетом важности передаваемых данных для бизнес-процессов, а также значимости биз-

нес-процессов и процессов деятельности [114]. В этом случае трафик, создаваемый важными распределенными приложениями, получает привилегированные возможности передачи за счет ограничения трафика низкоприоритетных приложений. Несмотря на очевидность и перспективность такого подхода, в настоящее время превалируют методы управления потоками путем реализации в УК функции Traffic shaping, которая, в сущности, является тривиальным ограничением трафика без учета или с учетом приоритетов пакетов, если в УК поддерживается QoS. Управление потоками при изменении значимости бизнес-процессов сложно реализовать на основе управления с помощью правил [195], поскольку это также требует постоянного экспертного вмешательства. К тому же, такие СУ не связаны напрямую с поддержкой бизнес-процессов [323, 375].

Применение приоритетной передачи данных в соответствии со стандартом IEEE802.1p не позволит организовать приоритетную передачу данных для бизнес-процессов по следующим причинам:

- малое количество приоритетов;
- приоритеты назначаются сетевым администратором, который не в состоянии оперативно отследить изменение значимости бизнес процессов, кроме того, управление приоритезацией приложений не входит в сферу его компетенции;
- устаревшее сетевое оборудование не поддерживает стандарт IEEE802.1p;
- не все коммутаторы поддерживают все восемь очередей, определенных в стандарте IEEE802.1p;
- нельзя запретить интенсивную генерацию низкоприоритетного трафика.

Существует большое количество работ и обзоров по адаптивному управлению сетями, учитывающих требования к качеству передачи данных, предъявляемых отдельными приложениями [118]. В отличие от систем поддержки работы приложений, которые предоставляют средства для автоматизации их работы, но не дают возможности контроля и управления перегрузками в сети, СУ приложениями и потоками позволяют реализовать приоритетную передачу информации отдельных приложений. Работы [388, 387] рассматривают применение механизмов QoS, позволяющих предоставить каждому приложению требуемую полосу пропускания, однако не учитывают важность работающих приложений. За счет централизованного управления получение требуемого качества передачи достигается для каждого из предопределенных потоков, но при этом низкоприоритетные потоки без учета передаваемой в них информации могут требовать такого же качества обслуживания, что и высокоприоритетные. Это не позволяет строить СУ сетями, оптимизирующие загруженность каналов связи с учетом известных приоритетов информационных потоков и требований к качеству передачи, выдвигаемых со стороны бизнес-процессов.

Поэтому разработка и модернизация методов и средств управления потоками информации для предотвращения перегрузки каналов связи путем контроля скорости передачи данных конкретными рабочими станциями и серверами или прикладными программами с учетом значимости генерирующих их бизнес-процессов является актуальной задачей.

Необходимо разработать метод управления информационными потоками как часть инструментария ИТ УИТИ, учитывающий приоритеты потоков данных, генерируемых приложениями, которые поддерживают бизнес-процессы. Этот метод должен обладать способностью адаптации к изменению приоритетов потоков, загруженности каналов связи и отказам сетевого оборудования.

9.2. МЕТОД УПРАВЛЕНИЯ СЕТЕВЫМ ТРАФИКОМ В КОРПОРАТИВНОЙ IP-СЕТИ

9.2.1. Суть предлагаемого метода управления информационными потоками в сетях с коммутацией пакетов

Управление сетевыми ресурсами является актуальной задачей, включающей в себя оптимальную маршрутизацию, обеспечивающую распределение и перераспределение потоков информации, и управление потоками, предполагающее ограничение входящей нагрузки и нагрузки в узлах сети. Традиционно эти задачи рассматриваются и решаются независимо друг от друга без учета динамики важности бизнес-процессов или значимости распределенных приложений.

Один из основных комплексов задач, решаемых ИТ УИТИ, связан с управлением сетевыми ресурсами, поскольку от работоспособности корпоративной сети существенно зависит эффективность функционирования ИУС. Большинство СУ сетями ориентированы на управление только оборудованием с целью поддержания эксплуатационных характеристик и показателей работы сети на заданном уровне. При этом управление, как правило, осуществляется без учета типов информационных потоков, транспортируемых сетью, и их значимости для поддержания бизнес-процессов. Это приводит к тому, что при перегрузке сети ухудшаются параметры передачи всех потоков, независимо от их важности. Совершенно естественным является стремление разработать и использовать в СУИ механизмы управления сетью, учитывающие важность передаваемых потоков и, в случае перегрузки сети, ограничивающие передачу низкоприоритетных потоков с целью сохранения качества передачи высокоприоритетных.

Данный параграф посвящена разработке подсистемы управления информационными потоками (ПУИП), в которой реализована часть ИТ УИТИ, предназначенная для адаптивного управления информационными потоками в корпоративной IP-сети с учетом значимости передаваемой информации для поддержания бизнес-процессов. ПУИП управляет распределением и перераспределением потоков в сети с учетом условий их передачи, изменяющихся приоритетов бизнес-процессов и текущего состояния сети.

В основу предлагаемого метода управления трафиком положен учет требований к характеристикам информационного обмена со стороны бизнес-процессов [119].

В настоящее время для мониторинга работы и управления сетевым оборудованием применяется большое количество разнообразных СУ как дорогостоящих фирменных, так и условно бесплатных универсальных [33, 120, 154]. При этом превалирует подход обособленного решения отдельных задач сетевого управления без рассмотрения в тесной взаимосвязи проблем управления распределением общих информационных и телекоммуникационных ресурсов ИТ-инфраструктуры с учетом значимости распределенных приложений, поддерживающих бизнес-процессы или процессы деятельности. Более перспективным является подход к распределению ИТ-ресурсов с учетом значимости передаваемой информации для бизнес-процессов [114, 148, 156, 157]. В этом случае выделение сетевых ресурсов бизнес-процессам, операциям, распределенным приложениям и пользователям проводится с учетом их значимости для бизнеса, задаваемой системой управления производительностью бизнеса, а перераспределение ресурсов происходит при изменении значимости бизнес-процессов или возникновении нештатных ситуаций в сети, создающих перегрузки. Перегрузки сетевого оборудования

или каналов связи приводят к функциональным отказам и делают невозможным эффективное функционирование ИУС. Для борьбы с перегрузками используются методы маршрутизации и управления потоками информации.

За счет использования протоколов маршрутизации, адаптивных к перегрузкам и отказам, можно решить проблему сетевых перегрузок, однако, если причиной перегрузок является существенное увеличение входящей нагрузки, то методами адаптивной маршрутизации можно лишь частично сгладить эту проблему. Более эффективным для борьбы с сетевыми перегрузками является управление входящими потоками информации. При этом, естественно, управление должно строиться таким образом, чтобы в случае угрозы возникновения сетевых перегрузок ограничивать или запрещать передачу низкоприоритетных потоков без ухудшения качества передачи потоков, генерируемых высокоприоритетными распределенными приложениями.

Для реализации такого управления необходимо постоянно иметь данные о значимости бизнес-процессов и изменении этой значимости, регулярно получать информацию о загруженности сети, состоянии узлов и каналов связи, выявлять, оценивать и ограничивать или блокировать низкоприоритетные потоки информации, вырабатывая оптимальные управляющие воздействия и оперативно их реализовывая.

В этом случае ПУИП должна управлять сетевым оборудованием и потоками информации, получая информацию о функционировании прикладных процессов и их важности. Для этого ПУИП необходимо работать практически на всех уровнях модели OSI от канального до прикладного уровня, используя возможности и функциональность каждого из уровней для того, чтобы эффективно реализовывать на одних уровнях то, что нельзя сделать на других. Например, максимальная пропускная способность и загруженность сетевых ресурсов достигается при коммутации кадров на канальном уровне модели OSI, в то время, как для реализации возможности ограничения разных типов трафика в зависимости от сервиса или адресатов потоков требуется обработка пакетов на сетевом уровне, что вызывает дополнительные задержки при обработке каждого отдельного пакета. При обработке пакетов на уровнях выше сетевого задержки, вызванные этой обработкой, необходимой для управления трафиком, суммарно превышают требования по каждому из потоков и не могут применяться в современных сетях с существующими требованиями к передаче потоков.

В основу предлагаемого метода положен учет требований к характеристикам информационного обмена со стороны бизнес-процессов. Предложенный метод учитывает значимость передаваемой информации для поддержания бизнес-процессов и осуществляет распределение и перераспределение потоков в сети с учетом условий их передачи, изменяющихся приоритетов бизнес-процессов и текущего состояния сети.

Предложенный метод управления информационными потоками позволяет:

- поддерживать согласованный уровень важнейших услуг;
- повысить эффективность использования сетевых ресурсов;
- оптимизировать распределение сетевых ресурсов между бизнес-процессами, распределенными приложениями и пользователями.

Выделение сетевых ресурсов бизнес-процессам проводится с учетом их значимости для бизнеса, а перераспределение ресурсов происходит при изменении значимости бизнес-процессов или возникновении нештатных ситуаций в сети,

создающих перегрузки. Управление осуществляется так, что в случае угрозы возникновения сетевых перегрузок ограничивается или блокируется передача потоков, генерируемых низкоприоритетными приложениями.

Бизнес-процесс z_k из множества $Z = \{z_k\}$, $k = \overline{1,K}$ порождает L_k информационных потоков $\lambda_k^P = \{\lambda_{kl}^P\}$, $k = \overline{1,K}$, $l = \overline{1,L_k}$. Потоки всех бизнес-процессов создают суммарный трафик:

$$\Lambda^P = \sum_{k=1}^{K} \sum_{l=1}^{L_k} \lambda_{kl}^P . \qquad (9.4)$$

Для каждого потока данных λ_{kl}^P из множества λ_k^P задается приоритет Pr_{kl}, $k = \overline{1,K}$, $l = \overline{1,L_k}$, причем Pr_{kl} соответствует приоритету l-го приложения и принимает значения из $\Pi = \{1, ..., Pr_{\max}\}$, где $Pr_{\max}$ – максимальное значение.

Кроме потоков, порождаемых бизнес-процессами, в сетях передаются потоки, не имеющие непосредственного отношения к поддержанию бизнес-процессов. К таким потокам можно отнести потоки, создаваемые пользователями при использовании сетевых сервисов и выполнении действий, не связанных с процессами деятельности. Например, запуск сетевых игр, обмен мультимедийными файлами, просмотр потокового видео и др. При этом создается трафик

$$\Lambda^U = \sum_{m=1}^{M} \sum_{g=1}^{G_m} \lambda_{mg}^U , \qquad (9.5)$$

где λ_{mg}^U – один из g-х потоков, $g = \overline{1,G_m}$, создаваемый m-м пользователем, $m = \overline{1,M}$, M – количество пользователей ИУС, G_m – количество потоков, связанных с рабочей станцией m-го пользователя. Потоки λ_{mg}^U, $m = \overline{1,M}$, $g = \overline{1,G_m}$, можно легко контролировать. Трафик Λ^U ограничивается или блокируется ПУИП в первую очередь, если он изначально не ограничен или не запрещен корпоративной политикой. Каждому потоку λ_{mg}^U могут быть присвоены приоритеты со значениями из П.

Присвоение процессам или приложениям и, соответственно, потокам значений приоритетов осуществляется, исходя из значимости процессов или приложений, определяемых на основе анализа производственного или жизненного цикла предприятия, а также сложившейся ситуации на рынке или в отрасли. Значимость бизнес-процессов может быть определена экспертным путем, в результате работы системы автоматизации управления выполнением бизнеса, директивными указаниями руководства и другими способами. В ходе работы предприятия приоритеты бизнес-процессов и приложений могут изменяться и ПУИП должна оперативно реагировать на такое изменение.

Весь остальной трафик Λ^T сети является технологическим и связан с необходимостью поддержания работоспособности ИУС. Потоки трафика Λ^T связаны с работой СУИ, резервным копированием данных, обновлением программного

обеспечения ИТ-системы и другими подобными задачами. Они также контролируются и управляются ПУИП. Всем потокам этого трафика или генерирующим их приложениям присваивается приоритет из множества П.

Суммарный трафик Λ^{Σ} ИУС определяется как

$$\Lambda^{\Sigma} = \Lambda^{P} + \Lambda^{U} + \Lambda^{T}, \tag{9.6}$$

и он не должен превышать суммарной пропускной способности сети C^{Σ}, т. е. должно выполняться условие:

$$\Lambda^{\Sigma} \le C^{\Sigma}. \tag{9.7}$$

Учитывая тот факт, что выполнение условия (9.7) не исключает перегрузок отдельных каналов, целесообразно управлять входящими потоками таким образом, чтобы выполнялось условие:

$$\Lambda_i^{\Sigma} \le C_i. \tag{9.8}$$

для всех $i = \overline{1, I}$, где I – максимальное количество каналов связи, Λ_i^{Σ} – суммарный трафик в i-м канале, C_i – пропускная способность i-го канала, причем $C^{\Sigma} = \sum_{i=1}^{I} C_i$.

Реализация в ПУИП алгоритмов работы, обеспечивающих выполнение условия (9.8), требует сбора и обработки гораздо большего объема информации, чем при работе по условию (9.7), однако это позволяет избежать локальных перегрузок в сети и существенно повысить качество работы распределенных приложений.

Входными данными для ПУИП являются сведения о загруженности отдельных каналов связи, выходными – команды управления сетевым оборудованием, ограничивающие или запрещающие прохождение отдельных потоков. Управление потоком осуществляется в каждом из узлов сети на входе в узел. При этом ПУИП стремится пропустить через каждый узел сети максимальное количество потоков с требуемыми параметрами качества обслуживания.

Значения приоритетов информационных потоков Λ^{P} и Λ^{T}, а также требования к качеству обслуживания этих потоков задаются соответствующей таблицей. Пример фрагмента таблицы выглядит следующим образом (табл. 9.1).

Таблица 9.1. Приоритеты потоков и требования к качеству передачи

	Приложение	Порт	Приоритет	Требуемая полоса пропускания, кб/с	Максимальная задержка, мс
1	ПУИП	22, 23, 80	1	1024	1024
2	SAP	3201–3299	2	1024	1024
3	Система биллинга	3306, 1527	3	512	1024
4	1С Бухгалтерия	512	4	128	512
5	Электронная почта	25, 110, 143	5	64	256
6	Обмен файлами	139, 445	6	512	256

Архив КП

Ограничение потоков возможно для приложения, конкретного пользователя и отдельного сетевого сервиса. Ограничение потоков информации проводится только после полной утилизации сетевых ресурсов.

Оптимальные маршрутизация и агрегация трафика с использованием MPLS – коммутации пакетов по меткам позволяют достичь максимального использования ресурсов сети и применять механизмы QoS для достижения требуемых показателей качества по каждому из потоков. Однако при этом возможны случаи перегрузки сети и отсутствуют механизмы, позволяющие исправлять проблемы такого рода. В связи с невозможностью ограничения потоков информации на канальном уровне ПУИП вынуждена управлять потоками информации на сетевом уровне.

9.2.3. Разработка структуры подсистемы управления информационными потоками

Для эффективного управления использованием сетевых ресурсов, предотвращения негативных последствий перегрузок и неисправностей сетевых элементов в ПУИП реализуются следующие группы функций:

- управление маршрутизацией в сети, агрегирование трафика и эффективное использование всей полосы пропускания каналов связи, реализуемые на сетевом уровне;
- управление входящими потоками при возникновении угроз перегрузки сети или каналов связи, включающее ограничение низкоприоритетных потоков трафика Λ^P, существенное ограничение или блокирование потоков трафика Λ^U;
- локализация и управление устранением неисправностей в сети.

Среднее время задержки D при передаче пакета по сети можно оценить с помощью формулы:

$$D = H \times S \times (R + P / C), \tag{9.9}$$

где H – количество шагов ретрансляции, S – средняя длина очереди маршрутизатора, R – среднее время на решение задач маршрутизации в отдельном УК, P – средняя длина пакетов, а C – скорость передачи по линии связи, которая при оценивании считается одинаковой для всех каналов.

Управление параметром H осуществляется в ПУИП протоколами динамической маршрутизации и оптимизацией сетевых потоков.

Оптимальное управление потоками осуществляется по критерию поддержки максимально возможной скорости следования пакетов при отсутствии потерь из-за переполнения входных буферов. Запишем его в виде

$$S_q - S_q(t) \geq 0, \text{ при } \tau(t) + u(t) \to \min, \tag{9.10}$$

где S_q – размер входного буфера, $S_q(t)$ – текущее значение очереди, $\tau(t)$ – межкадровый интервал, $u(t)$ – переменная управления.

Структура предлагаемой ПУИП представлена на рис. 9.2.

Эффективная реализация функций ПУИП обеспечивается продуманным взаимодействием модулей принятия решений, формирования команд, обработки данных мониторинга, сопряжения с сетевым оборудованием, взаимодействия с администратором и логирования.

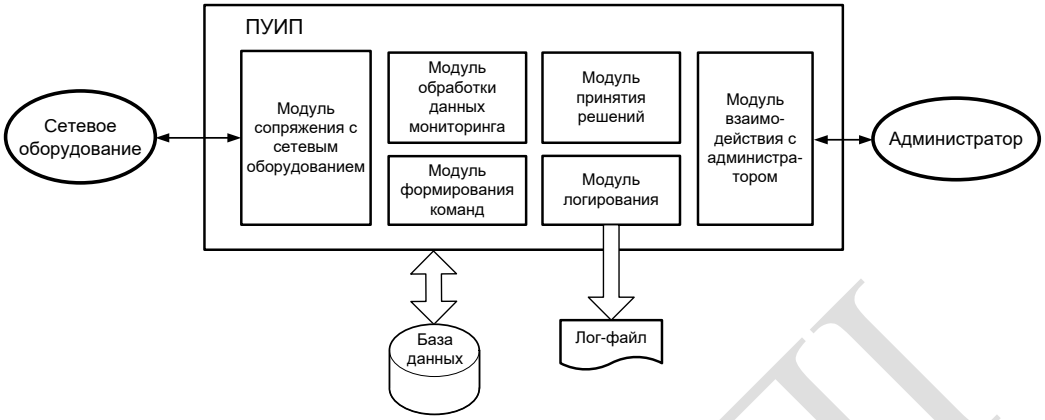


Рис. 9.2. Структура ПУИП

Основную роль в управлении информационными потоками играет модуль принятия решений. Он осуществляет анализ состояния сети и отдельных трактов передачи данных и на этой основе принимает решения о необходимости управления сетевым оборудованием для улучшения состояния сети и выполнения действий на оборудовании для устранения последствий неисправностей. Также этот модуль изменяет частоту получения данных мониторинга и задает степень детализации этих данных, необходимую для локализации неисправностей или получения более точных сведений о состоянии сетевых элементов. Для ускорения работы ПУИП этот модуль освобожден от формирования текстов команд и первичной обработки результатов мониторинга. Эти функции делегированы модулям формирования команд и обработки данных мониторинга соответственно.

Структура таблиц БД ПУИП приведена на рис. 9.3.

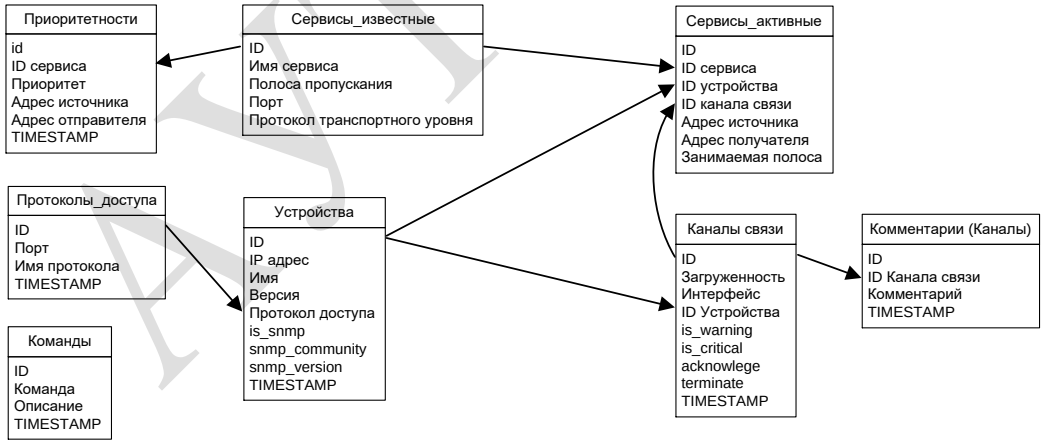


Рис. 9.3. Структура базы данных ПУИП

Хранение всех команд, доступных для выполнения, в отдельной таблице позволяет использовать только идентификаторы команд в пределах ПУИП.

В процессе работы подсистемы собираются данные с сетевых устройств, анализируется состояние загруженности и работоспособности отдельных каналов

связи и сети в целом. При этом определяется, какие потоки информации проходят через каждый интерфейс узлов сети и какую полосу пропускания они занимают.

Если канал связи загружен более чем на 80%, администратору выводится предупреждение о возможной перегрузке сети или отдельных трактов. Когда загрузка отдельного канала превышает 90%, администратору выводится предупреждение о возможной перегрузке сети или отдельных трактов. Когда загрузка канала превышает 95%, ПУИП автоматически предпринимает действия по ограничению входящей нагрузки и снижению загруженности канала. Для этого на сетевом уровне модели OSI осуществляется перебор списка потоков информации, проходящих через интерфейс, и выдается управляющее воздействие на сетевой узел на запрет прохождения или ограничения полосы пропускания некоторых потоков. Причем потоки, попавшие в таблицу приоритетности, будут ограничены по полосе пропускания до требований, указанных в таблице. Потоки, не числящиеся в таблице приоритетов, будут автоматически последовательно запрещаться для прохождения по каналу связи, близкому к перегруженному состоянию. Данная процедура будет выполняться до тех пор, пока загрузка канала не снизится ниже заданного порогового значения. Если этого будет недостаточно и в сети будут наблюдаться перегруженные участки, ПУИП ограничивает наименее приоритетные потоки Λ^P и запрещает потоки Λ^U. При дальнейшем снижении загруженности каналов запрещенные потоки будут постепенно разрешаться. Значения порогов 80%, 90% и 95% – значения, устанавливаемые по умолчанию. В процессе работы администратор может их корректировать.

Для получения от сетевого оборудования списков активных информационных потоков используются программные средства удаленного доступа и выполнения удаленных команд. Управление включением и отключением потоков информации на сетевом устройстве автоматически выполняется разработанным программным обеспечением ПУИП и реализуется модулем формирования команд и программным модулем сопряжения с сетевым оборудованием, использующим Expect::Perl.

На вход программного модуля сопряжения подается запрос, содержащий IP-адрес или имя устройства при наличии DNS сервера, а также команда для выполнения или пакет команд. После этого ожидается ответ от оборудования. Полученные ответы записываются в БД для ведения операционной и аналитической деятельности. В результате анализа загрузки каналов связи определяется необходимость выполнения дополнительных действий на каждом из каналов связи ИУС.

На рис. 9.4 приведен пример команд, используемых для ограничения потока информации.

Для эффективной работы ПУИП важно знать величину и нагрузку каждого потока на сеть. В зависимости от того, что необходимо видеть – текущую загрузку или общее количество пакетов, которые проходят через сетевое устройство, могут быть использованы различные способы получения необходимой информации. Первый способ предполагает выполнение удаленных команд и анализ полученных от оборудования ответов. Такой способ позволяет получить значение общего числа пакетов, переданных по каналу связи и относящихся к отдельному потоку, но не позволяет получить информацию о текущей нагрузке, создаваемой данным потоком для сети. Этот способ можно использовать только для сбора аналитической информации. Кроме того, он не требует дополнительных настроек оборудования под каждый из потоков.

```
------------ Настройка ----------------
access-list 1 permit tcp any eq 80
access-list 1000 deny ip any
!
interface ethernet 1
       ip policy route-map PBR
!
route-map PBR permit 10
        match ip address 1
        set ip precedence priority
        set ip next-hop 1.1.1.2
route-map PBR deny 1000
         match ip address 1000
------------ Настройка ----------------
```

Рис. 9.4. Пример команд ограничения потока информации

Второй способ заключается в активации на каждом порту средств расчета объема трафика по каждому из сервисов, разработанных производителем сетевого оборудования и использующих его специфические алгоритмы. Несмотря на то, что такие расчеты требуют от оборудования дополнительной процессорной мощности, они позволяют получать точную информацию по каждому из сервисов в текущий момент времени. Именно такой способ используется в ПУИП. Активация программных средств подсчета трафика осуществляется командой ip nbar protocol discovery на нужном интерфейсе. Вывод команды просмотра имеет вид, приведенный на рис. 9.5 (данные будут содержаться в таблице «Активные сервисы»):

```
Device# show ip nbar protocol-discovery
 FastEthernet0/0
                        Input                     Output
                        -----                     ------
Protocol                Packet Count              Packet Count
                        Byte Count                Byte Count
                        30sec Bit Rate (bps)      30sec Bit Rate (bps)
                        30sec Max Bit Rate (bps)  30sec Max Bit Rate (bps)
----------------        ------------------------  ------------------------
ssh                     27524274                  699300
                        4420992684                145872150
                        4000                      3000
                        3037000                   271000
```

Рис. 9.5. Пример команд просмотра загруженности каналов

Ограничение потоков выполняется по заданной таблице приоритетов и требований показателей качества каждого из потоков. ПУИП имеет возможность ограничить пропускную способность для каждого пользователя сети по всему потоку сразу или по группе потоков и пользователей. Выполнение процедуры ограничения сводится к выполнению команд на сетевом оборудовании.

После выполнения действий по ограничению IP-потока СУИП оповещает пользователя о запрете данного потока и возможных причинах такого ограничения. Для оповещения приложения используются протоколы транспортного и сетевого уровня, которые доводят до приложения информацию о запрете потока и, в ряде случаев, причинах такого запрета потока. Для оповещения пользователей могут быть использованы, например, стандартные средства ОС Windows (Messenger Service и netsend), которые позволяют вывести на экран пользователя сообщение с необходимым текстом или агентские технологии ПУИП [117].

Одна из задач, решаемых ПУИП, заключается в минимизации времени неработоспособности наиболее приоритетных потоков. ПУИП в случае отказа канала связи или сетевого оборудования сначала попытается перераспределить потоки, используя динамический протокол маршрутизации OSPF. Протокол автоматически перестраивает таблицы маршрутизации всех маршрутизаторов в сети, обходя отказавший канал передачи данных или оборудование.

Если этого будет недостаточно и в сети будут наблюдаться перегруженные участки, ПУИП ограничивает наименее приоритетные потоки Λ^P и запрещает потоки Λ^U. Для этого ПУИП постоянно собирает данные об активности потоков на каждом из интерфейсов работающего сетевого оборудования и при возникновении нештатной ситуации определяет потоки, которые должны быть запрещены для ликвидации перегрузки каналов связи.

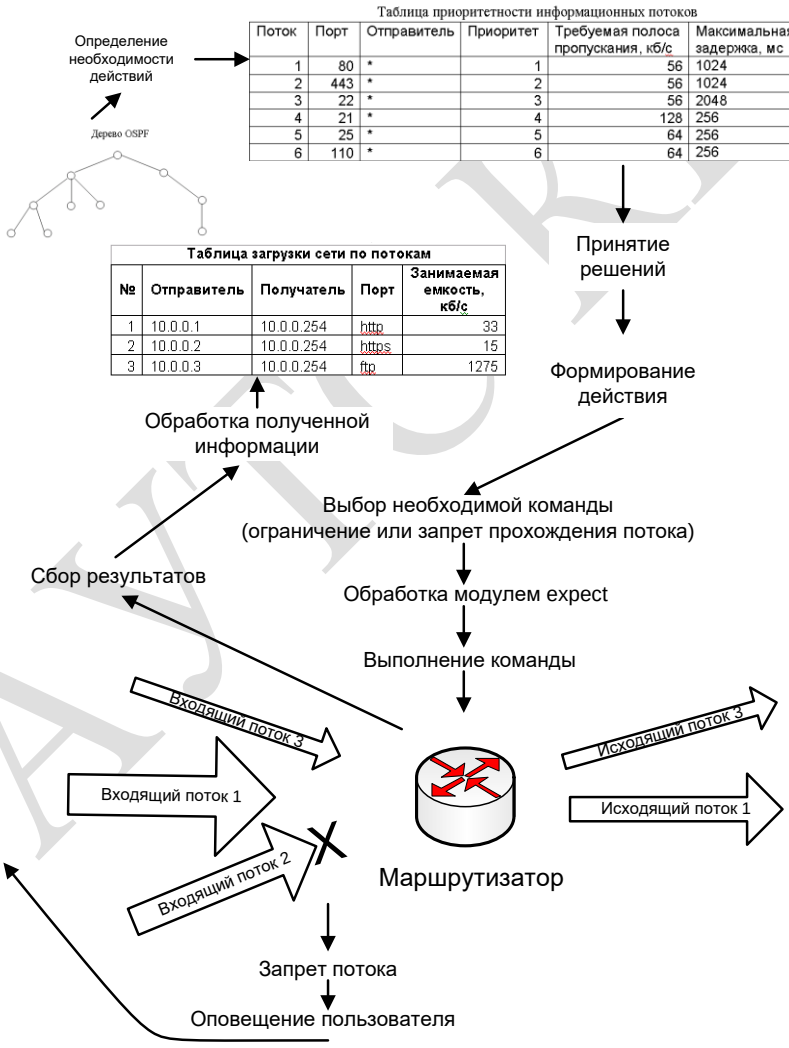


Рис. 9.6. Схема работы ПУИП

ПУИП ведет анализ активности сервисов на каждом из интерфейсов и находит закономерности в перегрузке каналов одними и теми же сервисами в сети. Подсистема может вырабатывать управляющие воздействия и ограничивать полосу пропускания для сервисов и исключать возможные всплески сетевой активности.

Схема работы ПУИП на примере обслуживания шести потоков приведена на рис. 9.6.

При этом поток 1 ограничивается, последующая передача потока 2 запрещается, а поток 3 передается без изменений.

Основным достоинством предлагаемой ПУИП является способность ограничения или блокирования только конкретных потоков отдельных пользователей. Реализация этой способности требует дорогостоящего оборудования и приводит к дополнительным задержкам в сети.

Предлагаемая ПУИП предназначена для использования в корпоративных IP-сетях, построенных на основе маршрутизаторов Cisco (серии не ниже 72xx). Система устанавливается на ОС Unix. Для работы системы требуется поддержка языков C, PHP, Perl. Производительность протокола каждого порта маршрутизатора должна быть не меньше средней интенсивности суммарного трафика, проходящего через порт. Производительность внутренней шины маршрутизатора должна быть не меньше средней интенсивности суммарного трафика, передаваемого между портами, принадлежащими разным модулям маршрутизатора.

Таким образом, предложен метод управления потоками в корпоративной IP-сети с учетом загруженности каналов связи и приоритетов потоков данных, позволяющий поддерживать высокоприоритетные услуги на согласованном уровне.

9.3. ИССЛЕДОВАНИЕ МОДЕЛЕЙ УПРАВЛЕНИЯ ПОТОКАМИ ДАННЫХ В КОРПОРАТИВНЫХ IP-СЕТЯХ

9.3.1. Постановка задачи исследования

Активное внедрение ИТ с целью автоматизации как можно большего количества процессов на предприятиях влечет и негативные последствия. Они могут проявляться в перегрузке телекоммуникационных сетей, обеспечивающих информационное взаимодействие между компонентами ИТ-инфраструктуры, что, в свою очередь, приводит к снижению эффективности или нарушения работы ИТ. Эта проблема особенно ощутима в крупных корпорациях, которые объединяют много филиалов и имеют развитую сеть региональных офисов. Корпоративные IP-сети, объединяющие большое количество пользователей, распределенных на больших территориях, традиционно создаются на основе дорогого коммуникационного оборудования, соединенного каналами связи с ограниченной пропускной способностью. Внедрение новых ИТ влечет за собой увеличение нагрузки на сети. Это сопровождается существенными экономическими затратами на модернизацию корпоративных IP-сетей для увеличения их суммарной пропускной способности. Поэтому одной из важнейших задач, решение которой в корпоративных сетях возлагается на СУИ, является управление информационных потоками с целью эффективного использования пропускной способности корпоративной сети без локальных и общих перегрузок [113, 119].

Управление информационными потоками чаще всего используется для борьбы с сетевыми перегрузками, которые возникают при превышении входным

нагрузкой пропускной способности сети. В работе [33] рассматриваются вопросы управления потоками информации с целью предотвращения потери работоспособности сети, вызванной чрезмерным входным и транзитным трафиком, но не уделяется внимание вопросам построения управляющего устройства.

В работе [95] рассматриваются вопросы динамического управления ресурсами сети на основе математических моделей, описанных в пространстве состояний. Рассмотрены некоторые подходы к моделированию сетей – детерминированный, стохастический и адаптивный. Основной недостаток этих подходов заключается в том, что в них рассматриваются лишь моделирования сети, а не задача управления трафиком.

В работе [5] рассматриваются вопросы проверки эффективности управления мультимедийным TCP-трафиком, внедренным на базе NDIS-драйвера, обеспечивающего работу TCP-подобного протокола в среде Matlab на уровне доступа к каналу связи и способного имитировать динамику интернета при передаче трафика локальной сетью. Недостатком подхода является то, что динамика работы не учитывает программирования в значительных объемах среды Matlab и необходимость внедрения управления на нижних уровнях модели OSI.

Целью данного параграфа является моделирование различных подходов к управлению потоками данных в корпоративной информационно-телекоммуникационной сети, которая использует IP-протокол, и сравнение их эффективности в решении проблемы предотвращения перегрузки в сети.

9.3.2. Модели управления потоками данных

Исследуются модели трех регуляторов, которые могут быть применены в ПУИП. Модели первая и вторая – аппаратный и программный регуляторы, работа которых соответствует алгоритмам, используемым в сетевом оборудовании. Третья модель – оптимальный по быстродействию регулятор, реализующий подход к управлению информационными потоками, основанный на применении методов ТАУ.

Аппаратный регулятор работает по методу управления трафиком в маршрутизаторах – policing, когда осуществляется задержка или отбрасывание пакетов с целью ограничения нагрузки на каналы связи. Управление реализуется путем ограничения информационного потока в случае превышения им заданного уровня. Таким образом, часть передаваемых данных отбрасывается.

Модель ПУИП с аппаратным регулятором приведена на рис. 9.7.

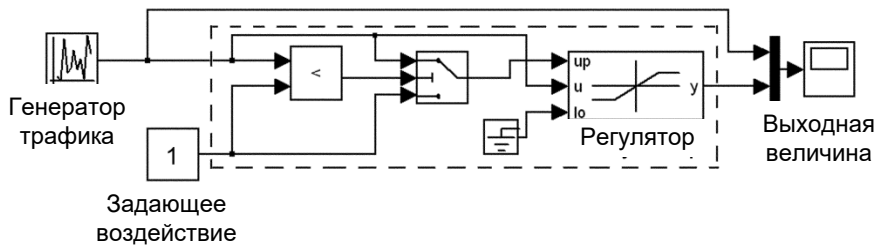


Рис. 9.7. Модель ПУИП на основе аппаратного регулятора

В данном случае регулятор работает следующим образом: если сравнение целевой и фактической величины потока данных в регуляторе свидетельствует о

необходимости ограничения потока, то трафик, превышающий целевое значение, отбрасывается. На рис. 9.8 приведена переходная характеристика аппаратного регулятора, когда целевое значение равно 1, сплошная линия – трафик после ограничения $\lambda_P(t)$, пунктирная – входной поток $\lambda(t)$.

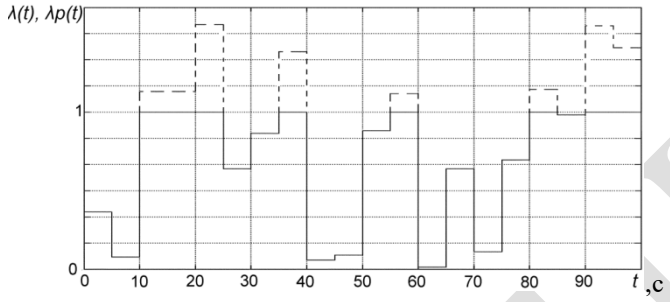


Рис. 9.8. Переходная характеристика аппаратного регулятора

Преимуществом этого способа управления потоками является простота реализации и универсальность регулятора, поскольку он аналогичным образом управляет всеми видами трафика. Однако имеется существенный недостаток – такое управление не является регулированием, оно, скорее, сводится к ограничению в случаях перегрузок в каналах передачи данных. При этом не вырабатывается никакое управляющее воздействие на генераторы трафика, обеспечивающие уменьшение количества передаваемой информации с целью исключения перегрузок, потерь информации и повторной передачи. Такой путь, в итоге, приводит к избыточному неэффективному использованию коммуникационных ресурсов и снижению эффективности функционирования распределенных приложений.

Программный регулятор (рис. 9.9) работает следующим образом: если после сравнения целевой и фактической величины потока данных возникает необходимость ограничить поток, то регулятор постепенно снижает коэффициент передачи, пока поток не будет достаточно ограничен. Однако в связи с особенностями данного метода возникает задержка вынесения решения, обусловленная необходимостью постоянного анализа загруженности каналов, сбора информации о пакетах, которые поступили, и других параметрах. При этом возникает ситуация, когда поток уже достаточно ограничен, но из-за отсутствия информации о загруженности каналов продолжается дальнейшее ограничение потоков. Когда же выясняется, что потоки слишком ограничены, регулятор осуществляет скачкообразное повышение коэффициента передачи, что влечет за собой пилообразную загруженность каналов связи, которую можно наблюдать на рис. 9.10.

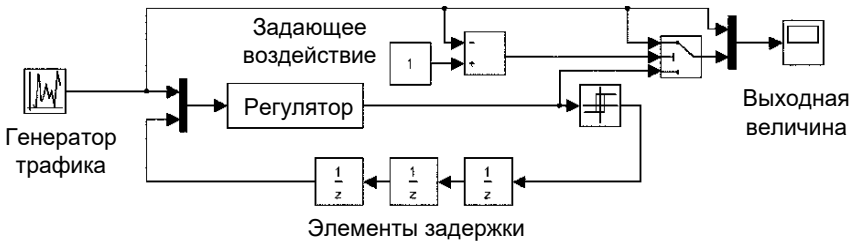


Рис. 9.9. Модель управления потоками трафика на основе программного регулятора

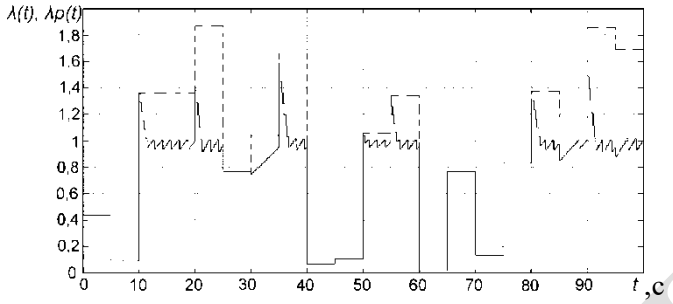


Рис. 9.10. Зависимость величины потоков от времени в системе с программным регулированием (сплошная линия $\lambda_p(t)$) и без него (пунктирная линия $\lambda(t)$). Желаемая загруженность канала равна 1

Преимуществом этого регулятора является универсальность, так как он позволяет регулировать разные типы трафика. Недостатком является большое влияние на качество управления задержек сбора и обработки информации и распространения управляющего воздействия, что отрицательно влияет на качество управления. Поскольку изменение коэффициента происходит линейно, большие задержки вызывают несвоевременное и несоответствующее управляющее воздействие.

Перспективным в решении задачи управления потоками сетевого трафика является использование принципов и методов ТАУ. Для того, чтобы рассмотреть сеть передачи данных и ПУИП с точки зрения ТАУ, необходимо представить их как систему автоматического регулирования (САР). Исследование любой САР начинается с формализации понятий и объектов, а также с формулирования целей и критериев управления.

Классическая схема САР содержит объект регулирования (ОР) и устройство управления, которое, анализируя регулируемую величину (т. е. выход объекта управления, подаваемого на регулятор с помощью обратной связи) и входное влияние, синтезирует необходимое управление. Формализуя задачу управления потоками, определим, что объектом управления является РОУ (станция, подсеть или любая совокупность объектов сети), генерирующий потоки информации. При этом в качестве выходной (регулируемой) величины примем величину загруженности канала связи, входного воздействия – пропускную способность канала передачи информации, а ошибки системы – величину перегрузки, возникающей в сети при чрезмерных потоках информации. Целью управления является минимизация ошибки системы за минимальное время. Данную систему можно рассматривать как разновидность дискретной системы стабилизации, поскольку при отсутствии перегрузок управление потоками не осуществляется, в противном случае – регулятор пытается поддерживать величину потока на целевом значении.

Ключевым моментом реализации подхода является определение математической модели, которая должным образом описывает поведение ОР. Обычно такая математическая модель представляет собой дифференциальное уравнение или систему дифференциальных уравнений, характеризующих динамику объекта управления в зависимости от различных входных воздействий. В реальных сетях генерирование трафика – стохастический процесс. Единственный способ, которым так или иначе можно повлиять на величину потока информации – ограничить его. Именно значение этого ограничения в зависимости от величины перегрузки и задержек в сети и является конечным управляющим воздействием.

Математическая модель объекта управления описывается передаточной функцией [126]:

$$W_{OP}(z) = \frac{(0.17z^2 - 0.179z + 0,544)(z^2 - 1,863z + 1,027)(z^2 - 1,973z + 1)}{(z^2 - 0,689z + 1)(z^2 - 1,9z + 1)(z^2 - 1,973z + 1)}. \quad (9.11)$$

Зная передаточную функцию ОР и выбрав желаемые характеристики замкнутой системы, можно синтезировать регулятор, который обеспечит конечной системе необходимые показатели качества. В данном случае желаемыми показателями качества являются отсутствие статической ошибки и максимальное быстродействие системы. В соответствии с этим и будет синтезироваться регулятор.

Замкнутая система описывается следующим выражением:

$$W_З(z) = \frac{W_P(z) \cdot W_{OP}(z)}{1 + W_P(z) \cdot W_{OP}(z)}, \quad (9.12)$$

где $W_З(z)$ – передаточная функция замкнутой системы, $W_P(z)$ – передаточная функция регулятора, $W_{OP}(z)$ – передаточная функция ОР.

Отсюда передаточную функцию регулятора можно представить как

$$W_P(z) = \frac{W_З(z)}{W_{OP}(z) \cdot (1 - W_З(z))}. \quad (9.13)$$

Синтез регулятора выполняется в несколько этапов.

Первый этап – факторизация передаточной функции ОР:

$$W_{OP}(z) = \frac{B(z)}{C(z)} = \frac{B(z)}{(z-1)^{r_0} \cdot C^-(z) \cdot C^+(z)}, \quad (9.14)$$

где полиномы $C^-(z)$ и $C^+(z)$ имеют нули вне круга и в кругу единичного радиуса соответственно; r_0 – степень астатизма.

Для рассматриваемого объекта все корни лежат вне круга единичного радиуса, поэтому $C^-(z) = 1$.

Второй этап – выбор вида желаемой передаточной функции замкнутой системы. Из возможных вариантов выбираем тот, который обеспечит желаемый процесс в произвольные моменты времени и придаст системе астатизм порядка $r = 1$ по заданному входному воздействию. Желаемая передаточная функция замкнутой системы такого типа описывается следующим выражением:

$$W_З(z) = \frac{B(z) \cdot M(z)}{Q(z)}, \quad (9.15)$$

где $M(z)$ – полином, $Q(z)$ – характеристический полином замкнутой системы.

Ошибка системы описывается выражением

$$1 - W_З(z) = \frac{(z-1)^r \cdot C^-(z) \cdot N(z)}{Q(z)}, \quad (9.16)$$

где $N(z)$ – полином.

АУТСКП

Характеристический полином для системы с максимальным быстродействием находим по формуле

$$Q(z) = z^k, \tag{9.17}$$

где

$$K = \|B\| + \|C^-\| + r = 7\,, \tag{9.18}$$

$$Q(z) = z^k, \tag{9.19}$$

Степени полиномов $M(z)$ и $N(z)$ определяются следующим образом:

$$\begin{aligned} \|M\| &= \|C^-\| = 0, \\ \|N\| &= \|B\| = 6. \end{aligned} \tag{9.20}$$

Соответственно полиномы $M(z)$ и $N(z)$ приобретают вид

$$\begin{aligned} M(z) &= m_0, \\ N(z) &= n_6 z^6 + n_5 z^5 + n_4 z^4 + n_3 z^3 + n_2 z^2 + n_1 z + n_0. \end{aligned} \tag{9.21}$$

Для определения значений коэффициентов полиномов (9.21) используем равенство

$$B(z) \cdot M(z) + (z-1) \cdot C_1^-(z) \cdot N(z) = Q(z). \tag{9.22}$$

Подставив (9.19) и (9.21) в (9.22) и решив (9.22), получим коэффициенты искомых полиномов.

Из выражений (9.14)–(9.16) получаем следующую формулу для вычисления передаточной функции регулятора:

$$W_P(z) = \frac{(z-1)^{r_0 - r} \cdot C^+(z) \cdot M(z)}{N(z)}. \tag{9.23}$$

Подставив все необходимые данные в (9.23), получаем рассчитанную передаточную характеристику регулятора:

$$W_P(z) = \frac{430z^6 - 2083z^5 + 4517z^4 - 5728z^3 + 4517z^2 - 2083z + 430}{z^7 - 73z^6 + 357z^5 - 947z^4 + 1624z^3 - 1712z^2 + 991z - 240}. \tag{9.24}$$

Модель ПУИП на основе оптимального по быстродействию регулятора приведена на рис. 9.11, а результаты моделирования ПУИП в среде Matlab с рассчитанным регулятором показаны на рис. 9.12 и 9.13.

Использование регулятора, построенного на основе методов ТАУ, для управления трафиком позволило получить быстрый процесс регулирования с установленным значением, которое находится в заданных пределах. Недостатком такого регулятора является эффективная работа только с четко описанным объектом управления, поскольку только при таких условиях будут достигаться желаемые критерии качества работы ПУИП. В реальных сетях четко описать объект управления, осуществляющего генерацию трафика, невозможно, а любое его математическое представление является только приближением.

АУТСКП

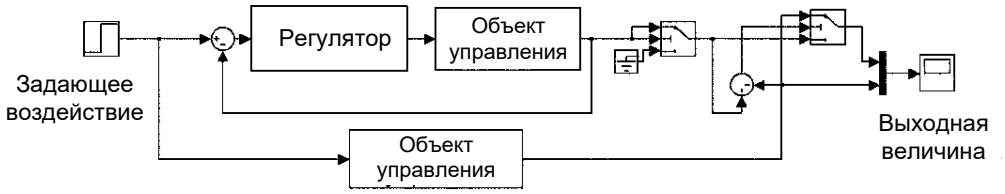


Рис. 9.11. Модель ПУИП на основе оптимального по быстродействию регулятора

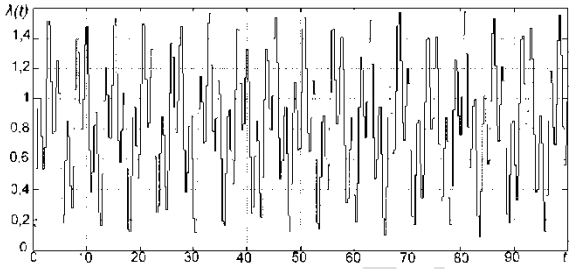


Рис. 9.12. Переходная характеристика системы без регулятора

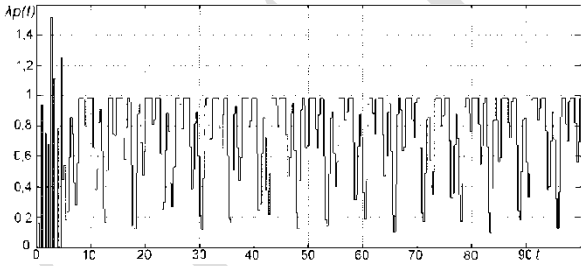


Рис. 9.13. Переходная характеристика систем управления потоками на основе оптимального по быстродействию регулятора

Возможность применения ТАУ в сфере телекоммуникационных сетей в целом и для решения поставленной задачи частности вызывает несомненный интерес, поскольку математический аппарат ТАУ позволяет достаточно четко рассчитать управляющее устройство, что придает ПУИТ желаемых показателей качества даже при наличии задержек в системе.

После моделирования описанных выше подходов к управлению потоками информации в сети можно выполнить сравнение данных регуляторов по следующим критериям: быстродействие регулятора, т. е. скорость, с которой регулятор выполняет необходимое управляющее воздействие; влияние задержек передачи данных на своевременность регулирующего воздействия; виды трафика, которые способен регулировать регулятор; простота реализации и расчета регулятора; эффективное использование имеющихся сетевых ресурсов.

По скорости выработки управляющего воздействия первая модель беспрекословно занимает лидирующую позицию, поскольку не требует глубокого анализа ситуации и, соответственно, времени на выработку и передачу управляющего воздействия, так как ограничение происходит сразу при превышении пропускной

способности канала передачи информации. Другие два регулятора в случае значительных задержек при передаче управляющего воздействия не могут полностью исключить перегрузку. При этом во второй модели такие задержки вызывают пилообразный вид установившегося значения загруженности канала.

Максимальное использование пропускной способности канала связи достигается при применении первой модели, поскольку канал загружается полностью вплоть до появления перегрузок. При использовании двух других моделей при появлении избыточного количества потоков происходит частичная разгрузка канала за счет реализации управления. Но нет оснований утверждать, что наиболее эффективное использование ресурсов обеспечивается первой моделью, поскольку из-за отсутствия регулирования интенсивности потоков при перегрузках возникает необходимость повторной передачи отброшенных данных, что приводит к дополнительному использованию сетевых ресурсов. Гораздо меньше перегрузок, а значит, и необходимости повторной передачи пакетов данных достигается при использовании второй и третьей моделей. Именно это свидетельствует о более эффективном использовании ресурсов сети.

По объему расчетов и реализации последний, рассчитанный с помощью методов ТАУ, регулятор является самым сложным. Но качество регулирования трафика таким регулятором намного превышает показатели предыдущих моделей. Наибольшим недостатком этого регулятора является возможность работы только с четко заданным трафиком. В тоже время, виды регулируемых потоков трафика при использовании первой и второй моделей регуляторов не ограничены, за исключением случая, когда время, необходимое регулятору на коррекцию коэффициента передачи, на порядок меньше при изменении величины потока.

Таким образом, регулятор, построенный на основе методов ТАУ, позволяет получить быстрый процесс регулирования. Однако узкий диапазон эффективной работы, ограниченный только четко описанным объектом управления, затрудняет его применение в реальных сетях. Но в сетях, функционирующих только с четко заданным видом трафика, его преимущества бесспорны.

Результаты сравнения регуляторов приведены в табл. 9.2.

Таблица 9.2. Результаты сравнения регуляторов

Критерий сравнения	Номер модели		
	1	2	3
Скорость выработки управляющего воздействия	+	-	-
Задержка в передаче управляющего воздействия	-	+	+
Возможность возникновения перегрузки	-	+	+
Использование пропускной способности канала	+	-	-
Возникновение необходимости повторной передачи	+	-	-
Простота расчетов и реализации регулятора	+	+/-	-
Качество регулирования	-	-	+
Чувствительность к закону изменения трафика	-	-	+
Адаптация к задержкам при принятии решений	-	-	+

Таким образом, разработаны и исследованы модели для трех методов управления сетевым трафиком, не допускающих перегрузок каналов связи.

9.4. МОДЕЛЬ ПОДСИСТЕМЫ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ПОТОКАМИ С АДАПТИВНЫМ РЕГУЛЯТОРОМ

9.4.1. Постановка задачи исследования

Современные методы коммуникации, в частности видео- и аудиоконференции, работа с удаленными базами данных, свойственная распределенным информационным системам, приводят к существенному увеличение объема информации. Генерация больших объемов трафика приводит к возникновению локальных и глобальных перегрузок каналов связи и коммуникационного оборудования, что влечет за собой снижение эффективности работы всей сети передачи данных. Это непосредственно влияет на производительность ИУС, обеспечивающих поддержку выполнения бизнес-процессов и процессов деятельности предприятий и организаций с распределенной ИТ-инфраструктурой. Поэтому возникает проблема повышения качества передачи высокоприоритетных данных.

Существует два пути решения этой проблемы – экстенсивный и интенсивный. Первый путь связан с внедрением высокоскоростных технологий передачи данных, переходом на оптоволоконные линии связи, повышением производительности сетевого оборудования и т. п. Такой путь требует существенных капиталовложений как на этапе построения высокоскоростных сетей передачи данных, так и на этапе их последующего обслуживания. Поэтому для большинства участников процесса обмена информацией в локальных и глобальных сетях из экономических соображений приемлемым оказывается второй путь – управление потоками информации в сети с учетом важности передаваемых данных [119].

В исследованиях, описанных в работах [119] и [33], рассматривается проблема управления информационными потоками, которое сводится к ограничению трафика. В работе [33] рассматриваются только общие правила и подходы к управлению трафиком для предотвращения перегрузок сети, вызванных большими входящими и транзитными нагрузками, когда управление потоками осуществляется без учета приоритетов передаваемой информации.

В работах [184, 255, 378], посвященных проблемам агрегации трафика и оптимизации сетевой нагрузки, решаются разнообразные задачи сетевого управления. Однако при этом не учитываются требования к качеству передачи информационных потоков, выдвигаемые распределенными приложениями, обеспечивающими поддержку выполнения бизнес-процессов.

В работе [118] используются механизмы QoS для предоставления приложениям требуемой полосы пропускания, но также не учитываются значимости приложений. В работе [119] управление базируется на системе правил ограничения объемов передаваемой информации. Однако при этом не рассматриваются вопросы построения адаптивного регулятора для использования в ПУИП корпоративной СУИ.

На сегодня неизвестны работы, посвященные разработке адаптивного регулятора ПУИП, учитывающего загруженность каналов связи и задержку во времени, обусловленную сбором данных, необходимых для расчета управляющего воздействия, а также передачей и выполнением команд управления. Поэтому представляется целесообразным разработать адаптивный регулятор ПУИП, как составную частью СУИ, получающую информацию о загруженности сети и вы-

рабатывающую управляющие воздействия с учетом задержек на передачу контрольной информации и управляющих воздействий, а также задержек на определение и реализацию сигналов управления.

9.4.2. Разработка математической модели адаптивного регулятора

По существу речь идет о реализации в ПУИП централизованного управления информационными потоками корпоративной ИУС, позволяющего оптимизировать загруженность каналов связи, предоставляя требуемое качество передачи высокоприоритетным потокам данных за счет ограничения потоков данных, генерируемых низкоприоритетными приложениями. Для реализации такого управления необходимо разработать математическую модель адаптивного регулятора, создать алгоритмы управления, рассмотреть возможность применения существующих протоколов управления сетевыми элементами и решить ряд других задач.

С точки зрения ТАУ сеть передачи данных можно описать следующим образом. Во-первых, объектом регулирования (ОР) выступают РОУ (рабочие станции, серверы, кластеры и другие элементы ИТ-инфраструктуры), генерирующие потоки информации с интенсивностью $\lambda(t)$. Во-вторых, задающее воздействие или входная величина соответствует пропускной способности тракта передачи данных и в реальных ПУИТ устанавливается на уровне 95–98% от пропускной способности канала связи. В-третьих, выходной величиной является интенсивность информационного потока $\lambda_P(t)$ на выходе ОР после реализации управляющего воздействия. В-четвертых, ошибкой регулирования является величина, соответствующая перегрузкам в канале связи.

Необходимо синтезировать регулятор, который на основе анализа выходной величины определяет управляющее воздействие, минимизирующее ошибку регулирования. Причем управляющее воздействие не вырабатывается, когда генерируемый трафик $\lambda(t)$ не превышает задающее воздействие. В противном случае проводится регулирование и разрабатываемую ПУИП можно рассматривать как систему стабилизации с принципом управления по отклонению [345].

Для реализации управления с минимальной ошибкой и малым временем регулирования необходимо рассчитать регулятор, что, в свою очередь, предполагает построение математической модели ОР. Потоки информации в ИУС имеют стохастическую природу, поэтому ОР не является постоянной структурой, которая может быть описана системой дифференциальных уравнений.

В тоже время управление осуществляется на интервале управления T_y, значение которого выбирается таким образом, чтобы выполнялось следующее условие:

$$T_y \geq \sum (T_{изм} + T_{п1} + T_{ув} + T_{п2} + T_p), \tag{9.25}$$

где $T_{изм}$ – время, необходимое на измерение загруженности сети; $T_{п1}$ – время доставки результатов измерения в ПУИП; $T_{ув}$ – затраты времени на определение управляющего воздействия в ПУИП, включая время работы регулятора; $T_{п2}$ – время передачи служебной информации о характере и величине управляющего воздействия на ОР; T_p – время реализации управляющего воздействия в ОР.

Если значение T_y меньше времени $T_{вар}$ – интервала между фиксируемыми изменениями характеристик трафика, иными словами выполняется условие

$$T_y \leq T_{вар}, \tag{9.26}$$

то при определении передаточных функций регулятора и управляющих воздействий можно исключить параметр времени.

При выполнении условия (9.26) интенсивность генерируемого трафика считается неизменной на интервале T_y. Такое утверждение справедливо для трафика потокового характера, передачи файлов больших размеров, трафика аудио- и видеоконференций и множества других типов трафика. В таких случаях график зависимости величины потока от времени можно представить в виде скачков случайной амплитуды и длительности, причем длительность скачка будет больше T_y. Выражение (9.26) часто будет выполняться и при малом значении интервала T_y.

Очевидно, что для постоянно изменяющегося ОР не существует линейного уравнения, которое бы должным образом описывало зависимость изменения величины трафика от времени и управляющего воздействия. Таким образом, априорной информации об объекте управления оказывается недостаточно для построения ПУИП с желаемыми показателями качества. Изменения в процессе работы значений параметров ОР вызывают необходимость соответствующих изменений значений параметров и структуры регулятора. Поэтому регулятор ПУИП в процессе функционирования должен реагировать на изменение свойств ОР путем изменения значений своих параметров или структуры таким образом, чтобы достигать наилучшего качества работы. Регуляторы такого типа являются адаптивными [345].

С учетом (9.26) модель трафика можно рассматривать как ОР с переменными параметрами, когда на промежутках времени $T_{вар}$, соответствующих отдельным скачкам, величина трафика не изменяется во времени. Тогда передаточную функцию ОР $W_{OP}(z)$ на этих промежутках времени можно представить пропорциональным звеном с коэффициентом $K_П$, равным величине потока:

$$W_{OP}(z) = K_П. \tag{9.27}$$

При изменении интенсивности потока происходит соответствующее изменение значения параметра ОР. Для передаточной функции ОР типа (9.27) не требуется изменять структуру регулятора. Изменять можно только значения параметров настройки. В этом случае регулятор можно рассматривать как аналитическую систему с самонастройкой по характеристикам объекта управления. Изменение значений параметров регулятора в соответствии с изменением параметра объекта управления проводится устройством настройки параметров регулятора (УНПР).

Основными этапами проектирования адаптивного регулятора являются синтез его структуры и разработка алгоритма адаптации.

Передаточная функция $W_З(z)$ замкнутой системы имеет вид (9.12), а регулятора $W_P(z)$ – (9.13).

Выбираем передаточную функцию замкнутой системы в виде апериодического звена первого порядка, которое обеспечивает нулевую статическую ошибку и малое время регулирования:

$$W_З(s) = \frac{1}{s+1} \underset{T_0=0,05}{\overset{z-преобразование}{\longrightarrow}} W_З(z) = \frac{0,09516}{z-0,9048}, \tag{9.28}$$

где T_0 – период дискретизации, который выбран равным 0,05 с.

На основе выражений (9.13), (9.27) и (9.28) получаем следующую передаточную функцию адаптивного регулятора:

$$W_P(z) = \frac{0,09516}{K(z-1)}. \tag{9.29}$$

Изменение значения параметра K регулятора проводится в соответствии с изменением передаточной функции ОР.

Алгоритм адаптации в данном случае является простым и заключается в установлении значения параметра K регулятора, равным значению параметра $K_П$ ОР. Реализация алгоритма осуществляется программно посредством УНПР.

Модель подсистемы управления информационными потоками с адаптивным регулятором, построенная в пакете Matlab 7.1, изображена на рис. 9.13.

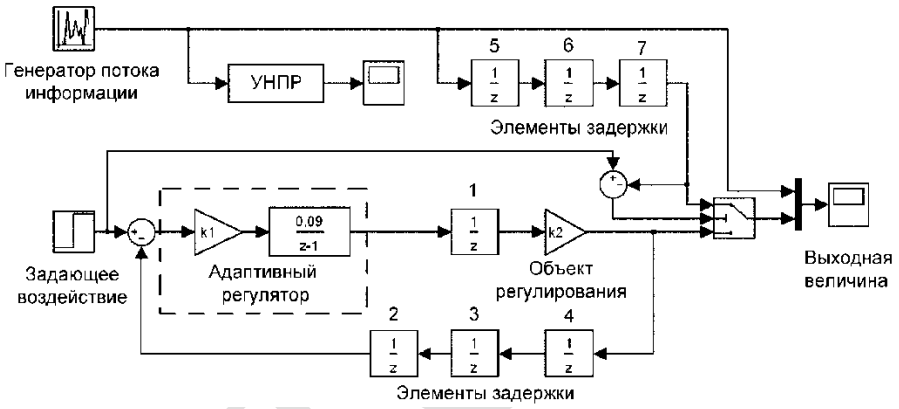


Рис. 9.13. Модель ПУИП с адаптивным регулятором

Для включения ОР в цепь регулирования в качестве пропорционального звена, которое изменяет значение параметра $K_П$ при изменении объема генерируемого трафика, на схеме модели рис. 9.13 задействовано два блока, каждому из которых соответствуют различные аспекты работы ОР. Этими блоками являются генератор потока информации и объект регулирования. В реальных ИУС этим блокам соответствует один элемент, являющийся источником потока данных. Генератор потока информации создает трафик с заданными статистическими свойствами, однако в силу специфики моделирования амплитуду сигнала этого блока, которым в реальных ИУС соответствует интенсивность трафика, нельзя регулировать. Поэтому в модель введен блок – объект регулирования, который позволяет осуществлять регулятору управление интенсивностью потока. Таким образом, на выходе объекта управления присутствует регулируемый поток информации $\lambda_P(t)$, а на выходе генератора потока информации – нерегулируемый $\lambda(t)$.

Поскольку регулирование осуществляется только в моменты времени, когда в канале присутствуют перегрузки, а в противном случае управляющее воздействие не вырабатывается, то в ПУИП реализовано два режима функционирования модели. В первом случае при допустимой загруженности канала связи контур регулирования с помощью переключателя находится в пассивном состоянии и информационный поток $\lambda(t)$ передается без ограничений. Во втором случае

при появлении перегрузок в канале связи, в действие вводится адаптивный регулятор, реализующий управление потоками данных. Настройка параметра регулятора осуществляется посредством УНПР.

Элементы задержки 2–4 позволяют при моделировании учесть интервал времени, соответствующий интервалу $T_{изм} + T_{п1} + T_{ув} + T_{п2}$, а элемент задержки 1 – интервал T_p. Элементы задержки 5–7 учитывают время прохождения потоков данных по сети.

9.4.3. Результаты моделирования

При тестировании роботы адаптивного регулятора ПУИП рассмотрены три типа трафика.

Первый тип трафика приведен на рис. 9.14 пунктирной линией и представляет собой стохастический трафик $\lambda(t)$ со ступенчатым изменением интенсивности потока.

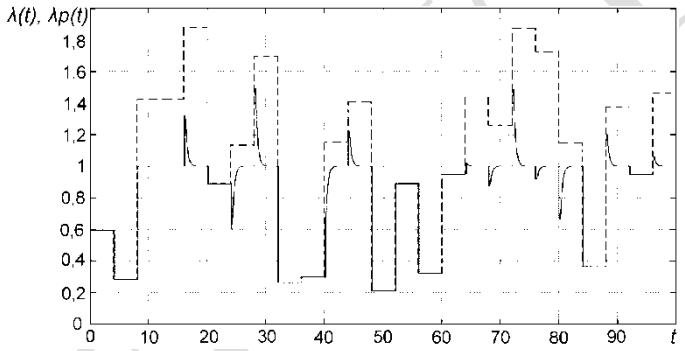


Рис. 9.14. Загруженность канала связи $\lambda_P(t)$ с адаптивным регулированием потоков информации (сплошная линия) и без регулирования $\lambda(t)$ (пунктирная линия)

Второй тип трафика приведен на рис. 9.15 и соответствует трафику, генерируемому, например, IRC-клиентами. В этом случае происходит резкое и частое изменение величины потока. На рис. 9.16 приведен график загруженности канала связи $\lambda_P(t)$ при использовании адаптивного регулятора.

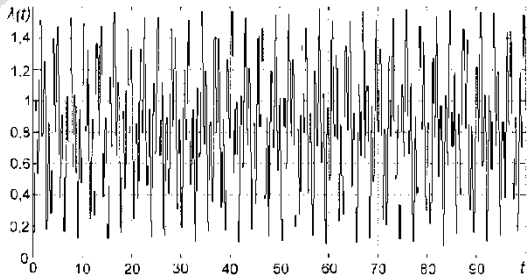


Рис. 9.15. Загруженность канала связи $\lambda(t)$ без регулирования

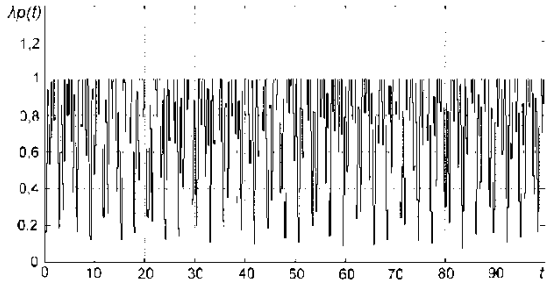


Рис. 9.16. Загруженность канала $\lambda_P(t)$
при использовании адаптивного регулятора

Следующий тип трафика соответствует реальному email трафику, генерируемому пакетом OPNET на основе параметров, приведенных на рис. 9.17.

Attribute	Value
Send Interarrival Time (seconds)	exponential (360)
Send Group Size	constant (3)
Receive Interarrival Time (seconds)	exponential (360)
Receive Group Size	constant (3)
E-Mail Size (bytes)	constant (2000)
Symbolic Server Name	Email Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Рис. 9.17. Параметры email трафика в пакете OPNET

Результат генерации трафика, соответствующий экспериментальным исследованиям обмена email трафиком между сервером и клиентской станцией в реальных сетях, полученный специалистами OPNET Technologies Inc., приведен на рис. 9.18. График загруженности канала связи после адаптивного регулирования изображен на рис. 9.19.

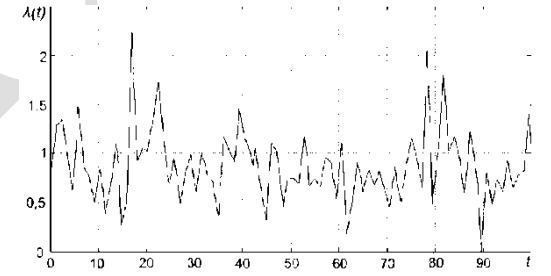


Рис. 9.18. График загруженности канала связи email
трафиком без управления

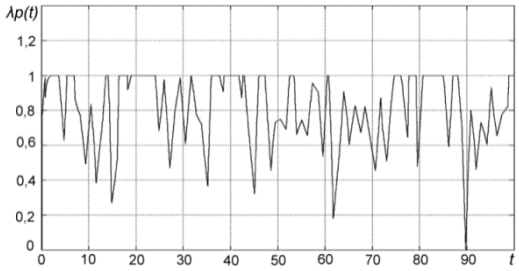


Рис. 9.19. График загруженности канала email трафиком при адаптивном управлении

На основании результатов проведенных исследований можно сделать вывод о том, что синтезированный методами ТАУ адаптивный регулятор при использовании его в ПУИП позволяет управлять интенсивностью входящих в сеть информационных потоков, исключая перегрузки каналов связи. Регулятор ограничивает запредельные информационные потоки при отсутствии априорной информации об объекте управления, при ошибке регулирования не более 10^{-3}. Значения параметров регулятора изменяются в режиме реального времени в зависимости от значений параметров объекта управления.

ВЫВОДЫ К ГЛАВЕ 9

1. Анализ особенностей управления сетевым трафиком в корпоративной ИТ-инфраструктуре показал, что повышение качества предоставления ИТ-сервисов может быть осуществлено повышением скорости прохождения транзакций пользователя. Это может быть реализовано выделением дополнительных коммуникационных ресурсов и приоритетной передачей сообщений, относящихся к критически важным ИТ-сервисам.

2. Разработан метод управления информационными потоками в корпоративной информационно-телекоммуникационной системе. В отличие от существующих предлагаемый метод учитывает важность приложений, генерирующих информационные потоки, и в случае перегрузки каналов связи ограничивает интенсивность входящих потоков низкоприоритетных приложений, выделяя дополнительную полосу высокоприоритетным приложениям. Применение предлагаемого метода в ИТ-инфраструктуре корпоративного уровня позволяет повысить уровень услуг, предоставляемых высокоприоритетными приложениями.

3. Разработаны и исследованы модели для трех методов управления сетевым трафиком, не допускающих перегрузок каналов связи. Результаты проведенных исследований показывают, что синтезированный методами ТАУ адаптивный регулятор при использовании его в ПУИП позволяет управлять интенсивностью входящих в сеть информационных потоков, исключая перегрузки каналов связи. Регулятор ограничивает информационные потоки при отсутствии априорной информации об объекте управления. Значения параметров регулятора изменяются в режиме реального времени в зависимости от значений параметров объекта управления.

ГЛАВА 10

МЕТОДЫ И АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ УПРАВЛЕНИЯ РАСПРЕДЕЛЕНИЕМ РЕСУРСОВ И НАГРУЗКОЙ ИТ-ИНФРАСТРУКТУРЫ

Глава посвящена алгоритмам решения задач управления нагрузкой и распределения ресурсов ИТ-инфраструктуры. Предложены и исследованы алгоритмы разных классов – точные эвристические, варианты генетических алгоритмов. Определены условия эффективного применения предложенных алгоритмов.

10.1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ВИРТУАЛЬНЫХ МАШИН ПО СЕРВЕРАМ

Глобальная информатизация, вызванная стремительным развитием ИТ, обусловила повышение требований к количественным и качественным параметрам информационного обслуживания бизнеса [81]. Реакцией на эти процессы стало распространение ИУС, предназначенных для поддержки процессов принятия управленческих решений министерств, ведомств и крупных корпораций. Создание высокопроизводительной, надежной ИТ-инфраструктуры ИУС реализуется на основе ЦОД.

В области ИТ накоплен определенный опыт создания и эксплуатации ЦОД, который подтвердил, что эксплуатация ЦОД возможна только при наличии организационной структуры, оснащенной современным инструментарием и комплексом методик сбора, анализа информации, принятия решений и управления их отработкой. Их создание требует глубокого понимания функционирования ИТ-инфраструктуры, четкой постановки проблем исследования, разработки математических моделей и соответствующих методов решения задач и, наконец, реализации упомянутых инструментария и методик в структуре СУИ. Среди проблем, которые еще требуют решения, остается важная проблема распределения и управления нагрузкой и ресурсами ЦОД.

Математическую модель распределения ВМ по серверам можно представить следующим образом.

Пусть ЦОД содержит упорядоченное множество физических серверов

$$N = \{N_i\},\ i = \overline{1,n}, \tag{10.1}$$

где n – количество серверов.

Каждый сервер $N_i,\ i = \overline{1,n}$, характеризуется двумя параметрами, определяющими его вычислительные возможности: Ω_i – мощность процессора и Γ_i – объем ОЗУ.

Подлежит распределению упорядоченное множество ВМ:

$$K = \{K_j\},\ j = \overline{1,m}, \tag{10.2}$$

где m – количество ВМ.

На каждой ВМ разворачивается одно или несколько приложений.

Каждая ВМ имеет потребности в процессорном времени – ω_j и в оперативной памяти – γ_j.

Вводится вектор технологической и организационной возможности развертывания каждой ВМ (приложения) из множества K в текущей системе:

$$C = \{C_1, ..., C_m\}, \tag{10.3}$$

где

$$C_j = \begin{cases} 1, \text{ если СУИ может разворачивать или сворачивать ВМ } K_j; \\ 0, \text{ в противном случае.} \end{cases} \tag{10.4}$$

Вводится матрица R размером $m \times n$ совместимости сервера из множества N и ВМ из множества K, по которой будет определяться возможность развертывания конкретной ВМ на конкретном сервере на текущем этапе распределения

$$R = \left| r_{ji} \right|, \tag{10.5}$$

где

$$r_{ji} = \begin{cases} 1, \text{ если ВМ } K_j \text{ можно развернуть на сервере } N_i; \\ 0, \text{ в противном случае.} \end{cases} \tag{10.6}$$

Результаты распределения представляются в виде матрицы:

$$X = \begin{vmatrix} x_{11} & \dots & x_{1m} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nm} \end{vmatrix}, \tag{10.7}$$

где

$$x_{ji} = \begin{cases} 1, \text{ если ВМ } K_j \text{ развернута на сервере } N_i; \\ 0, \text{ в противном случае.} \end{cases} \tag{10.8}$$

Матрица X является решением задачи и определяет распределение множества K ВМ на множестве N физических серверов.

Постановка задачи. Если все серверы множества N имеют идентичные технические характеристики и, следовательно, одинаковые вычислительные ресурсы, то можно считать, что $\Omega_i = 1$ и $\Gamma_i = 1$, $i = \overline{1,n}$, т. е.

$$\{\Omega_i, \Gamma_i\} \big| N_i = \{1,1\}, i = \overline{1,n}. \tag{10.9}$$

Здесь сделан переход от измерения ресурсных возможностей серверов в абсолютных единицах, когда память измеряется в Гб, а частота процессора – в ГГц, к относительным. Тогда потребности в ресурсах ВМ из множества K определяются

АУТС КП

как часть от ресурсных возможностей сервера, нормированных относительно максимально возможной величины, соответствующей (10.9).

Если задача кластеризации не рассматривается, то требования каждой ВМ из множества *K* к ресурсам не могут превышать возможностей сервера:

$$\omega_j \le 1 \text{ и } \gamma_j \le 1, \quad j = \overline{1,m}. \tag{10.10}$$

При решении задачи распределения ВМ для всех серверов множества *N* должны выполняться следующее ресурсные ограничения:

$$\sum_{j=1}^{m} x_{ij}\gamma_j \le 1,\ i = \overline{1,n}. \tag{10.11}$$

Ограничение (10.11) требует, чтобы емкость ОЗУ сервера N_i была достаточной для размещения запланированных для размещения на нем ВМ

$$\sum_{j=1}^{m} x_{ij}\omega_j \le 1,\ i = \overline{1,n}. \tag{10.12}$$

Ограничение (10.12) требует, чтобы потребности всех ВМ K_j в емкости процессоров удовлетворялись в совокупности соответствующим параметрам физического сервера, на которых эти ВМ будут располагаться.

Введем вектор $\vec{y} = \{y_i\}$, $i = \overline{1,n}$, причем $y_i = 1$, если на сервере N_i располагается хотя бы одна ВМ, и $y_i = 0$ в противном случае. Тогда критерием оптимальности при решении задачи размещения ВМ на физических серверах при определенных условиях есть

$$\min \sum_{i=1}^{n} y_i, \tag{10.13}$$

т. е. серверы должны заполняться ВМ таким образом, чтобы было задействовано минимальное количество серверов. Такой подход позволяет минимизировать суммарные затраты *S* на обслуживание и энергоснабжение всего парка серверов ЦОД.

В общем случае целевую функцию можно представить следующим образом:

$$S = \sum_{i=1}^{n} s_i y_i, \tag{10.14}$$

где s_i – затраты на обслуживание и энергоснабжение *i*-го сервера.

Для серверов с идентичными характеристиками выражение (10.14) примет вид

$$S = s\sum_{i=1}^{n} y_i, \tag{10.15}$$

где *s* – стоимость обслуживания и затраты на энергоснабжение для одного сервера.

АУТСКИ

Тогда возникает задача размещения ВМ на серверах таким образом, чтобы выражение (10.14) или (10.15) достигало минимума с учетом ограничений (10.10)–(10.12). Иными словами, для повышения эффективности использования вычислительных ресурсов ЦОД необходимо максимально плотно размещать ВМ на серверах.

10.2. МЕТОДЫ РЕШЕНИЯ ЗАДАЧ УПРАВЛЕНИЯ НАГРУЗКОЙ И РАСПРЕДЕЛЕНИЕМ ОГРАНИЧЕННЫХ РЕСУРСОВ

Актуальность проблемы управления нагрузкой и ресурсами ИТ-инфраструктуры и, в частности, ЦОД вызывает не только необходимость разработки комплекса математических моделей управления распределением ресурсов, но и определения или разработки методов решения проблемы управления нагрузкой и ресурсами для различных вариантов организации и технологий реализации серверного парка. Необходимо не только определить общую модель планирования, управления и диспетчеризации ресурсов и нагрузки, но и определить требования к характеристикам применяемых для решения задач методов.

Комплекс моделей управления нагрузкой и ресурсами ЦОД для различных технологий хостинга предложен в цикле работ [148, 146, 158, 159, 160, 228, 363, 364, 365, 366, 367, 368].

Комплексная проблема управления нагрузкой и ресурсами ЦОД обычно решается путем реализации последовательно (при необходимости, итеративно) выполняемых этапов планирования, управления и диспетчеризации ресурсов и нагрузки, модели которых в общем сводятся к оптимизационным проблемам вида

$$\max f(x), \tag{10.16}$$

$$x \in \Omega, \tag{10.17}$$

где $f(x):\Omega \rightarrow R$ – функция, которая определяет эффективность решения x, Ω – область допустимых решений.

Без потери обобщенности результатов можно считать, что $f(x) \geq 0$, а Ω – дискретное или непрерывное пространство.

Функция $f(x)$ и область Ω зависят от технологий хостинга, построения прикладного ПО, кластеризации и виртуализации, а также принятых критериев эффективности и ресурсных, технологических и других актуальных ограничений. Чаще всего модели планирования, управления и диспетчеризации ресурсов и нагрузки ЦОД имеют вид линейных и нелинейных вариантов задач линейного, булевого или смешанного программирования.

На данный момент существует ряд подходов к решению задач и готовых решений по планированию, управлению распределением ресурсов и нагрузкой, описанных в работах [41, 74, 136, 174, 173], большинство из которых для поиска решений задачи многокритериального поиска оптимума используют:

- методы математического программирования (алгоритмы линейного, булевого или смешанного программирования);
- эвристические методы (эвристические алгоритмы с учетом специфики критериев и ограничений);
- методы эволюционного поиска (варианты генетического алгоритма).

Методы линейного и нелинейного программирования, как правило, гарантируют нахождение оптимального решения. Однако, высокая вычислительная сложность, экспоненциально возрастающая с увеличением размерности задачи многокритериального поиска, делает эти методы не совсем подходящими для решения задач управления распределением ресурсов ИТ-инфраструктуры, которые имеют большую размерность и жесткие временные ограничения.

Поскольку обойти методы перебора для решения большинства задач планирования, управления и диспетчеризации ресурсов и нагрузки, которые составляют NP-полные проблемы, нельзя, оправдано использование эвристических методов или генетического алгоритма (ГА), позволяющих получить контролируемое приближение к точному результату.

Эвристические методы значительно сокращают временные затраты на решение практически любой задачи многокритериального поиска. Обычно с помощью этих методов находят решение, близкое к оптимальному. Несомненным преимуществом таких методов является возможность учета дополнительного набора технологических ограничений, однако это сужает область использования подобных алгоритмов. К тому же, эти методы не защищены от попадания в локальный оптимум.

При оперативном управлении ИТ-инфраструктурой время работы алгоритмов ограничено и работу алгоритмов с итерационными процедурами останавливают по истечении допустимого времени [73].

Поэтому методы искусственного интеллекта становятся актуальными для решения задач распределения и управления нагрузкой и ресурсами ЦОД. Особенно перспективными представляются ГА, которые относятся к адаптивным методам поиска экстремума в проблемах функциональной оптимизации, условной или безусловной [169].

Если множество возможных решений проблемы закодировано, то основанный на принципах моделирования биологических процессов ГА способен шаг за шагом улучшать исходный набор решений (начальную популяцию) до получения приемлемого для исследователя результата (популяции с соответствующими свойствами). Проблемы проектирования, анализа и управления ИУС очень просто кодировать, поскольку они чаще всего относятся к проблемам условной оптимизации, сформулированных в виде задач математического программирования с булевыми переменными. В случаях непрерывных или целочисленных переменных или смешанных задач математического программирования также существуют эффективные схемы кодирования решений.

В большинстве работ, рассматривающих задачи управления ресурсами и нагрузками ЦОД при виртуальном хостинге (см., например, [364, 367, 368]), поиск лучших вариантов распределения ресурсов между пользователями происходит с помощью эвристик, организации рекурсивного поиска лучшего варианта и иногда матричных методов.

В работах [141, 143, 228] для решения задач подобного типа предлагается использовать генетические алгоритмы различных модификаций. Использование алгоритмов генетического типа мотивируется возможностью нахождения глобальных оптимумов, в то время как обычные эвристики могут приводить к неплохим, но локальным оптимумам, природа возникновения которых зависит от формулировки критерия поиска в задаче и реализации самой эвристики.

Для решения поставленных задач на основе сформулированных математических моделей в разделе избраны:

• эвристические алгоритмы;
• генетические алгоритмы (ГА).

Поэтому для заданного класса задач целесообразно сравнить результаты применения ГА с результатами эвристических алгоритмов для определения конкурентоспособности использования ГА.

10.3. ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ ИТ-ИНФРАСТРУКТУРЫ

Стоит задача создания специализированных эвристических алгоритмов распределения ресурсов с учетом сложившейся в ИТ-инфраструктуре ситуации, технологических, ресурсных и других ограничений.

Ниже приведены три эвристических алгоритма распределения ВМ по серверам в условиях избытка ресурсов:

• эвристический алгоритм последовательного заполнения серверов;
• эвристический алгоритм типа «Т»;
• эвристический алгоритм типа «М».

Алгоритм последовательного заполнения является простейшим из всех рассматриваемых алгоритмов. Он часто используется, поскольку базируется на накопленном опыте и позволяет получить рациональное, но очень редко оптимальное решение для ЦОД.

Алгоритм последовательного заполнения.

Шаг 1. Выбрать первую ВМ из упорядоченного множества ВМ *K*.

Шаг 2. Выбрать из упорядоченного множества серверов (10.1) первый сервер.

Шаг 3. Если для выбранных сервера и ВМ не выполняется условие (10.12), то удалить сервер из упорядоченного множества *N* и перейти на шаг 2.

Шаг 4. Добавить ВМ на сервер и исключить из списка ВМ. Если список ВМ пуст, то закончить выполнение алгоритма, иначе – перейти на шаг 1.

Эвристические алгоритмы типов «Т» и «М». Кратко рассмотрим их сущность.

Предварительно упорядоченное множество ВМ *K* разбивается на два подмножества – упорядоченное подмножество ВМ K_{ram}, включающее в себя ВМ, для которых выполняется условие $\gamma_j \geq \omega_j$, $j = \overline{1,m}$, и упорядоченное подмножество ВМ K_p, включающее ВМ, у которых $\gamma_j < \omega_j$, $j = \overline{1,m}$.

Очевидно, что распределение ВМ по серверам путем поочередного выбора ВМ из подмножеств K_{ram} и K_p позволяет более компактно загружать серверы из множества *K*, чем это делает алгоритм последовательного заполнения. Именно по такому принципу работает эвристический алгоритм «Т».

Эвристический алгоритм «Т».

Шаг 1. Упорядоченные подмножества (списки) K_{ram} и K_p сортируются в порядке убывания запрашиваемых величин ОЗУ для K_{ram} и процессорного времени для K_p.

Шаг 2. Из списка *N* выбирается первый сервер.

Шаг 3. Задается $k = 1$, $h = 1$.

Шаг 4. Из списка K_{ram} выбирается элемент K_k. Если список K_{ram} пуст или достигнут конец списка, то осуществляется переход на шаг 6.

Шаг 5. Если для выбранного сервера не выполняется условие (10.12), то $k = k + 1$ и осуществляется переход к шагу 4. В противном случае ВМ K_k добавляется на сервер и удаляется из списка K_{ram}.

Шаг 6. Из списка K_p выбирается элемент K_h. Если список K_p пуст или достигнут конец списка, то осуществляется переход на шаг 8.

Шаг 7. Если для выбранного сервера не выполняется условие (10.12), то $h = h + 1$ и осуществляется переход к шагу 6. В противном случае ВМ K_h добавляется на текущий сервер и удаляется из списка K_p. Переход на шаг 3.

Шаг 8. Если списки K_{ram} и K_k пусты, то выполнение алгоритма завершается.

Шаг 9. Если достигнут конец списков K_{ram} и K_k, то текущий сервер удаляется из множества N и осуществляется переход на шаг 2. В противном случае осуществляется переход на шаг 3.

Эвристический алгоритм «М».

Алгоритм «М» предназначен для распределения ВМ как с пропорциональными, так и с непропорциональными требованиями к ресурсам.

Учитывая, что потребности в памяти и процессорном времени выражены в относительных единицах, для оценки обобщенной потребности j-й ВМ, $j = \overline{1,m}$ в вычислительных ресурсах введем коэффициент q_j, который определяется следующим образом:

$$q_j = \sqrt{\omega_j^2 + \gamma_j^2},\ j = \overline{1,m}. \tag{10.18}$$

Аналогичным образом введем коэффициент, определяющий обобщенную загруженность i-го сервера $i = \overline{1,n}$ и зависящий от количества задействованных ресурсов:

$$Q_i = \sqrt{\alpha_i^2 + \beta_i^2}, i = \overline{1,n}, \tag{10.19}$$

где α_i и β_i – доля задействованной оперативной памяти и процессорного времени i-го сервера соответственно.

Алгоритм «М».

Шаг 1. Сортировать список серверов в порядке убывания значений коэффициента задействованных ресурсов Q_i, $i = \overline{1,n}$.

Шаг 2. Сортировать список ВМ в порядке убывания обобщенной потребности в ресурсах q_i для всех $j = \overline{1,m}$.

Шаг 3. Выбрать ВМ с наибольшим значением коэффициента q_j для всех $j = \overline{1,m}$.

Шаг 4. Выбрать сервер с наибольшим значением коэффициента Q_i для всех $i = \overline{1,n}$.

Шаг 5. Если для выбранных сервера и ВМ выполняется условие (10.12), то ВМ добавляется на сервер, удаляется из списка ВМ, а для данного сервера проводится перерасчет значения коэффициента Q_i, после чего осуществляется переход на шаг 7. Если условие (10.12) не выполняется, то осуществляется переход на следующий шаг.

АУТСКИ

Шаг 6. Выбрать следующий сервер из списка серверов и перейти на шаг 5.

Шаг 7. Если список ВМ пуст, то конец алгоритма, иначе осуществляется переход на шаг 1.

Вычислительная сложность алгоритма составляет $m \times n$.

Результаты работы эвристических алгоритмов и их сравнение с ГА будут приведены ниже.

10.4. РЕШЕНИЕ ЗАДАЧ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ НАГРУЗОК И РЕСУРСОВ С ИСПОЛЬЗОВАНИЕМ КЛАССИЧЕСКОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА

10.4.1. Применение классического генетического алгоритма

Сравнительно новыми методами поиска решений задач многокритериального поиска оптимума являются методы эволюционного поиска, к которым относятся генетические алгоритмы и методы эволюционного программирования.

Генетический алгоритм представляет собой комбинацию переборного и градиентного методов [64, 129, 169]. Механизмы кроссовера и мутации реализуют переборную часть, а отбор лучших решений – градиентный спуск.

Если на некотором множестве задана функция от нескольких переменных, то ГА находит точку, где значение функции расположено достаточно близко к максимально возможному значению. Ограничивая время работы алгоритма, получают одно из лучших решений, которые можно найти за это время.

Базовая схема классического алгоритма (КГА) [317] приведена на рис. 10.1.

Рис. 10.1. Базовая схема классического ГА

Генетические методы сочетают преимущество эвристических методов (быстрый поиск решения, близкого к оптимальному) со способами выхода из локальных экстремумов.

В КГА для задачи управления нагрузкой и ресурсами начальная популяция создается случайным образом. Размер популяции (количество хромосом) N является фиксированным и не изменяется в течение всего алгоритма. Каждая особь генерируется как L – битная строка, где L – длина кода особи. Каждая особь – решение текущей задачи. Более пригодная особь – это пригодное решение.

Алгоритм состоит из трех стадий:

- создание промежуточной популяции отбором из числа представителей текущего поколения;

• создание промежуточного поколения путем кроссовера с участием отобранных представителей, что приводит к формированию популяции новой эпохи;

• мутация полученных потомков с определенной заранее заданной достоверностью.

При решении конкретной задачи эффективность ГА зависит от генетических операторов и выбора значений параметров, а также способа представления задачи. Оптимизация этих факторов приводит к повышению скорости и устойчивости поиска [20, 169, 193, 245, 246, 317, 319].

Для КГА характерны следующие формулировки [169, 245, 246, 254]:

• задается функция оптимальности (целевая функция), которая определяет эффективность каждого найденного решения;

• для заданных ограничений инициализируется начальная популяция потенциальных решений, которая кодируется как вектор;

• каждой хромосоме присваивается значение эффективности в соответствии с целевой функцией;

• каждой хромосоме присваивается вероятность воспроизведения, зависящая от ее эффективности;

• создается новая популяция хромосом, причем с большей вероятностью воспроизводятся самые эффективные элементы. На основе хромосом старой популяции путем применения операций кроссовера и мутации создаются потомки;

• процесс останавливается, когда получено удовлетворительное решение или истекло отведенное время.

Результатом поиска оптимального распределения ВМ по серверам ЦОД является двумерный массив размерностью $m \times n$, элементы которого принимают значения 0 или 1 соответственно для случаев, когда ВМ не размещается или размещается на определенном сервере.

10.4.2. Разработка базового варианта генетического алгоритма решения задачи управления распределением ресурсов

Генетический алгоритм поиска оптимального распределения ВМ по серверам ЦОД имеет особенности, учет которых целесообразен при оптимизации его структуры и работы.

Важными моментами при создании КГА является выбор структуры хромосомы и определение характера основных операторов КГА: кроссовера, пополнения популяции, мутации и механизмов селекции.

Первый этап. Выбирается структура хромосомы. Учитывая, что каждая хромосома является одним из решений задачи, в качестве хромосомы в генетическом алгоритме целесообразно выбрать матрицу X размещения ВМ по серверам (10.7), элемент x_{ji} которой определяется из выражения (10.8) для модели без кластеризации.

В качестве гена – составной части хромосомы, которая остается неделимой при генетических операциях, для модели без кластеризации с учетом ограничений (10.10)–(10.12) целесообразно выбрать значение колонки в матрице X. Тогда значением гена в КГА станет значение распределения конкретной ВМ на серверах серверного парка хостингового ЦОД.

Согласно принципам реализации кроссовер представляет собой обмен участками между двумя хромосомами. Будем использовать одноточечный вариант

кроссовера между хромосомами. Поскольку ген является неделимой частью при генетических операциях, то операция кроссовера сводится к обмену двух матриц X^n и X^m участками матрицы, начиная с определенной колонки и до конца.

Вариант мутации также связан с выбранными моделями гена и хромосомы. В основном предлагаются мутации инверсного плана [64, 169], которые в условиях рассматриваемых задач не пригодны для использования из-за наличия описанных ранее ограничений.

Второй этап. Определяется характер основных операций.

Процесс мутации для модели без кластеризации заключается в переносе ВМ с одного сервера на другой. Для матричного представления (10.7) это будет выглядеть как сдвиг в столбце матрицы на произвольную позицию вверх или вниз.

Процесс мутации для моделей с кластеризацией заключается в перераспределении коэффициентов занятости между серверами для конкретных ВМ. В матричном виде (10.7) это примет вид произвольного изменения коэффициентов столбца, но таким образом, чтобы при перерасчете суммы долей процессорных емкостей серверов в итоге получалось значение, равное потребностям в процессорной емкости текущей ВМ.

Определив состав и характер основных операций КГА поиска оптимального размещения и выбрав генно-хромосомную модель, перейдем к формализации самого алгоритма, как последовательности определенных шагов.

Схематическое изображение предлагаемого базового ГА приведено на рис. 10.2.

Предлагаемый КГА реализует следующую последовательность процессов:

Шаг 1. Генерация стартовой популяции численностью *N*.

При генерации каждая новая особь (план распределения ресурсов) проверяется на жизнеспособность по наборами критериев: для модели без кластеризации по ограничениям (10.10)–(10.12). Такая селекция уже на этапе создания стартовой популяции необходима для генерации здоровых популяций в течение последующих эпох системы.

Шаг 2. Вычисление значения целевой функции (ЦФ) для каждой из особей текущей популяции. ЦФ для каждой из отдельных задач – это критерий оптимальности.

Шаг 3. Вычисление среднего значения уровня приспособленности популяции в целом – среднего значения ЦФ.

Шаг 4. Проверка на удовлетворение полученным результатом по значению средней и лучшей ЦФ текущей популяции.

Если результат удовлетворяет требованиям – вывод результата и конец выполнения алгоритма.

Если нет – переход на шаг 5.

Шаг 5. Селекция особей из текущей популяции для получения на их основе генотипа новых особей для новой эпохи.

Отбор происходит на основе значения ЦФ (критерия приспособленности) каждой конкретной особи популяции. Особь переходит в следующую эпоху, если значение ее критерия приспособленности больше, чем среднее значение приспособленности (ЦФ) всей популяции. Количество особей старой эпохи, которые переходят в новую, ограничено половиной популяции. Все особи, которые не прошли этап отбора, считаются вымершими и в последующих шагах работы алгоритма не участвуют.

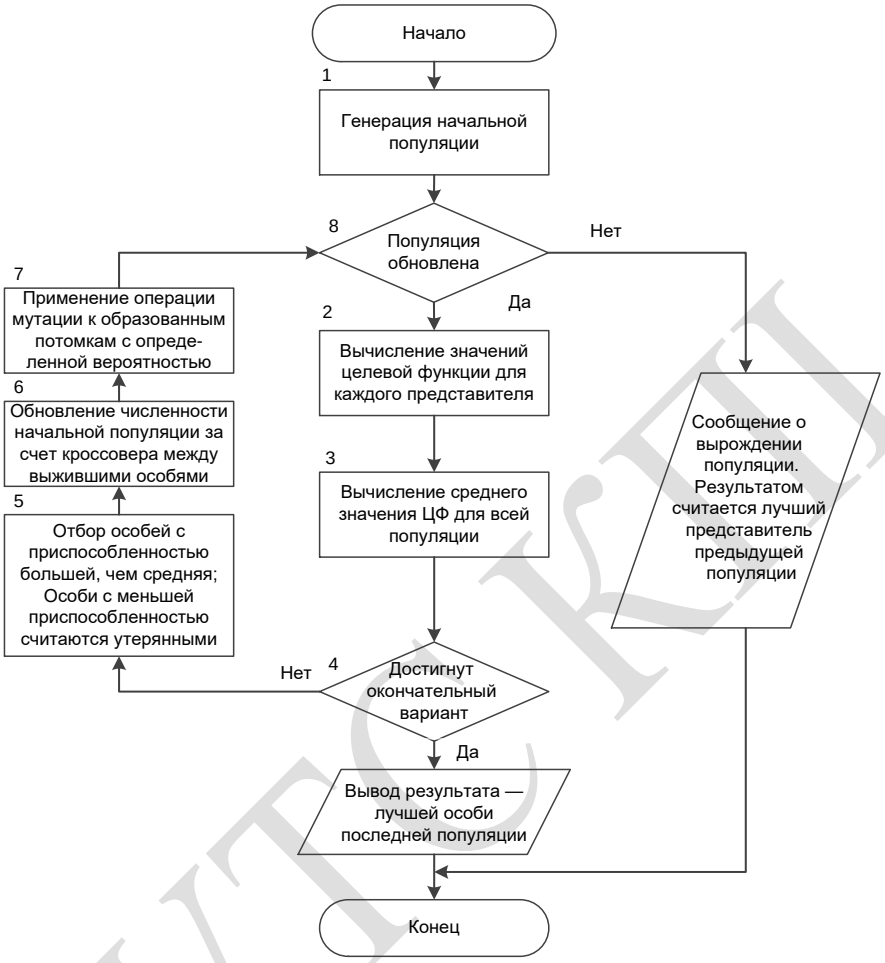


Рис. 10.2. Базовый ГА поиска оптимального распределения ресурсов серверного парка ЦОД между виртуальными машинами

Шаг 6. Путем использования определенного выше оператора кроссовера выполняется восстановление численности популяции. Материалом для кроссовера выбираются лучшие особи ушедшей эпохи.

Во время операции кроссовера получают два новых потомка, каждый из которых проверяется на жизнеспособность по критериям (10.10)–(10.12) для модели без кластеризации.

Процесс кроссовера повторяется до полного восстановления численности стартовой популяции или до переполнения стека попыток. В последнем случае сообщается, что произошло вырождение популяции. Лучшей популяцией при этом признается популяция ушедшей эпохи.

Шаг 7. С определенной вероятностью выполняется определенная выше операция мутации над одним из потомков, полученным в результате выполнения кроссовера. Результат мутации проверяется на жизнеспособность по ограничениям (10.10)–(10.12).

Шаг 8. Выход на новый виток процесса – новую эпоху.

АУТСКИ

Результаты исследования предложенного КГА приведены в следующем параграфе.

Из последовательности правил видно, что операции кроссовера и мутации происходят на каждой итерации КГА. Одновременное использование обоих операторов может привести к потере лучших решений или быстрого вырождения популяции. Целесообразнее было бы разделить стратегии «использования» и «исследования» и на каждой итерации КГА по определенному критерию выбирать необходимый вариант генетического оператора.

Такой принцип положен в основу функционирования предлагаемого управляемого ГА, рассматриваемого ниже.

10.5. УПРАВЛЯЕМЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ В ЗАДАЧАХ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ

10.5.1. Обоснование целесообразности применения генетических алгоритмов для решения задач распределения ресурсов

Высокая эффективность использования вычислительных ресурсов ЦОД достигается за счет применения технологии виртуализации. Большая стоимость инфраструктуры ЦОД, существенные затраты на обслуживание и энергоснабжение обусловливают необходимость разработки методов и алгоритмов эффективного использования вычислительных ресурсов. В настоящее время существенные финансовые потери ЦОД обусловлены нерациональным распределением ВМ по физическим серверам, поскольку при распределении ресурсов зачастую используется неэффективный метод последовательного заполнения или метод экспертных оценок, когда распределение ВМ по физическим серверам администратор выполняет вручную. В то же время, максимально плотное заполнение серверов виртуальными машинами приводит к сокращению затрат на электроэнергию и обслуживание за счет отключения незадействованных серверов, либо позволяет обслужить большее количество заказчиков тем же серверным парком.

В работе [158] показано, что применение КГА для решения задач распределения вычислительных ресурсов ЦОД связано с рядом проблем, обусловленных быстрым вырождением популяции, отсутствием хорошего генетического материала при генерации начальной популяции, неоптимальным распределением ресурсов при наличии относительно малых по требованиям ВМ, которые должны быть размещены на серверах ЦОД.

В [143] предложен вариант управляемого генетического алгоритма (УГА) для решения задач малой размерности. Однако проведенное сравнение УГА только с одним эвристическим алгоритмом не позволяет в полной мере оценить преимущества УГА. Кроме того, рассматривается применение оператора кроссовера только второго порядка, не исследовано влияние вероятности мутации на правильность получаемого результата и не рассмотрены принципы генерации начальной популяции.

В задачах распределения ресурсов часто необходимо использовать целевые функции, являющиеся нелинейными и не унимодальными, ограничения – нелинейные и невыпуклые, при этом переменные могут быть булевыми, целыми, непрерывными либо смешанными. Во многих случаях реализация традиционных стратегий требует огромных объемов вычислений, а «время жизни» полученного

результата зачастую невелико, поскольку ситуация безостановочно изменяется и постоянно требует новых решений. В таких случаях особый интерес представляет применение ГА, имеющих ряд преимуществ, обусловленных их природой, которые могут дать хорошие результаты при решении задач с такими свойствами. В то же время использование ГА в контексте данной проблемы почти не исследовано.

Методы эволюционного поиска в виде ГА представляют собой комбинацию переборного и градиентного методов [64, 129, 169]. ГА относятся к адаптивным методам поиска экстремума в задачах функциональной оптимизации условной или безусловной [169], обеспечивают быстрый поиск близкого к оптимальному решения и выход из локальных экстремумов, а также позволяют улучшать результат на протяжении отведенного времени работы [20, 77, 169, 254, 317]. Это позволяет рассматривать их в качестве основного претендента на решение задач управления распределением ресурсов ИТ-инфраструктуры. Целесообразность использования ГА для решения задач распределения ресурсов частично доказана в работах [143, 158]. Там же сформирован ряд проблем, требующих решения для полноценного использования ГА при управлении ресурсами ЦОД.

Поскольку проблемы проектирования, анализа и управления ИТ-инфраструктурой составляют отдельный класс проблем большой размерности, то для ГА при решении этих проблем важным условием является эффективное управление сходимостью. Для этого необходимо иметь в ГА механизм, который, исходя из размерности задачи и заданной точности решений, мог бы найти компромисс между количеством итераций алгоритма и точностью решений. Поэтому возникает необходимость в разработке УГА, учитывающего особенности проблемы, характеристики предыдущей популяции и последовательности наилучших решений с целью определения такой последовательности операторов ГА, которая обеспечивает быструю сходимость.

Основой для разработки предложенного УГА послужили работы [20, 64, 129, 136, 169, 254, 281, 317, 332], посвященные проблемам исследования сходимости ГА. Эти работы сыграли важную роль в формулировании УГА, поскольку они знакомят с методиками оценки параметров процессов, протекающих в ГА, и с подходами к управлению скоростью его сходимости.

На рис. 10.3 приведена блок-схема УГА, представляющего собой универсальную схему применения ГА с контролируемой сходимостью к классу задач проектирования, анализа, оптимизации и управления ИТ-инфраструктурой.

Основное отличие предлагаемого УГА от рассмотренного выше КГА заключается в реализации механизма получения новой популяции особей на базе представителей ушедшей эпохи.

В КГА получение популяции новой эпохи происходит путем последовательного выполнения операторов кроссовера (C), мутации (M) (с определенной вероятностью, над потомком, полученным в результате применения кроссовера) и селекции (S) (отбор лучших из полученных результатов). Определенного рода селекция проводилась также перед кроссовером при отборе по определенному принципу особей, принимающих участие в кроссовере. Таким образом, в формировании популяции каждой последующей эпохи участвовали все три оператора ГА.

Подобная организация процесса дает сходимость поиска, но не гарантирует оптимальности решения. Это связано с наличием проблемы баланса «исследования» и «использования» [169, 254, 281, 332], которая заключается в балансирова-

нии между использованием уже существующего материала (закрепления оптимальных схем решения с помощью кроссовера) и исследованием новых оптимумов, которые не входят в изученное пространство (получение качественно новых схем посредством использования мутации). Большое количество мутаций может привести к разрушению схемы лучшего представителя текущей популяции, в то время как использование кроссовера приводит к повышению скорости сходимости процессов поиска к локальному или глобальному оптимуму.

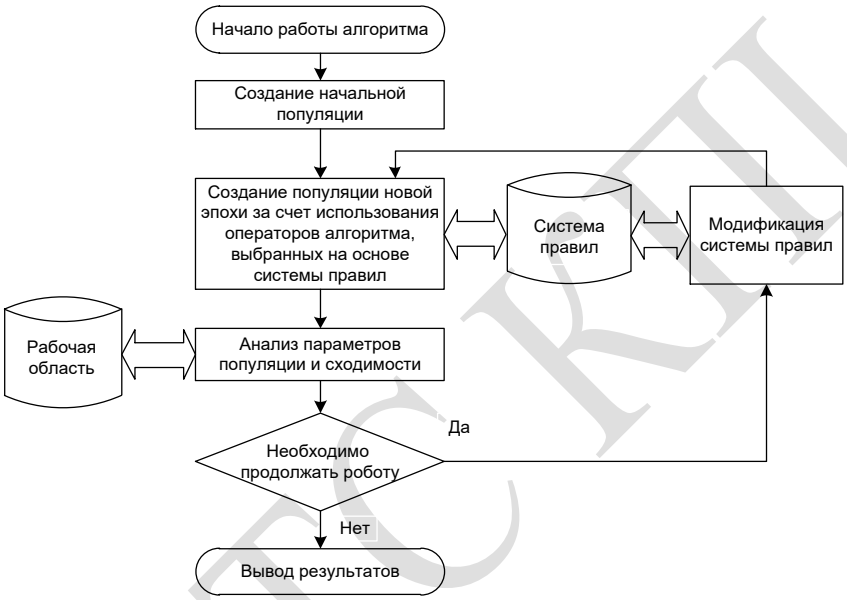


Рис. 10.3. Блок-схема управляемого генетического алгоритма

Из постановки проблемы видно, что при использовании КГА явных механизмов балансирования между «исследованием» и «использованием» нет. Процессы происходят без контроля в последовательности, заданной схемой самого алгоритма.

При использовании УГА (см. рис. 10.3) осуществляется балансирование между «исследованием» и «использованием» путем введения системы правил, на базе которой происходит выбор оператора кроссовера или мутации, посредством которого будет получена популяция следующей эпохи. Таких образом, в каждой отдельной эпохе процесса поиска проводится «использование» существующего материала (закрепление лучших решений) или «исследование» новых областей пространства решений. Селекция будет происходить в каждом из этих случаев, оставляя в составе популяции новой эпохи только лучших представителей из совокупности представителей прошлой эпохи и полученных на их базе потомков. В случае, когда количество полученных потомков будет слишком мало, совокупность, из которой будет делаться отбор, пополняется дополнительным вливанием случайно сгенерированных представителей для обеспечения разнообразия генетического материала популяции.

10.5.2. Суть предлагаемого управляемого генетического алгоритма

КГА обладает положительными свойствами: простота реализации; применимость для решения широкого класса различных задач; возможность решения задач с изменяющейся средой и крупномасштабных задач оптимизации; стойкость к попаданию в локальный экстремум. Он также не требует информации о поверхности ответа, а разрывы, имеющиеся на поверхности ответа, оказывают незначительное влияние на оптимизацию. Это позволяет ГА конкурировать по качеству предлагаемого решения с эвристическими алгоритмами. Однако КГА имеет и недостатки, к которым обычно относят отсутствие строго регламентированной системы правил применения операторов мутации (*M*) и кроссовера (*C*) [129]; плохую адаптируемость в случаях, когда необходимо найти глобальный экстремум; существенные затраты времени для поиска решения.

Предлагаемый управляемый ГА позволяет устранить или существенно сгладить недостатки КГА при решении задач распределения ВМ по серверам. Так, применение в УГА многоточечного кроссовера, когда выбираются несколько точек разрыва и родительские хромосомы обмениваются участками генетического кода, находящегося между этими точками, позволяет улучшить качество получаемого результата для задач с малыми ВМ.

Важным элементом ГА является размер популяции. Если популяция слишком мала, генетического материала может не хватить для решения задачи.

В КГА самые приспособленные особи не всегда переходят в следующее поколение. Это означает, что новая популяция не всегда содержит хромосому с наибольшим значением функции приспособленности из предыдущей популяции. В работе [20] предложена элитарная стратегия отбора популяции, которая заключается в защите наилучших хромосом на последующих итерациях. Элитарная стратегия с сохранением родительских особей применяется для предотвращения потери такой особи путем гарантированного включения ее в новую популяцию.

В предлагаемом УГА реализована элитарная стратегия отбора с частичной заменой популяции, которая заключается в том, что лучшие особи текущей популяции переходят в следующую без изменений и используются для генерации новых особей. Это позволяет сохранить наилучших представителей текущей популяции и получить качественный генетический материал в последующих популяциях.

Для сокращения времени работы алгоритма, возможности получения глобального экстремума и борьбы с другими недостатками КГА предлагается в процессе решения задачи осуществлять управление ГА путем изменения значений таких параметров, как размер начальной популяции, количество эпох генерации, скорость сходимости, давление отбора и коэффициент прироста популяции.

Для того чтобы сформировать систему правил УГА, необходимо выбрать параметры процесса, по которым будет оцениваться характер его сходимости и необходимость использования конкретного оператора *C* или *M* для каждой конкретной эпохи. Подходы к определению параметров таких процессов изложены в работах [281, 332] при исследовании сходимости ГА.

Введем следующие параметры.

При генерации популяции новой эпохи целесообразно иметь параметр, указывающий на вырождение популяции, связанное с недостаточным разнообра-

зием генетического материала, что, в свою очередь, требует проведения соответствующих мероприятий, например, дополнения генетического материала. Для этого вводится *коэффициент прироста популяции:*

$$r_{gg} = \frac{l}{b}, \tag{10.20}$$

где l – количество наследников, b – размер популяции.

При генерации популяции новой эпохи посредством использования одного из операторов *C* или *M* полученное количество потомков может быть очень малым по сравнению с размером популяции. Этот факт указывает на вырождение популяции, связанное с многообразием генетического материала, и требует соответствующих действий, связанных с дополнительным «вливанием» нового генетического материала. Для определения вырождения может применяться введенный коэффициент прироста популяции.

Для характеристики степени приближения текущей популяции к оптимальной используется параметр ρ – *давление отбора*. Алгоритм останавливается, когда значение целевой функции (ЦФ) для наиболее приспособленной особи совпадает или приближается к среднему значению ЦФ для всей популяции. Параметр давления отбора определим как отношение значения ЦФ, т. е. критерия оптимальности наиболее приспособленной особи $S_{\max}(x)$ к среднему значению $\langle S(x) \rangle$ ЦФ для всей популяции:

$$\rho = \frac{S_{\max}(x)}{\langle S(x) \rangle}. \tag{10.21}$$

Как видно из (10.21), значение параметра ρ не может быть меньше 1. Значение давления отбора равно единице в случае, когда ЦФ всех представителей популяции имеет одно и то же значение.

В качестве показателя, характеризующего сходимость УГА при переходе от эпохи к эпохе, используется *скорость сходимости* $\beta_{g,g+1}$, определяемая как разница между значениями давлений отбора популяций предыдущей ρ_g и текущей эпох ρ_{g+1}:

$$\beta_{g,g+1} = \rho_g - \rho_{g+1}. \tag{10.22}$$

Введение этих параметров позволяет обоснованно применять операторы для улучшения качества работы ГА.

Определим систему правил применения операторов для получения популяции следующей эпохи:

1. Если значение коэффициента прироста популяции r_{gg} меньше заданного критического значения $r_{кр}$, то возникает необходимость выполнить дополнительное «вливание» нового генетического материала, т. е. применить оператор RI. Формализованное описание правила:

$$\textbf{IF } (r_{gg} \le r_{кр}) \textbf{ THEN RI}. \tag{10.23}$$

АУТСКИЙ

2. Если скорость сходимости $\beta_{g,g+1}$ становится отрицательной при переходе от g-й до $(g + 1)$-й эпохи, то для формирования популяции следующей эпохи будет использоваться оператор кроссовера *C*.

При переходе от эпохи к эпохе средняя приспособленность популяции не уменьшается, поскольку происходит селекция только лучших представителей, поэтому скорость сходимости может стать отрицательной только при условии, что давление отбора текущей эпохи стало больше давления отбора предыдущей. Это, в свою очередь, возможно при условии нахождения нового максимального значения, т. е. нахождения лучшего решения задачи. Ради сохранения лучшего решения необходимо от стратегии «исследования» перейти к стратегии «использования», т. е. использовать кроссовер.

Запишем это правило в формализованном виде следующим образом:

$$\textbf{IF } (\beta_{g,g+1} < 0) \textbf{ THEN } \boldsymbol{C}. \tag{10.24}$$

3. Если значение давления отбора ρ_{g+1} и скорости сходимости $\beta_{g,g+1}$ не меньше граничных $\rho_{\text{гр}}$ и $\beta_{\text{гр}}$ соответственно, т. е. характер сходимости процессов еще не определен, нет схождения к «локальному» оптимуму, то для получения следующей популяции используем кроссовер.

Запишем это правило в формализованном виде следующим образом:

$$\textbf{IF } ((\rho_{g+1} \ge \rho_{\text{гр}}) \textbf{ AND } (\beta_{g,g+1} < \beta_{\text{гр}})) \textbf{ THEN } \boldsymbol{C}. \tag{10.25}$$

4. Если значения давления отбора и скорости сходимости меньше граничных, то для получения популяции следующей эпохи используется мутация *M*.

В этом случае поиск оптимального решения сходится к «локальному» оптимуму и нужно перейти от стратегии «использования» к стратегии «исследования», т. е. к мутации.

Формализованное описание правила имеет вид

$$\textbf{IF } ((\rho_{g+1} < \rho_{\text{гр}}) \textbf{ OR } (\beta_{g,g+1} < \beta_{\text{гр}})) \textbf{ THEN } \boldsymbol{M}. \tag{10.26}$$

Кроме правил выбора следующего оператора генерации популяции в систему правил также следует внести правила остановки процесса поиска.

5. Если значения скорости сходимости и давления отбора на протяжении двух эпох подряд были меньше критичных, то процесс поиска решения останавливается, решением является популяция текущей эпохи, дающая максимальное значение ЦФ. Условие проверки на популяциях двух последних эпох вводится для предоставления шанса мутации обнаружить новый лучший результат в неисследованных областях пространства решений.

Запишем это правило в формализованном виде следующим образом:

$$\begin{aligned}&\textbf{IF } (((\rho_{g+1} \le \rho_{\text{кр}}) \textbf{ AND } (\beta_{g,g+1} \le \beta_{\text{кр}})) \textbf{ AND } ((\rho_{g} \le \rho_{\text{кр}}) \textbf{ AND }\\ &(\beta_{g,g+1} \le \beta_{\text{кр}}))) \textbf{ THEN EXIT}.\end{aligned} \tag{10.27}$$

АУЦКП

6. Если максимальное значение ЦФ популяции текущей эпохи становится меньше, чем максимальное значение ЦФ прошлой эпохи, то процесс останавливается, последней правильной эпохой считается предыдущая, а лучшим результатом поиска – максимум ЦФ предыдущей эпохи.

Запишем это правило в формализованном виде следующим образом:

$$\textbf{IF } (S_{\max}(x)_{g+1} < S_{\max}(x)_g) \textbf{ THEN STEP\_BACK; EXIT.} \qquad (10.28)$$

На рис. 10.4 приведена схема выбора правил применения операторов для задачи распределения ВМ по серверам.

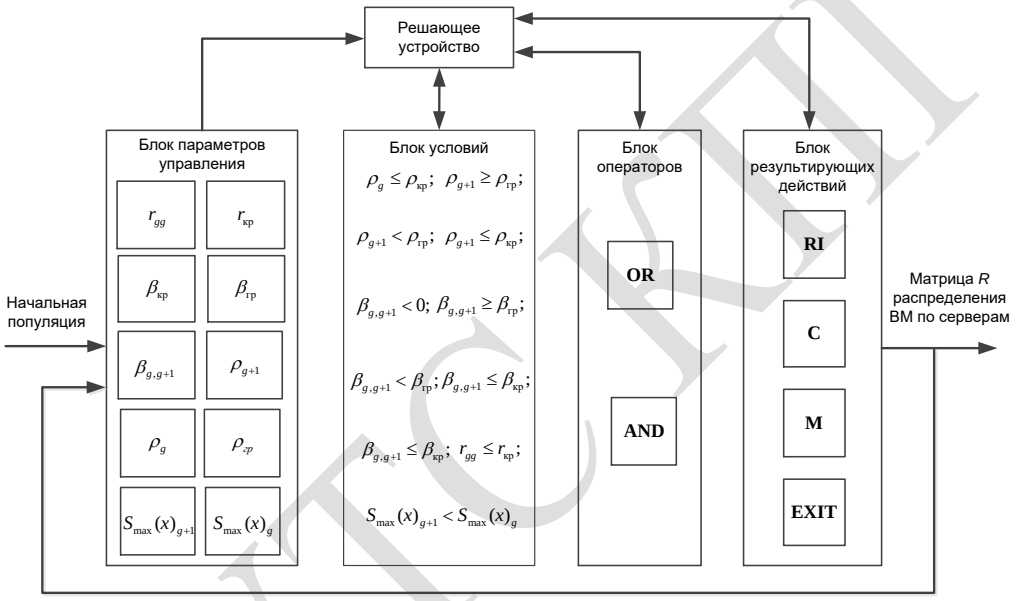


Рис. 10.4. Процесс выбора правил УГА

Как видно из правил (10.23)–(10.28), для организации управления процессами в УГА необходимо задать начальные значения параметров предельных и критических скоростей сходимости и давлений отбора – соответственно $\beta_{гр}$, $\beta_{кр}$, $\rho_{гр}$, $\rho_{кр}$, а также критичное значение $r_{кр}$ для коэффициента прироста популяции. Настройкой этих коэффициентов можно регулировать скорость сходимости поиска УГА и характер основных процессов, которые происходят при поиске решения. Эти процессы происходят на этапе модификации системы правил (рис. 10.3), что позволяет организовать своего рода обратную связь.

Отметим, что в общем случае множество правил управления процессами в ГА можно расширять, не ограничиваясь описанными выше правилами. Но введение дополнительных показателей сходимости ГА может привести к увеличению базы правил схемы, приведенной на рис. 10.3. При этом может возникнуть ситуация одновременного срабатывания нескольких правил, т. е. появление конфликтующих правил, по-разному определяющих ход одного и того же процесса. Тогда следует выбирать правила, используя их ранги, основанные на вкладе каждого правила в конечный результат, например, путем введения рангов правилам

системы или их динамической оценки в ходе процесса, основываясь на вкладе каждого правила в конечный результат. Хотя могут быть и другие схемы устранения конфликтных ситуаций.

Управляемый генетический алгоритм.

Шаг 1. Задаются параметры: количество особей в начальной популяции – *b*, количество эпох моделирования – *g*, значения предельных и критических скоростей сходимости и давлений отбора соответственно – $\beta_{гр}$, $\beta_{кр}$, $\rho_{гр}$ и $\rho_{кр}$.

Шаг 2 Генерация начальной популяции с учетом ограничения (10.12).

Шаг 3. Вычисление значения ЦФ для каждой особи.

Шаг 4. Сортировка особей текущей популяции в порядке убывания значения ЦФ.

Шаг 5. Расчет параметров управления: давления отбора, скорости сходимости, коэффициента прироста популяции согласно выражениям (10.22)–(10.20).

Шаг 6. Проверка условий:

- количество эпох равно нулю;
- значения скорости сходимости и давления отбора на протяжении двух эпох меньше критичных;
- максимум значения ЦФ для популяции текущей эпохи становится меньше, чем максимум значения ЦФ прошлой эпохи.

Шаг 7. Если любое из этих условий выполняется, то лучшая особь из текущей популяции представляется как решение задачи и алгоритм прекращает работу, иначе осуществляется переход на шаг 8.

Шаг 8. Анализ параметров управления УГА. В зависимости от значений параметров выбирается оператор кроссовера или мутации. На этапе нулевой эпохи работы алгоритма для генерации новой популяции используется оператор кроссовера. В соответствии с рис. 10.4 проводится анализ значений параметров управления УГА и выбирается оператор для генерации следующей популяции.

Шаг 9. Применяя стратегию элитарного отбора с сохранением родительских хромосом, получаем новую популяцию особей для моделирования следующей эпохи. Переход на шаг 3.

10.5.3. Исследование управляемого генетического алгоритма

Особенности проведения исследований. В реальных ЦОД упорядоченное множество ВМ *K* формируется на основании поступивших от клиентов заявок на вычислительные ресурсы. Поступление заявок и потребности в вычислительных ресурсах носят случайный характер. Заказчик может запросить для своих нужд ВМ с любым соотношением требуемой ОЗУ и процессорного времени.

Запрашиваемые ВМ могут быть с пропорциональными потребностями в памяти и процессорном времени, когда выполняется условие $\omega_j \cong \gamma_j$, $j = \overline{1,m}$, т. е. в относительных единицах запрашиваемая память γ_j каждой *j*-й ВМ, приблизительно равна запрашиваемому процессорному времени ω_j, и с непропорциональными потребностями, когда это условие не выполняется. Условно разделим ВМ на четыре класса – малые, средние, большие и очень большие. К малым относятся ВМ с требованиями в диапазоне 0,05–0,15, к средним – в диапазоне 0,15–0,3, к большим – 0,3–0,45 и к очень большим – 0,45–0,6 от возможностей сервера.

При проведении экспериментальных исследований осуществлялась генерация задаваемого количества множеств ВМ в соответствии с задаваемым соотношением разделения ВМ на пропорциональные и непропорциональные, а также задаваемым соотношением распределения ВМ по классам.

Основной задачей исследований было найти такие значения элементов матрицы X, при которых целевая функция (10.14) или (10.15) достигает минимума.

Результаты экспериментальных исследований. Оценка результатов работы УГА проводится в сравнении с алгоритмом последовательного заполнения и двумя предложенными эвристическими алгоритмами решения задачи распределения ВМ в условиях избытка ресурсов.

Результаты работы эвристических алгоритмов и УГА оцениваются по количеству высвобожденных серверов $N_{\text{В}}$ относительно количества серверов $N_{\text{ПЗ}}$, задействованных в результате работы алгоритма последовательного заполнения. Значение $N_{\text{В}}$ определяется как разность между количеством серверов $N_{\text{ПЗ}}$, задействованных в результате работы алгоритма последовательного заполнения, и количеством серверов $N_{\text{Т}}$, $N_{\text{М}}$, $N_{\text{УГА}}$, задействованных в результате работы эвристических алгоритмов «Т», «М» и УГА соответственно.

На рис. 10.5 показана модель эксперимента. Как видно из рисунка, исследования проведены для различных вариантов соотношения ресурсов серверов и потребностей ВМ в ресурсах и для различных вариантов потребностей ВМ в ОЗУ и мощности процессора.

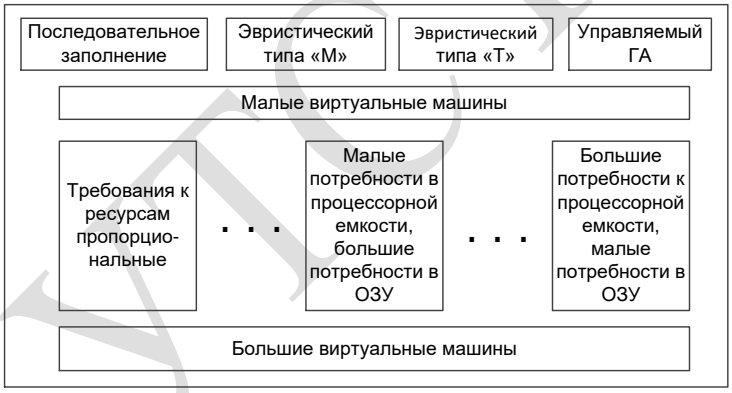


Рис. 10.5. Модель исследования УГА

На графиках (рис. 10.6) показана зависимость количества высвобожденных серверов $N_{\text{В}}$ от размерности задачи для различных вариантов соотношений запрашиваемых ресурсов. По оси абсцисс откладывается количество ВМ, которые необходимо расположить на серверах ЦОД. По оси ординат – количество высвобожденных серверов для каждого из алгоритмов.

1. Случай пропорционального соотношения требований к ресурсам, когда величина запрашиваемого процессорного времени в относительных единицах примерно равна величине запрашиваемой оперативной памяти, т. е. $\omega_j \approx \gamma_j$, $j = \overline{1,m}$.

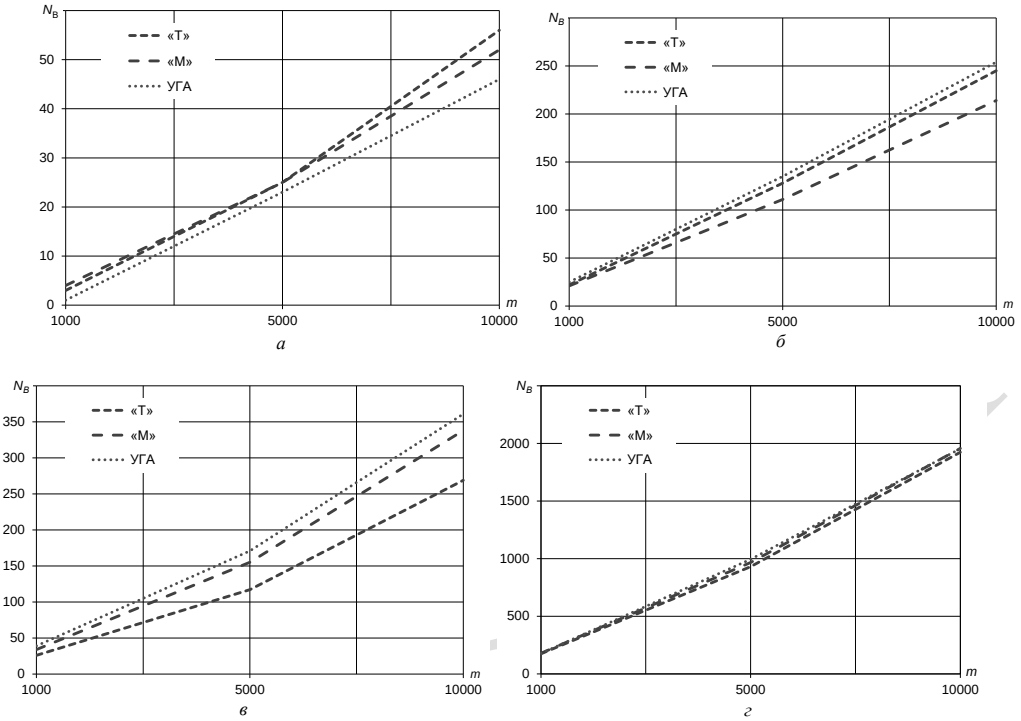


Рис. 10.6. Зависимость количества высвобожденных серверов от размерности задачи, когда требования к запрашиваемым ресурсам лежат в диапазонах:
а – [0,05; 0,15]; *б* – [0,15; 0,3]; *в* – [0,3; 0,45]; *г* – [0,45; 0,6]

Анализ графиков, приведенных на рис. 10.6, позволяет сделать следующие выводы:

- все предлагаемые алгоритмы эффективнее, чем алгоритм последовательного заполнения;
- если запрашиваемые ресурсы не превышают 0,15 от возможностей сервера, то алгоритм «Т» показывает лучшие результаты, но уступает алгоритмам «М» и УГА при увеличении объема запрашиваемых ресурсов ВМ и при ассиметричных требованиях к запрашиваемым ресурсам;
- УГА иногда проигрывает алгоритму «М» в связи с необходимостью многократного применения оператора многоточечного кроссовера для попадания в глобальный экстремум.

2. Случай непропорциональных требований к ресурсам, когда в относительных единицах величина запрашиваемого процессорного времени значительно превышает объем запрашиваемой оперативной памяти, т. е. $\omega_j \gg \gamma_j$, $j = \overline{1,m}$. Подобный тип задач возникает при наличии большого количества ВМ с приложениями, требующими сложных вычислений.

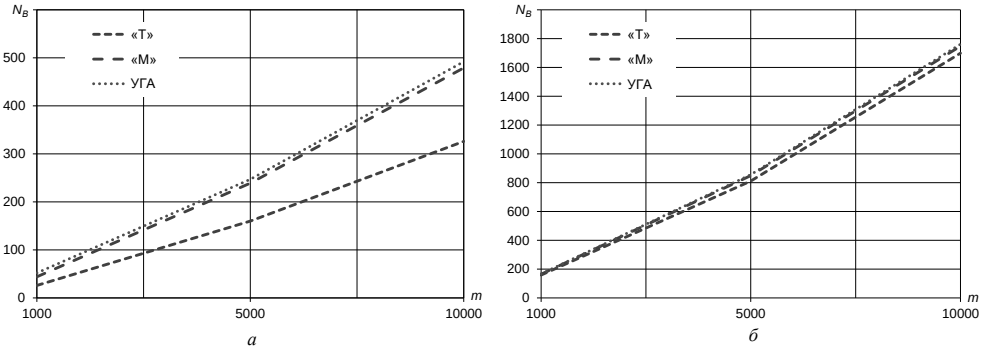


Рис. 10.7. Зависимость количества высвобожденных серверов от размерности задачи, когда требования к запрашиваемым ресурсам лежат в диапазонах: *а* – к процессорному времени – [0,3; 0,45], к ОЗУ – [0,05; 0,15]; *б* – к процессорному времени – [0,45; 0,6], к ОЗУ – [0,05; 0,15]

3. Величина запрашиваемого процессорного времени значительно меньше объема запрашиваемой оперативной памяти, т. е. $\omega_j \ll \gamma_j$, $j = \overline{1,m}$. Такая задача возникает, когда на серверах ЦОД размещаются ВМ с приложениями, требующими обработки больших объемов данных.

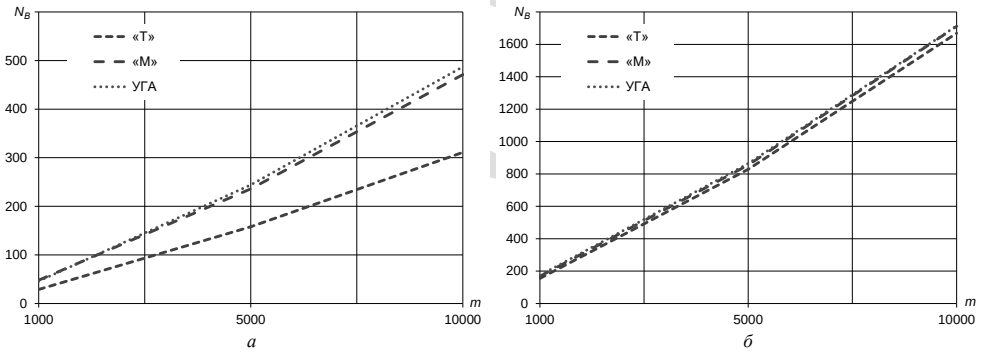


Рис. 10.8. Зависимость количества высвобожденных серверов от размерности задачи, когда требования к запрашиваемым ресурсам лежат в диапазонах: *а* – к процессорному времени – [0,05; 0,15], к ОЗУ – [0.3; 0.45]; *б* – к процессорному времени – [0,05; 0,15], к ОЗУ – [0,45; 0,6]

Как видно из графиков, приведенных на рис. 10.7 и 10.8, УГА имеет преимущество для решения задач распределения ВМ с непропорциональными запрашиваемыми ресурсами. Эвристические алгоритмы «Т» и «М» показывают примерно одинаковые результаты, проигрывая УГА 5–7%. Алгоритм последовательного заполнения показывает существенно худшие результаты. Это объясняется тем, что алгоритм последовательного заполнения анализирует использование только одного вида ресурса – процессорного времени или оперативной памяти, в то время как эвристические алгоритмы и УГА учитывают использование обоих типов ресурсов.

4. Самый распространенный случай на практике, когда количество запрашиваемого процессорного времени и оперативной памяти для всех ВМ распределено случайным образом в диапазоне [0,05; 0,6] (рис. 10.9).

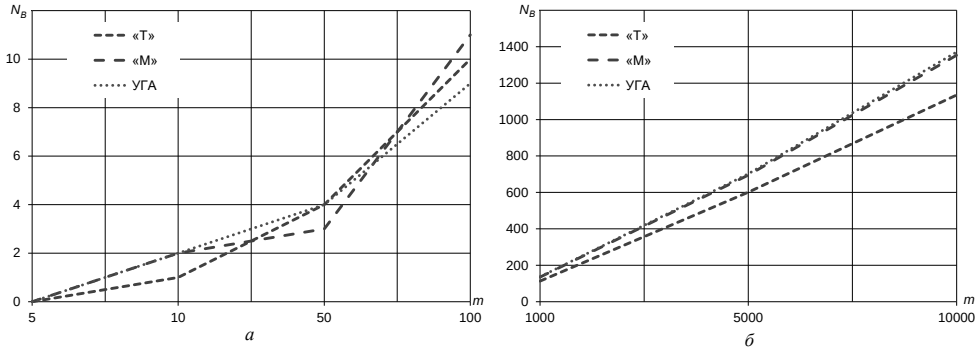


Рис. 10.9. Зависимость количества высвобожденных серверов от размерности задачи, когда требования к ресурсам случайным образом распределены в диапазоне [0,05; 0,6], а размерность задачи составляет: *а* – 5–100; *б* – 1000–10000

В данном случае предложенный УГА показывает лучшие результаты по высвобождению ресурсов ЦОД. Результаты экспериментов приведены в табл. 10.1. Исходные данные генерировались случайным образом с равномерным законом распределения.

Таблица 10.1. Зависимость значения ЦФ от размерности задачи

Количество размещаемых ВМ, шт.	Количество задействованных серверов, шт.			
	Последовательное заполнение	Алгоритм «Т»	Алгоритм «М»	УГА
5	2	2	2	2
10	6	5	4	4
50	22	18	19	18
100	44	34	33	35
500	238	183	172	171
1000	463	350	328	326
5000	2380	1779	1685	1678
10000	4699	3564	3346	3328

Как видно из табл. 10.1, применение УГА для решения задачи распределения ВМ по физическим серверам ЦОД за счет использования таких параметров управления, как размер начальной популяции, количество эпох генерации, скорость схождения, давление отбора и коэффициент прироста популяции, позволяет существенно сократить количество задействованных серверов, что непосредственно влечет за собой уменьшение экономических затрат центров обработки данных. Сравнение результатов предложенного в работе УГА с результатами работы классического ГА, приведенными в [158], позволяет сделать вывод о существенных преимуществах УГА.

Разработанная система правил, основанная на изменении параметров управления в процессе вычислений, строго регламентирует применение операторов мутации и кроссовера, что позволяет ускорить процесс получения результата. Применение оператора многоточечного кроссовера дает возможность решать за-

дачу распределения ресурсов при наличии малых ВМ с результатами, не уступающими по времени результатам эвристических алгоритмов, а по точности превосходящими их. Генерация начальной популяции с использованием эвристических алгоритмов позволяет получать готовое решение уже на первой эпохе, а в дальнейшем улучшать первоначальный результат.

На основании результатов проведенных исследований приведем рекомендации по выбору алгоритма распределения ВМ:

- для распределения непропорциональных по ресурсам ВМ целесообразно использовать УГА, тогда как для пропорциональных ВМ лучше использовать эвристические алгоритмы;
- УГА проигрывает эвристическим алгоритмам при размещении малых ВМ, поскольку требует большего времени вычислений для нахождения решения. Это объясняется многократным применением оператора многоточечного кроссовера;
- для средних ВМ УГА позволяет задействовать существенно меньшее количество серверов для размещения ВМ, чем эвристические алгоритмы, причем экономия возрастает с увеличением размерности задачи;
- для больших ВМ все алгоритмы предлагают примерно одинаковые по качеству решения, поскольку сокращается количество возможных вариантов расположения ВМ на серверах.

Таким образом, разработанный УГА управляет применением операторов в ходе решения задачи. Определена система правил и условий применения операторов кроссовера и мутации. Показано, что лучшие результаты дает УГА и алгоритм типа «М», которые уступают алгоритму типа «Т» только при размещении малых ВМ с пропорциональными требованиями к ресурсам. В последнем случае алгоритм «Т» позволяет обеспечить экономию 5–18% ресурсов ЦОД по сравнению с алгоритмом последовательного заполнения.

10.6. АДАПТИВНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ ЦОД

10.6.1. Особенности адаптивных генетических алгоритмов

Для эффективного использования дорогостоящих вычислительных ресурсов ЦОД, хранилищ данных и других ИКТ необходимо, чтобы СУИ обеспечивали рациональное распределение ВМ, осуществляли управление нагрузкой и решали множество других подобных задач. В работе [161] показаны преимущества применения ГА по сравнению с эвристическими методами для решения задач такого рода. В тоже время требуют решения фундаментальные проблемы, обусловленные спецификой этих алгоритмов, поскольку ГА одновременно используют несколько типов генетических операторов – унарных, бинарных и множественных. Трудно подобрать стратегию генерации значений вероятностей использования отдельных операторов так, чтобы их применение давало положительный результат на протяжении всего времени работы ГА. Кроме того, каждый из операторов имеет множество параметров, оказывающих влияние на результаты работы алгоритма, и нахождение оптимальных значений этих параметров представляет собой достаточно сложную задачу. Решение этих проблем ГА позволит повысить его эффективность. Поскольку в общем виде оно не представляется возможным,

необходимо разработать эффективную стратегию подбора параметров операторов и определения вероятностей применения операторов в процессе эволюции ГА. Причем этот процесс должен проходить без участия администратора и позволять работать с различными типами задач управления ИТ-инфраструктурой.

В работах [143, 158] показано, что применение ГА для решения задач распределения вычислительных ресурсов позволяет получать достаточно производительные решения. В этих исследованиях сформулирован список проблем, которые необходимо решить при использовании ГА для задач различной размерности и ограничений.

В работах [185, 352] предложена идея адаптации генетических операторов и подстройки вероятностей их использования. Недостатком этих работ является адаптация только одного оператора – кроссовера и использование возможных подходов к адаптации независимо друг от друга. Предложенный в работе [161] и описанный выше УГА позволяет корректировать параметры работы алгоритма на всех этапах решения задачи. Помимо этого УГА решает некоторые проблемы КГА, в частности вырождение популяции, попадание в локальные экстремумы и т. д. Его главными недостатками являются участие администратора и привязка к узкому классу задач.

Использование УГА позволяет регламентировать использование операторов. Однако остается проблема выбора параметров, с которыми будут использованы операторы. Также необходимо разработать систему поощрения операторов по такому естественному принципу: если оператор с определенными параметрами позволяет быстро достигать сходимости ГА, его следует использовать чаще. Причем этот процесс должен проходить без участия администратора и распространяться на различные типы задач.

Поэтому необходимо разработать ГА, способный самостоятельно и динамически изменять вероятности использования операторов ГА и значения параметров этих операторов в зависимости от результатов, получаемых в ходе решения задачи.

Генетические алгоритмы, параметры которых могут изменяться в процессе работы для достижения высокой скорости сходимости при высоком качестве, называются *адаптивными* или *динамическими* [37, 38]. Задача эффективного управление сходимостью ГА не решена до настоящего времени и представляет собой одну из самых интересных и актуальных задач разработки и внедрения адаптивных генетических алгоритмов (АГА).

Для разработки адаптивного ГА воспользуемся результатами исследований, выполненных в работах [37, 38]. Обобщая результаты детального анализа исследований АГА, их авторы выделили наиболее значимые операторы и параметры, которые могут участвовать в процессе адаптации и работы АГА:

- оператор кроссовера и параметры – количество точек и вероятность кроссовера;
- оператор мутации и параметр вероятность мутации;
- механизм селекции и давление отбора;
- размер популяции (фиксированный или динамически изменяемый).

Заметим, что параметры и их значения можно подбирать экспериментально. Значительно больший интерес представляет разработка АГА, самостоятельно выбирающих операторы, которые будут применяться по ходу работы АГА, и подбирать значения параметров этих операторов. Обычно подбор параметров осуществляется путем выполнения множества тестов и определения операторов

и значений параметров, при которых ГА дает наилучшие результаты [37]. При этом в процессе определения параметров участвует оператор, принимающий решение о дальнейшем ходе АГА, что не позволяет достичь оптимальной продуктивности ГА.

Параграф посвящен разработке и исследованию адаптивного генетического алгоритма, в котором реализованы различные механизмы управления сходимостью.

Анализ работ [37, 185, 269, 276, 282, 298, 299, 314, 352, 372] позволил сформировать концепцию АГА, определить последовательность и сущность его основных шагов. Наибольшим отличием АГА от УГА является полная автономность работы и возможность адаптации к решаемой задаче и условиям протекания процесса решения задачи.

10.6.2. Общая постановка задачи адаптации генетического алгоритма

Предлагаемая идея довольно проста: можно ли придать ГА свойства классических алгоритмов поиска, которые были популярными при решении инженерных задач в 70-80-х годах прошлого столетия? ГА, «стреляя» по области решений и отбирая лучшие популяции, постоянно улучшает решение. То же самое делают и алгоритмы поиска. Однако последние используют некоторую дополнительную информацию для увеличения скорости сходимости. Нельзя ли наделить алгоритмы поиска свойствами ГА? В работе [161] был предложен УГА, использование которого позволяет получать хорошие результаты. Скорость сходимости увеличивается за счет подбора вероятностей основных операций ГА на основе результатов, полученных на предыдущих шагах работы. Специальные правила, построенные на использовании параметров давления отбора, скорости сходимости и коэффициентов прироста популяции, позволяют изменять вероятности применения генетических операторов и тем самым увеличить скорость сходимости.

Следующим естественным шагом должна быть адаптация ГА путем отбора тех же вероятностей, но на основе некоторого критерия оценки популяции. Таким критерием может служить вероятность получения оптимального решения на основе популяции либо скорости сходимости.

Поиск эффективной стратегии выбора частоты применения генетических операторов и параметров этих операторов для решения задач, возникающих при управлении ИТ-инфраструктурой, является прямой задачей адаптации ГА.

Выделяют три типа адаптации: структурную, параметрическую и адаптацию алгоритма [101]. Структурная адаптация предполагает изменение структуры объекта. Для класса задач, решаемых в данной работе, такой возможности не существует, поэтому будут использованы только два оставшихся типа адаптации. Параметрическая адаптация подразумевает коррекцию параметров в процессе поиска решения, в то время как адаптация алгоритма подразумевает внесение изменений в сам алгоритм.

Общая постановка задачи адаптации ГА сводится к минимизации некоторой функции F, которая служит критерием адаптации для ГА и зависит от таких параметров как типы операторов, которые следует использовать в эволюционном процессе, частота применения этих операторов и значения параметров, с которыми применяются операторы. Определим функцию адаптации следующим образом:

$$F(M,C,\alpha_M,\alpha_C,\beta_M,\beta_C), \tag{10.29}$$

где *M, C* – параметры, регламентирующие применение операторов мутации и кроссовера, и принимающие значения из множества {0,1}, причем равенство параметра нулю означает, что оператор не участвует в эволюционном процессе, а единица означает, что оператор принимает участие в процессе эволюции; α_M, α_C – частоты использования операторов, принимающие значения из интервала [0;1]; β_M,β_C – множества возможных значений операторов мутации и кроссовера.

Нахождение явного вида функции (10.29) даже для простых случаев требует огромного количества вычислений, что сведет на нет все преимущества ГА. Использование структурной адаптации в рамках решения задач управления ИТ-инфраструктурой невозможно, поэтому при осуществлении адаптации предлагается использовать некоторую информацию касательно свойств функции (10.29). Такой информацией являются типы операторов и параметры этих операторов, с которыми они принимают участие в эволюционном процессе.

Предлагается решение задачи адаптации ГА осуществлять последовательным циклическим использование двух ГА. Первый применяется с целью параметрической адаптации и используется для настройки параметров генетических операторов, увеличивающих скорость сходимости. Второй служит для алгоритмической адаптации и осуществляет подстройку ГА в зависимости от результатов, получаемых в ходе решения задачи. Таким образом, предлагается использовать два механизма адаптации ГА, один из которых обеспечивает динамическое изменение значений вероятностей применения операторов, а второй подстраивает значения его параметров. В первом случае операторы, которые в процессе работы позволили улучшить решение, чаще применяются при генерации следующих поколений. Во втором случае контролируются и изменяются значения параметров операторов.

Схема работы предлагаемого АГА приведена на рис. 10.10. Математической моделью задачи являются задачи линейного программирования. К этому классу задач относится большинство задач, возникающих при управлении ИТ-инфраструктурой, например, задача распределения ресурсов, задача эффективного размещения файлов на серверах хранилища данных и т. п. Блок проверки условий служит для задания критериев остановки работы АГА. Это могут быть ограничения по времени работы либо по количеству эпох эволюционирования.

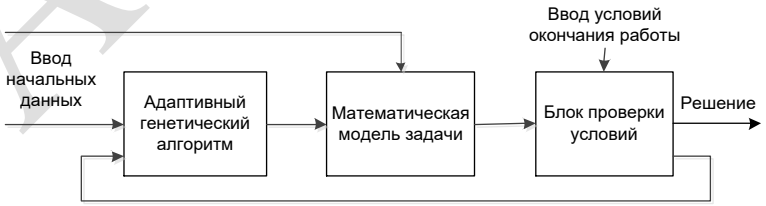


Рис. 10.10. Схема работы адаптивного генетического алгоритма

Как правило, для адаптации ГА [298] под решаемые задачи используется некоторый параметр управления [299]. Эффективность детерминистического управления доказана для некоторых задач [185], однако его обобщение является проблематичным. Адаптивное управление [269] использует обратную связь для

определения того, как должны измениться параметры. Под адаптивным управлением [352] понимается введение дополнительной информации, позволяющей корректировать поведение операторов.

На сегодняшний день основные усилия исследователей ГА направлены на адаптацию только одного из параметров. Наиболее полное сочетание форм управления [299] было приведено в работе [282]. В ней адаптация проводилась одновременно по трем параметрам – вероятности мутации, кроссовера и размера популяции. Другие адаптивные алгоритмы представлены в работах [276, 314, 372].

В данной работе предлагается обобщенная схема ГА, способного по ходу работы решать задачи параметрической и алгоритмической адаптации – динамически адаптировать как вероятность использования, так и значения параметров каждого из операторов. Вероятности применения операторов адаптируются детерминистическим способом. В этом случае все операторы применяются поочередно, а оператор, который в процессе работы позволил улучшить решение, автоматически будет использован в следующих эпохах. Таким образом, операторы, способствующие сходимости задачи, при генерации следующих поколений применяются чаще.

10.6.3. Суть предлагаемого адаптивного генетического алгоритма

Приведем базовые определения и введем некоторые обобщения для классических операторов.

Кодирование данных. Каждая хромосома представляет собой последовательность битов. Все хромосомы $n = 2^k$, $k > 0$, $b_i \in \{0,1\}$, для всех $i = \overline{1, n}$, имеют длину 2^k, где k – константа.

Мутация. Стандартный оператор мутации изменяет гены путем инвертирования. Этот тип оператора имеет параметр *alter_rate*, который определяет количество генов в хромосоме, подлежащих инвертированию.

Предлагается традиционный алгоритм мутации усовершенствовать таким образом, чтобы вместо решения задачи для каждого гена о необходимости его инвертирования проводить инвертирование значения гена, равного 1, в 0 с вероятностью p_{10}, а 0 заменять на 1 с вероятностью p_{01}. Такой подход позволяет применять ГА для подбора наилучших значений параметров генетических операторов. Например, если в родительской хромосоме некоторые значения вероятностей p_{10} и p_{01} не дали производительного решения, то в хромосоме потомка эти параметры обмениваются значениями, что может привести к улучшению результата.

Кроссовер. Предлагается использовать модифицированный оператор кроссовера, который генерирует c, $1 \le c < n$, точек кроссовера, делящих хромосомы на $c + 1$ участков. При этом одно потомство наследует четные участки хромосомы первого родителя и нечетные участки второго родителя, второе потомство получается противоположным образом.

Основным отличием предлагаемого алгоритма является использование двух стратегий адаптации, в связи с чем процесс получения решения происходит путем циклического повторения двух этапов. На первом этапе с помощью параметрической адаптации выбираются наиболее производительные значения парамет-

ров для каждого из используемых типов операторов. На втором этапе оценивается параметр производительности и с учетом результатов алгоритмической адаптации выбирается наиболее производительный оператор. Если полученное в результате решение не удовлетворяет заданным критериям, процесс повторяется с первого этапа.

ГА используют несколько типов операторов – унарные (мутация), бинарные (кроссовер), множественные (многоточечный кроссовер). Каждый тип оператора имеет набор параметров, влияющих на его поведение, а работа ГА начинается с определенными значениями параметров.

Пусть задано множество типов операторов ГА $T = \{t^1, ..., t^j, ..., t^J\}$, где $j = \overline{1, J}$ – количество типов операторов, применяемых в ГА. Например, модифицированный оператор кроссовера или модифицированный оператор мутации. Каждый тип оператора t^j, $j = \overline{1, J}$ имеет набор параметров $P(t^j) = (p_{j,1}, ..., p_{j,k}, ..., p_{j,K})$, $k = \overline{1, K}$. Например, модифицированный оператор мутации имеет два параметра – вероятности p_{10} и p_{01}, а кроссовер – один параметр, а именно количество точек кроссовера.

Для генерации новых поколений ГА использует операторы с параметрами, соответствующими его типу. Параметры принимают значения из некоторого множества или интервала. Так, значения параметров мутации выбираются из интервала (0, 1), тогда как параметр кроссовера может принимать любое положительное целочисленное значение, меньшее $(n-1)$.

Для увеличения шансов на выживание и размножение операторов со значениями параметров, показавших лучшие результаты, применяется параметрическая адаптация. Рассмотрим АГА, который использует типы операторов из множества *T*. Для каждого оператора типа t^j, $j = \overline{1, J}$ строится популяция значений параметров. Например, если для АГА определены два оператора: модифицированный оператор мутации и модифицированный оператор кроссовера, то фиксированным набором экземпляров операторов могут быть: для мутации – $(alter\_rate, p_{10}, p_{01}) \in \{(0,1; 0,1; 0,9), \quad (0,7; 0,8; 0), \quad (0,4; 0,1; 0,1)\}$, а для кроссовера – $(c) \in \{(1), (2), (n-1)\}$.

Обозначим набор используемых операторов АГА как $O = \{op_1, ..., op_h, ..., op_H\}$, $h = \overline{1, H}$, где H – количество операторов АГА.

Поведение каждого оператора корректируется изменением значений его параметров с использованием для каждого оператора эволюционной процедуры, функционирующей в пространстве возможных значений оператора. Для каждого оператора с момента запуска АГА проводятся изменения значений его параметров. Поскольку оптимизируется относительно небольшое пространство поиска, то для поиска набора значений операторов, приводящих к улучшению результатов работы АГА, для каждого отдельного оператора также используется ГА. ГА, осуществляющий поиск лучших значений операторов, обозначим ОГА.

Популяция для каждого оператора состоит из хромосом, представляющих собой набор возможных значений параметров этого оператора. Для генерации новых популяций используется одноточечный кроссовер. ОГА, применяемый

АУПСКП

для поиска в пространстве значений отдельного типа оператора наилучших значений, запускается как процедура главного ГА, обозначаемого далее ГГА и работающего над непосредственным поиском решения задачи. Полученное в результате использования ОГА решение служит исходными данными для ГГА. При этом АГА рассматривается как совокупность циклов ОГА и ГГА.

На этапе работы ГГА решается задача алгоритмической адаптации – увеличение вероятностей использования операторов, которые дают лучшие решения. Для оценки работы операторов введем следующие понятия.

Событие – применение конкретного оператора. *Абсолютное улучшение* – событие, когда значение ЦФ нового поколения больше, чем значение ЦФ предшествующих поколений. *Улучшение* – новое поколение имеет значение ЦФ лучшее, чем у родителей. *Стабилизация решения* – значение ЦФ нового поколения незначительно отличается от родительского на протяжении ряда эпох. *Вырождение* – новое поколение имеет худшее значение ЦФ, чем родители, и ни одно из поколений не лучше, чем родительское.

Введем параметр производительности, показывающий эффективность использования оператора в текущем поколении и используемый в качестве обратной связи для эволюционного процесса. Основываясь на значениях показателей производительности, АГА корректирует значения вероятностей использования операторов. Параметр производительности для оператора op_h, $h = \overline{1,H}$ определим как функцию четырех переменных $\pi_{op_h}(ae, e, pw, w)$, где ae – количество абсолютных улучшений, e – количество улучшений, pw – количество стабилизированных решений, w – количество вырождений. Общее количество событий на шаге работы АГА определяется как $N = ae + e + pw + w$.

Параметр производительности введен с целью определения, какой из операторов дает лучшее решение на данном шаге работы АГА. Частота использования операторов учитывается относительными частотами появления событий определенного типа. При $N > 0$ относительные частоты рассчитываются так: для абсолютного улучшения $\varphi_{ae} = ae / N$; улучшения – $\varphi_e = e / N$; стабилизированного решения – $\varphi_{pw} = pw / N$; вырождения – $\varphi_w = w / N$.

Предлагается следующий порядок сравнения параметров производительности, который на примере двух операторов можно формулировать следующим образом:

а) превалирует оператор, параметр производительности которого характеризируется большим числом абсолютных улучшений;

б) если количество абсолютных улучшений одинаково, сравнение производится по количеству улучшений;

в) если количество улучшений одинаково, сравниваются количества стабилизированных решений;

г) если последнее не позволяет выделить лучший оператор, то более производительным будет тот оператор, у которого меньше количество вырождений;

д) при равном количестве вырождений используется мутация.

Для определения порядка использования операторов при работе АГА введено понятие *награда*. Каждому оператору $op_h \in O$, $h = \overline{1,H}$ приписывается награда $\rho(op_h)$, которая увеличивается в результате накопления положительного опыта. В первую очередь используется оператор, который имеет наибольшую

награду. Значение параметра $\rho(op_h)$, $h=\overline{1,H}$ постоянно обновляется, а опыт, приобретенный при последних испытаниях, рассматривается как более актуальный.

Пусть $\chi(op_h)$, $h=\overline{1,H}$ – ранг оператора op_h, присваиваемый в зависимости от производительности таким образом, что наивысший ранг получает наиболее производительный тип оператора. Присвоение ранга проводится после завершения работы ОГА. Награды обновляются в конце каждой эпохи АГА согласно формуле

$$\rho(op_h)=\delta\rho(op_h)+\beta+\gamma\chi(op_h), h=\overline{1,H}, \qquad (10.30)$$

где δ – коэффициент ослабления, являющийся константой из множества {0, 1}. Причем δ = 0 для оператора, только вступившего в работу, а δ = 1 означает, что весь предыдущий опыт оператора полностью учтен. Коэффициент усиления γ служит для награждения лучших операторов. Коэффициент β, обычно имеющий значение меньше γ, используется для гарантии того, что оператор будет обязательно использован на этапе АГА.

Ниже приведен пример предлагаемого АГА для двух операторов – кроссовера и мутации, имеющего ограничения на продолжительность выполнения в виде количества эпох работы.

Шаг 1. Задать условия остановки алгоритма в виде количества эпох эволюционирования ГА или время работы алгоритма.

Шаг 2. Инициализировать начальную популяцию случайным образом с учетом ограничений (10.9)–(10.12) и следующего правила: каждый раз при обращении к популяции для решения задачи ее особи сортируются в порядке убывания значения ЦФ. Наилучший представитель популяции – хранимое решение задачи. Лучшая из популяций запоминается как хранимая популяция. На начальном этапе хранимой является первичная популяция.

Шаг 3. Случайным образом инициализировать популяцию значений операторов с учетом ограничений, накладываемых для каждого типа операторов. Например, для мутации и кроссовера могут использоваться следующие значения параметров $(alter\_rate, p_{10}, p_{01}) \in \{(0,1; 0,1; 0,1),\ (0,2; 0,2; 0,2),\ (0,3; 0,3; 0,3)\}$ и $(c) \in \{(1),(2),(3)\}$ соответственно.

Шаг 4. Использовать каждое из значений параметров операторов для генерации промежуточных популяций. С помощью ЦФ выбрать значения, позволившие достичь наибольших показателей производительности для каждого оператора. Например, это может быть $(alter\_rate, p_{10}, p_{01}) = (0,1; 0,1; 0,1)$ для мутации, а для кроссовера – $(c) = (1)$. В случае улучшения полученных результатов изменить хранимую популяцию, используя результаты, полученные посредством самого производительного их двух операторов, и перезаписать хранимое решение задачи. При равенстве операторов по производительности выбрать оператор мутации. Если не удалось улучшить начальное решение, перейти на шаг 3 и использовать ОГА для корректировки значений параметров применением оператора кроссовера. Если не удалось улучшить параметры операторов путем кроссовера, применить мутацию во избежание возможного попадания в локальные экстремумы. Применение параметра мутации при корректировке значений параметров

операторов проводится по циклу с помощью следующих правил: для кроссовера – увеличить на единицу количество точек кроссовера, а если количество точек равно $(n-1)$, принять следующее количество точек кроссовера равным 1; для мутации – увеличить каждое из значений параметров мутации на 0,1, а если значение любого из параметров равно 0,9, принять следующее значение равным 0,1. Если решение не улучшается, закончить работу и считать полученное решение наиболее производительным.

Шаг 5. Присвоить ранги операторам, рассчитать награды, применить операторы M раз (общее количество этапов работы ГГА равно $2\times M$ на первой эпохе и $3\times M$ на всех последующих) в порядке уменьшения величины награды. Если применение оператора улучшило решение, то использовать решение в качестве нового хранимого решения задачи, после чего продолжить работу с улучшенной популяцией.

Шаг 6. По истечении заданного числа этапов ГГА обновить значение параметров производительности и выбрать наиболее производительный оператор. Например, если $(alter\_rate, p_{10}, p_{01}) = (0{,}1;\ 0{,}1;\ 0{,}1)$ оказался наиболее производительным, то необходимо сохранить его и использовать в следующих эпохах. Перезаписать хранимое решение задачи.

Шаг 7. Если условия остановки АГА не выполнены, то перейти на шаг 3 и начать новую эпоху. На этапе работы ОГА выбрать наиболее производительные значения для всех типов операторов. Например, если для мутации $(alter\_rate, p_{10}, p_{01}) = (0{,}2;\ 0{,}2;\ 0{,}2)$, а для кроссовера – (c) =(1), то на этапе ГГА работа над популяцией будет проводиться для трех операторов (оператор мутации $(alter\_rate, p_{10}, p_{01}) = (0{,}1;\ 0{,}1;\ 0{,}1)$ переходит из предыдущей эпохи). По окончании работы ГГА пересчитать и сравнить параметры производительности, выбрать лучший из трех операторов и перейти к следующей эпохе (шаг 3).

Шаг 8. Если условия остановки АГА выполняются, то завершить работу, использовать текущее хранимое решение в качестве решения задачи, иначе перейти на шаг 3 и продолжать работу до выполнения условия остановки.

10.6.4. Программная реализация модели управления распределением нагрузки и ресурсов

С целью проверки эффективности разработанных алгоритмов поиска оптимальных результатов многокритериальной задачи управления распределения нагрузки и ресурсов при виртуальном хостинге выполнена программная реализация алгоритмов с общим интерфейсом с помощью среды Microsoft Visual Studio 2010. Языком программной реализации выбран C#.

Особенности использованных математических моделей. Кроме размещения на серверах виртуальных машин предусматривается возможность размещения на серверах ЦОД экземпляров приложений.

В последнем случае на множестве N серверов (10.1), количество которых равно n, распределяется множество приложений:

$$A = \{A_1, \ldots, A_m\}. \tag{10.31}$$

При этом математические модели, связанные с управлением распределением ресурсов при распределении экземпляров приложений, имеют приведенные ниже особенности.

Оптимальное размещение приложений должно учитывать следующие ограничения:

1) суммарные независимые от нагрузки требования определенных для сервера экземпляров приложений не должны превышать независимые от нагрузки параметры сервера;

2) требования приложений, зависящие от нагрузки, определяются на основе потенциала всех пользователей и должны быть обеспечены суммарными возможностями зависимых от загрузки параметров тех серверов, на которых размещаются экземпляры этого приложения;

3) ограничения, определенные матрицей *R* (10.5) и вектором *C* (10.3).

В качестве критериев можно использовать:

1) максимальную суммарную важность приложений, которые получили ресурсы;

2) равномерную загрузку серверов;

3) максимальные суммарные эксплуатационные затраты на высвобожденные сервера;

4) минимальную суммарную стоимость реализации операций перехода от предыдущего размещения к новому.

Модель планирования распределения нагрузки и ресурсов для задачи максимальной экономии ресурсов. В зависимости от стратегии хостинговой компании можно выбрать одну из моделей, которые различаются критериям.

Общим для каждой из задач есть правила отбора, сформулированные на основе удовлетворения потребностей конкретных экземпляров приложений ресурсами серверов:

$$\sum_{j=1}^{m} x_{ji}\gamma_j \le \Gamma_i, \overline{i=1,n}. \tag{10.32}$$

Ограничение (10.32) требует, чтобы емкость ОЗУ сервера N_i была достаточной для размещения запланированных для размещения на нем экземпляров приложений;

$$\sum_{j=1}^{m} x_{ji}\omega_j \le \Omega_j, \ \overline{i=1,n}. \tag{10.33}$$

Ограничение (10.33) требует, чтобы потребности всех пользователей приложения A_j в емкости процессоров удовлетворяли в совокупности выбранным физическим серверам с экземплярами этого приложения;

$$\sum_{i=1}^{n} x_{ji} \ge 1, \ \overline{j=1,m}. \tag{10.34}$$

Ограничение (10.34) требует, чтобы был хотя бы один экземпляр каждого приложения на каком-то сервере.

Если хостинговой организации важна экономия ресурсов, то возникает задача минимизации расходов на поддержку клиентских запросов. т. е. выбирается

АУГСП

такое минимальное по затратам на их поддержку подмножество серверов, которое сможет поддержать необходимое для удовлетворения требований существующих пользователей множество экземпляров приложений.

Обозначим через s_i эксплуатационные расходы на поддержку сервера N_i (электроэнергия, работы, комплектующие) и введем еще одну переменную y_i, $i=\overline{1,n}$, которая характеризует наличие нагрузки на сервер и приобретает следующие значения:

$$y_i = \begin{cases} 1, \text{если физический сервер } N_i \text{ поддерживает хотя бы один экземпляр любого приложения;} \\ 0, \text{ если физический сервер } N_i \text{ не поддерживает ни одного экземпляра никакого приложения.} \end{cases} \quad (10.35)$$

Легко увидеть связь между введенными переменными:

$$y_i = x_{1i} \vee x_{2i} \vee \ldots x_{mi}, \; i=\overline{1,n}. \quad (10.36)$$

С использованием введенных обозначений критерий этой задачи приобретет следующий вид:

$$\min \sum_{i=n}^{n} s_i y_i, \quad (10.37)$$

где s_i – стоимость поддержки i-го сервера серверного парка ЦОД.

Исходя из специфики задачи, можно перейти от минимизации затрат на поддержку серверов к максимизации получаемой при этом экономии. При этом критерий оптимальности приобретает следующий вид:

$$\max\left(S - \sum_{i=n}^{n} s_i y_i \right), \quad (10.38)$$

где S – суммарная стоимость поддержки серверов серверного парка конкретной конфигурации.

Лучшие по критерию значения следует выбирать из решений, соответствующих структуре булевых значений матрицы R (10.5) и удовлетворяющих ограничениям (10.32)–(10.34).

Математические модели управления нагрузкой и ресурсами хостинга с кластеризацией. При хостинге с кластеризацией появляется возможность реализации концепции параллельных вычислений, что при управления ресурсами и нагрузкой позволяет в более полной мере использовать в ЦОД процессорные мощности. При этом возникают избыточные затраты ОЗУ, вызванные несколькими реализациями одного и того же экземпляра определенного приложения сразу на нескольких серверах хостинга.

Формулировку модели управления ресурсами при хостинге с кластеризацией удобно проводить в виде сравнения с моделью хостинга без кластеризации. При этом описание модели будет выглядеть следующим образом.

АУТСКП

Аналогично хостингу с кластеризацией вводится множество серверов (10.1) и приложений (10.31), а также параметров, характеризующих ресурсные мощности серверов – Ω_i и Γ_i.

Первое отличие от модели без кластеризации проявляется в формулировании требований конкретного приложения:

ω_j – требование к суммарной процессорной емкости серверов, на которых расположены экземпляры применения A_j;

γ_j – требование к ОЗУ каждого из серверов, на которых расположены экземпляры приложения A_j.

Подобные различия в формулировании требований к серверным ресурсам объясняются необходимостью организации распределенных вычислений на группе серверов. Каждый сервер из группы должен, кроме доли процессорного времени, выделить ресурсы ОЗУ и поддерживать через сеть связь с другими серверами группы для сопоставления процессов и обеспечения суммарного результата выполнения приложения.

Очевидно, что требования к суммарной процессорной емкости серверов, которые определяются на основе потребностей всех пользователей, могут обеспечить несколько серверов своей суммарной процессорной емкостью. Тогда на каждом из этих серверов размещается экземпляр приложения. Но величина γ_j определяется для экземпляра приложения A_j, и если ОЗУ сервера меньше γ_j, то экземпляр приложения A_j на этом сервере не может быть размещен.

Введение матрицы R (10.5) текущей совместимости сервера и приложения, а также вектора C (10.4) технологической возможности развертывания приложения в системе происходит аналогично модели без кластеризации.

Очередное отличие появляется при формулировании результата. Результатом поиска оптимального распределения экземпляров приложений по серверам хостингового ЦОД с кластеризацией является двумерный массив размерностью $n \times m$, элементы которой принимают значения из промежутка от 0 до 1 и характеризуют долю серверных ресурсов процессорной емкости, выделяемой под конкретный экземпляр приложения:

$$x_{ji} = \begin{cases} 0 < \ldots \le 1, \text{если экземпляр приложения } A_j \text{ запущен на сервере } N_i; \\ 0, \text{ в противном случае.} \end{cases} \quad (10.39)$$

Вводится дополнительный параметр, характеризующий факт наличия экземпляра определенного приложения на сервере:

$$x_{ji} = \begin{cases} 1, \text{ если } x_{ji} > 0 \\ 0, \text{ если } x_{ji} = 0. \end{cases} \quad (10.40)$$

Разделение основной задачи на подзадачу планирования распределения ресурсов и нагрузки и подзадачу диспетчеризации остается подобным варианту хостинга без кластеризации.

Модель планирования распределения нагрузки и ресурсов для задачи максимальной экономии ресурсов. Согласно изменениям, внесенных в базовую модель

серверов и приложений для варианта без кластеризации, изменения претерпели и формулировки основных ограничений (10.32)–(10.34):

$$\sum_{j=1}^{m} x_{ji} \cdot \gamma_j \le \Gamma_i, \quad i = \overline{1,n}. \tag{10.41}$$

Ограничение (10.41) требует, чтобы емкость ОЗУ сервера N_i была достаточной для размещения планируемых для него экземпляров приложений;

$$\sum_{i=1}^{n} x_{ji} \cdot \Omega_i \le \omega_j, \quad j = \overline{1,m}. \tag{10.42}$$

Ограничение (10.42) требует, чтобы потребности всех пользователей приложения A_j в процессорной емкости удовлетворяли в совокупности выбранные доли физических серверов с экземплярами этого приложения;

$$\sum_{i=1}^{n} x_{ji} \ge 1, \quad j = \overline{1,m}. \tag{10.43}$$

Ограничение (10.43) требует, чтобы был хотя бы один экземпляр каждого приложения на каком-то сервере.

Добавляется специфическое ограничение, связанное с организацией распределенных вычислений для нескольких экземпляров приложений в пределах одного сервера:

$$\sum_{j=1}^{m} x_{ji} \le 1, \quad i = \overline{1,n}. \tag{10.44}$$

Ограничение (10.44) требует, чтобы емкость процессоров сервера N_i была достаточной для размещения и эффективной работы всех экземпляров приложений в долях, планируемых для него.

Выражение для нахождения значения параметра, характеризующего наличие нагрузки на сервере y_i, $i = \overline{1,n}$ (10.36), также претерпело изменения. На основании очевидной связи между введенными переменными запишем следующее выражение:

$$y_i = x_{1i} \vee x_{2i} \vee \ldots \vee x_{mi}, \quad i = \overline{1,n}. \tag{10.45}$$

С использованием введенных обозначений критерий для этой задачи примет вид, аналогичный критерию варианта для хостинга без кластеризации (10.37):

$$\min \sum_{i=n}^{n} s_i y_i. \tag{10.46}$$

Аналогичным образом формулируется и критерий максимизации экономии (10.38).

Лучшее значение выбирается из решений, соответствующих структуре булевых значений матрицы R (10.5) и ресурсным ограничениям (10.41)–(10.44).

АлтГТУ

В разработанном программном продукте использованы математические модели планирования и распределения экземпляров приложений по серверам серверного парка хостинговой ЦОД для кластерной и без кластерной организации хостинга, т. е. для технологий с поддержкой распределенных процессов на группе серверов и без таковой поддержки.

Из возможных зависимых и независимых от нагрузки параметров сервера к рассмотрению были приняты следующие:

- зависимые от нагрузки: Ω_i – процессорная емкость *i*-го сервера;
- не зависимые от нагрузки: Γ_i – емкость ОЗУ.

Значительное внимание в разработанном программном продукте уделялось проверке выполнения физических ограничений, связанных с ресурсными мощностями серверов парка: (10.32)–(10.34) – для хостинга без кластеров и (10.41)–(10.44) – для формулирования задач управления ресурсами и нагрузкой при хостинге с кластеризацией.

Описание объектной модели программного продукта. Объектная модель, положенная в основу программного продукта, имеет целью воссоздание математической модели управления распределением нагрузки и ресурсов ЦОД при виртуальном хостинге.

Объектная модель разработанного программного продукта состоит из объектов – сущностей: Application, Server, Individual, Individual.Interaction и вспомогательных классов – сервисов, которые поставляют необходимый для выполнения этапов процессов функционал: Verification, Target Function, Data Generator, Genetic Factory, Performance Parameter, Awards, Logger.

Рассмотрим кратко объекты приведенной модели.

Объекты – сущности:

1. Application. Этот объект реализует математическую модель применения и имеет атрибуты, которые характеризуют основные потребности приложения к: ω_j – процессорной емкости серверов, на которых расположены экземпляры приложения A_j, $j=\overline{1,m}$ из A; γ_j – ОЗУ серверов, на которых расположены ВМ или экземпляры приложения A_j, $j=\overline{1,m}$.

Кроме набора параметров приложения данный класс сохраняет список всех приложений пользователей, а также поставляет сортировщики для списка по основным параметрам. Данный функционал используется в эвристических методах.

2. Server. Этот объект реализует математическую модель сервера серверного парка ЦОД и имеет атрибуты, которые характеризуют основные ресурсные мощности сервера: Ω_i – процессорная емкость сервера N_i; Γ_i – емкость ОЗУ сервера N_i;

Кроме набора параметров приложения, этот класс хранит список всех серверов ЦОД, а также поставляет сортировщики для списка по основным параметрам (функционал используется в эвристических методах) и набор сервисов для проверки текущего состояния сервера (свободные ресурсы, возможность размещения определенного приложения и т. д.).

3. Individual. Этот объект играет роль конечного результата или хромосомы при поиске оптимального решения с помощью ГА. В атрибутах объекта хранятся:

АИСТ РТУ

• значение целевой функции данной хромосомы при определенном критерии поиска;

• матрица X (10.7) объектов Individual.Interaction представляет собой фактическое значение хромосомы.

4. Individual.Interaction. Этот объект играет роль носителя значения конкретного развертыванию определенного приложения на определенном сервере. Основная роль – согласование на уровне объектной модели ЦОД без кластеризации и с кластеризацией вариантов реализации математической модели управления.

Атрибуты объекта содержат дискретные и непрерывные коэффициенты развертывания.

Рассмотрим классы-сервисы:

1. Verificator. Класс-сервис, который предоставляет функционал для проверки ограничений (10.32)–(10.34) и (10.41)–(10.44).

2. Target Function. Класс-сервис, который позволяет вычислять значения целевой функции для определенной хромосомы по каждому из описанных в данной работе критериев.

3. Data Generator. Класс-сервис, который поставляет функционал для начальной генерации стартовой популяции на основе случайных значений и при условии выполнения ограничений (10.32)–(10.34) и (10.41)–(10.44).

4. Genetic Factory. Класс-сервис, который поставляет функционал для реализации основных операций ГА – кроссовер, мутация, селекция.

Способы реализации операторов разработаны ниже.

5. Logger. Класс-сервис, который поставляет функционал для ведения наблюдения за ГА в текущем режиме путем вывода отчета процесса в специально выделенные текстовые поля.

6. Performance Parameter. Класс-сервис, позволяющий вычислять параметр производительности для каждого из операторов и проводить сравнение между значениями параметрами производительности. Содержит правила сравнения параметров.

7. Awards. Класс-сервис рассчитывает награды на этапе работы ОГА и сравнивает их на этапе работы ГГА.

Особенности интерфейса и процесса моделирования работы алгоритмов. Разработанный программный продукт имеет интерфейс, который обеспечивает ряд функций, необходимых для выполнения экспериментов.

Стартовое окно программы приведено на рис. 10.11.

Ознакомимся с вспомогательными окнами программного продукта, после чего вернемся к рассмотрению основного функционала.

Первые из вспомогательных окон вызываются нажатием кнопки «Работа с данными». Функционал программного продукта, который доступен через это окно, состоит в обеспечении пользователя возможностью видоизменять исходные данные, используемые для работы как эвристических, так и генетических алгоритмов.

Вкладка «Сервера» (рис. 10.12) позволяет создавать новые серверы, модифицировать уже существующие или удалять из списка серверов ненужные объекты. С точки зрения моделирования эта вкладка имитирует администрирование хостингового ЦОД на предмет конфигурации серверного парка и его состава.

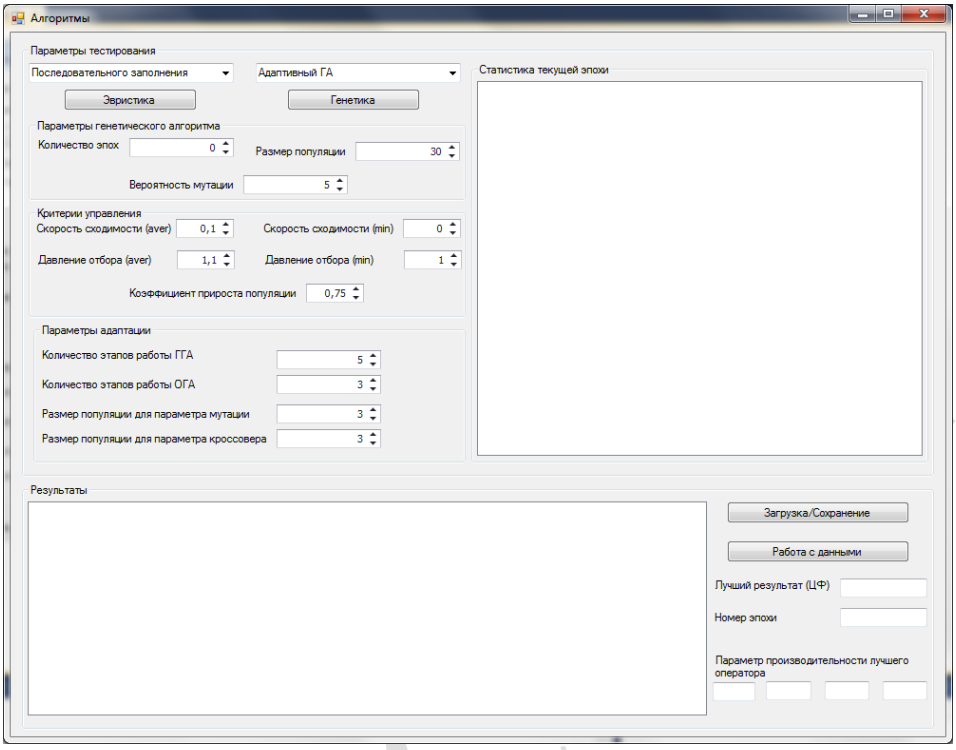


Рис. 10.11 Стартовое окно разработанного программного продукта

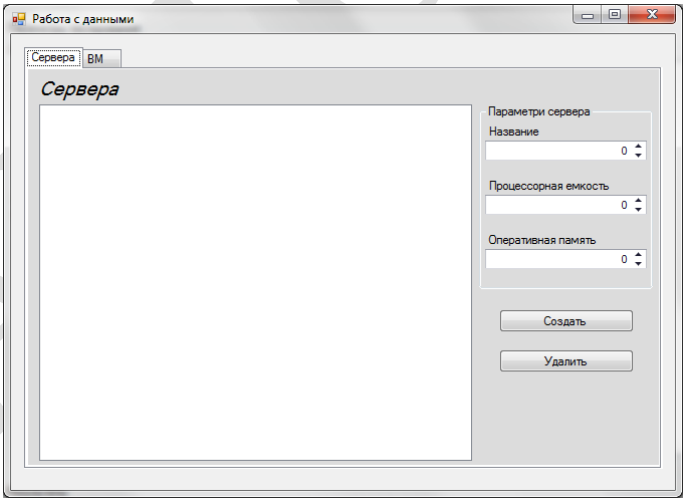


Рис. 10.12. Окно «Работа с данными»: Работа с серверами

Вкладка «ВМ» (рис. 10.13) позволяет создавать новые приложения, модифицировать уже существующие или удалять из списка приложений ненужные объекты. С точки зрения моделирования эта вкладка имитирует внешнюю нагрузку

на сервер и позволяет конфигурировать необходимые условия для проведения планирования распределения ресурсов между приложениями.

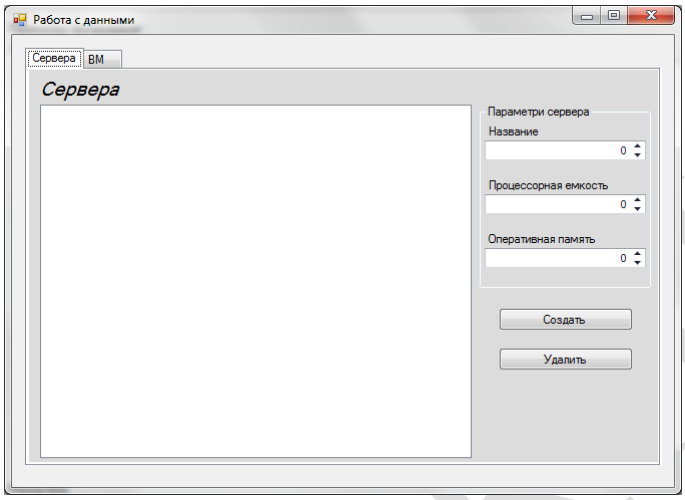


Рис.10.13. Окно «Работа с данными»: Работа с приложениями

Второе из вспомогательных окон открывается нажатием кнопки «Загрузка/Сохранение» (рис. 10.14). Функционал этого окна позволяет сохранять текущие объекты в файл данных или загружать из него определенное количество объектов каждого типа.

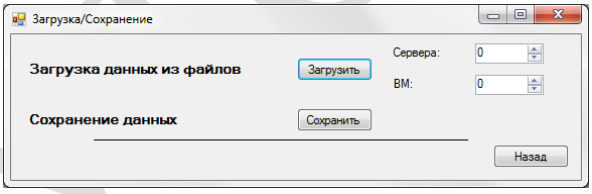


Рис. 10.14. Окно «Загрузка/Сохранение»

Рассмотрим более подробно основной функционал ПО, а именно функционал окна, изображенного на рис. 10.11. Это окно предоставляет следующие функциональные возможности:

- выбирать один из эвристических алгоритмов (последовательного заполнения, типа «Т», типа «М»);
- выбирать вариант генетического алгоритма (базовый, управляемый, адаптивный);
- регулировать параметры ГА, такие как количество эпох, вероятность мутации, размер популяции, параметры управления ГА, параметры АГА;
- запускать на моделирование эвристический и генетический алгоритмы;
- получать детальный отчет о результатах эксперимента.

ПО позволило провести опыты для сравнения эффективности работы каждой из эвристик и вариантов ГА.

10.6.5. Результаты экспериментальных исследований

Промоделируем применение предлагаемого АГА для решения задачи (10.1)–(10.47). Здесь основным понятиям, введенным для формулирования математической модели задачи, соответствуют выражения (10.1)–(10.6), ограничениям – выражения (10.9)–(10.12), критерий оптимальности задается выражением (10.13), а ЦФ – выражениями (10.14), (10.15). По содержанию задача распределения множества *K* ВМ формулируется следующим образом: необходимо разместить ВМ на серверах так, чтобы выражение (10.14) или (10.15) достигало минимального значения.

Проведены экспериментальные исследования решения задачи с помощью трех алгоритмов: КГА, УГА и АГА.

Модель экспериментального исследования АГА приведена на рис. 10.15.

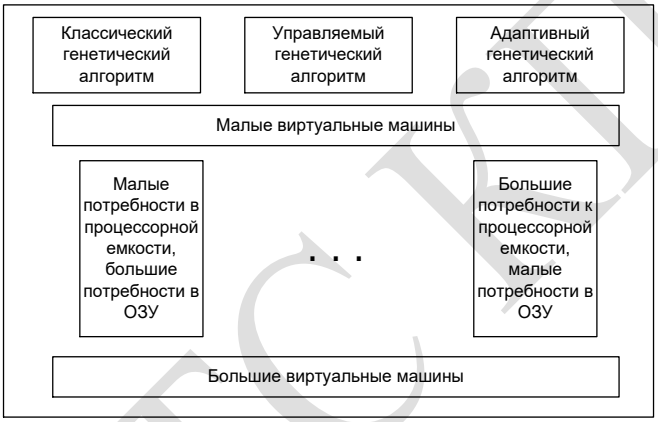


Рис. 10.15. Модель исследования АГА

При исследованиях рассматривались три варианта соотношений ресурсов, наиболее приближенные к реальным ситуациям:

- случай непропорциональных потребностей в ресурсах, когда в относительных единицах запрашиваемая величина процессорного времени (ЦП) значительно превышает запрашиваемый объем ОЗУ, т. е. $\omega_j >> \gamma_j$, $j = \overline{1,m}$. Подобный тип задачи возникает при наличии большого количества приложений, требующих сложный вычислений;
- случай непропорциональных потребностей в ресурсах, когда запрашиваемая величина ЦП значительно меньше запрашиваемого объема ОЗУ, т. е. $\omega_j << \gamma_j$, $j = \overline{1,m}$. Такая задача возникает, когда на серверах ЦОД размещаются ВМ с приложениями, требующими обработки больших объемов данных;
- объемы запрашиваемых ресурсов ЦП и ОЗУ распределены случайным образом в диапазоне [0,05; 0,6].

Пределы изменения запрашиваемых ресурсов ВМ для различных экспериментов приведены в табл. 10.2.

В работе [161] доказано, что при количестве ВМ меньше пятидесяти эвристические и генетические алгоритмы дают приблизительно одинаковые результаты, а с увеличением количества ВМ выигрыш ГА становится все более ощутимым.

Таблица 10.2. Результаты исследования АГА

№ эксперимента	Ограничения, ОЗУ	Ограничения, ЦП
1	0,3–0,45	0,05–0,15
2	0,45–0,6	0,05–0,15
3	0,05–0,15	0,3–0,45
4	0,05–0,15	0,45–0,6
5	0,05–0,6	0,05–0,6

Результаты работы КГА, УГА и АГА оцениваются по критерию количества высвобожденных серверов. Предполагается, что изначально для развертывания каждой ВМ выделяется отдельный сервер. Далее с помощью исследуемых алгоритмов проводится оптимизация размещения ВМ на серверах с оценкой максимального количества высвобожденных серверов для каждого из алгоритмов. На рис. 10.16 показан график зависимости количества высвобожденных серверов от размерности задачи для случая, когда объемы запрашиваемых ресурсов ЦП и ОЗУ для всех ВМ распределены случайным образом в диапазоне [0,05; 0,6].

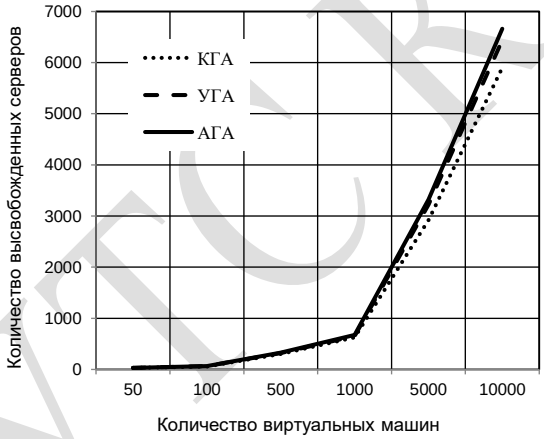


Рис. 10.16. Зависимость количества высвобожденных серверов от размерности задачи

Для сравнения результатов работы УГА и АГА вводится понятие дополнительно высвобожденных серверов $N_В$. Значение $N_В$ определяется как разность между количеством серверов $N_{КГА}$, высвобожденных в результате работы классического ГА, и количеством серверов $N_{УГА}$ и $N_{АГА}$, высвобожденных в результате применения УГА и АГА соответственно. Таким образом, на графиках рис. 10.17 показан выигрыш УГА и АГА относительно КГА в виде зависимости количества дополнительно высвобожденных серверов $N_В$ от размерности задачи для различных вариантов соотношений запрашиваемых ресурсов. Результаты экспериментов приведены на рис. 10.17.

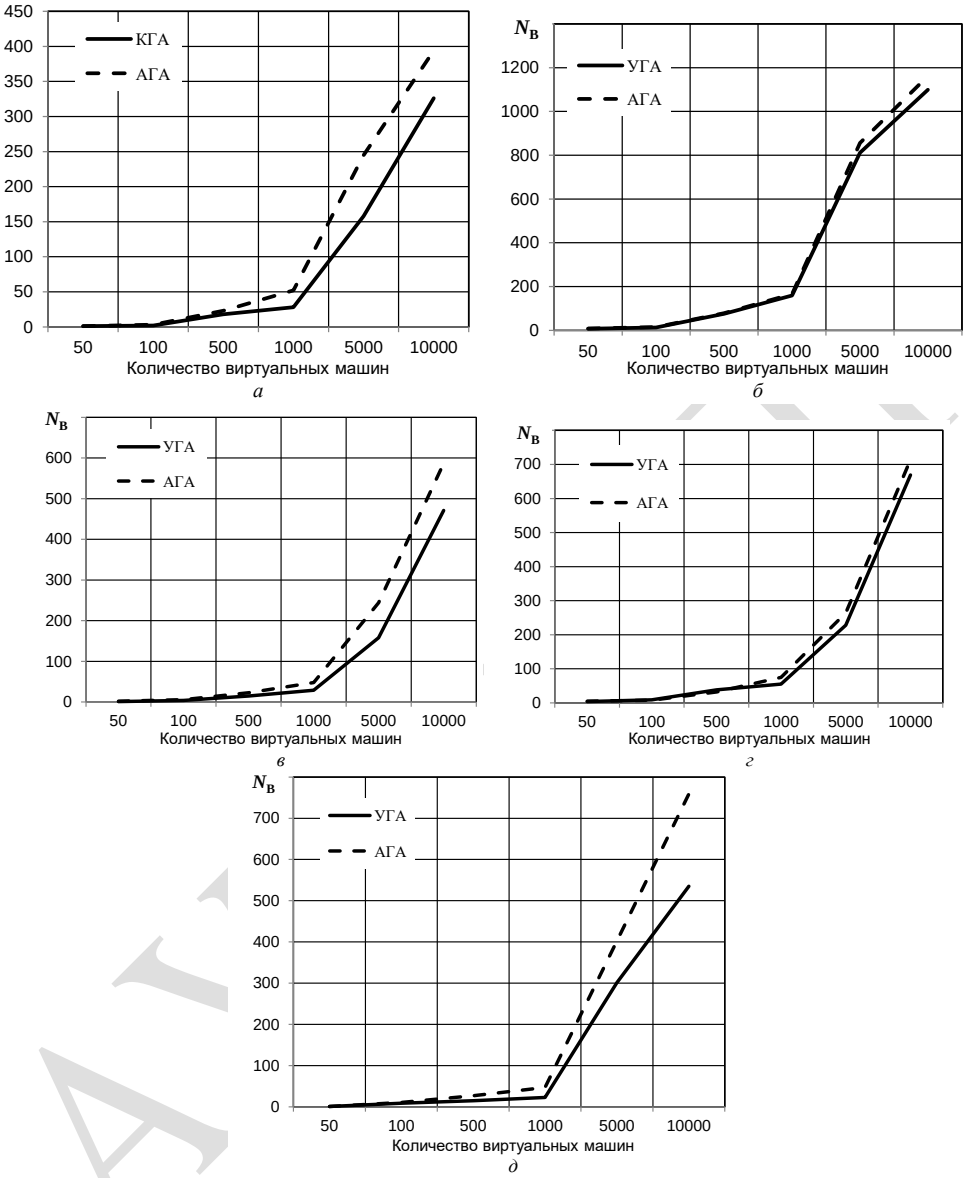


Рис. 10.17. Зависимость количества дополнительно высвобожденных серверов от размерности задачи, когда требования к запрашиваемым ресурсам лежат в диапазонах: *а* – к ЦП – [0,05; 0,15], к ОЗУ – [0,3; 0,45]; *б* – к ЦП – [0,05; 0,15], к ОЗУ – [0,45; 0,6]; *в* – к ЦП – [0,05; 0,15], к ОЗУ – [0,3; 0.45]; *г* – к ЦП – [0,05; 0,15], к ОЗУ – [0,45; 0,6]; *д* – к ЦП – [0,05; 0,6], к ОЗУ – [0,05; 0,6]

Анализ графиков рис. 10.17 позволяет сделать следующие выводы:

- использование УГА и АГА эффективнее, чем использование классического ГА;
- предложенный АГА доминирует независимо от условий эксперимента;

• при наличии ВМ с большими требованиями к ОЗУ либо ЦП (рис. 10.17,*б*, и рис. 10.17,*г*) эффективность использования УГА и АГА выравнивается, однако преимущество остается за АГА;

• для случая разброса требований в широком диапазоне [0,05; 0,6] (рис. 10.17,*д*) использование АГА наиболее эффективно.

Таким образом, предложен класс адаптивных генетических алгоритмов, основанный на циклическом применении двух ГА, решающих задачи параметрической и алгоритмической адаптации, что позволяет определить стратегию выбора значений параметров операторов ГА и вероятность их применения в процессе эволюции ГА. Отличием предложенного адаптивного ГА от подобных алгоритмов является применение как параметрической, так и алгоритмической адаптации. Введены понятия модифицированных операторов кроссовера и мутации, параметра производительности генетических операторов. Сформированы правила сравнения операторов, позволившие регламентировать последовательности применения генетических операторов и корректировать вероятность их применения. На примере задачи распределения ресурсов ЦОД доказана работоспособность и эффективность предлагаемого адаптивного ГА, его преимущества по сравнению с классическим и управляемым вариантами генетического алгоритма.

10.7. АЛГОРИТМ ПОСЛЕДОВАТЕЛЬНОГО АНАЛИЗА ВАРИАНТОВ В ЗАДАЧЕ РАСПРЕДЕЛЕНИЯ ВИРТУАЛЬНЫХ МАШИН

10.7.1. Постановка задачи исследования

Эффективность рассмотренных выше эвристических алгоритмов зависит от структуры алгоритма и параметров размещаемых ВМ или приложений. Эвристические алгоритмы не гарантируют нахождения оптимального решения задачи размещения ВМ или приложений. Одним из алгоритмов, обеспечивающих оптимальный результат, является алгоритм последовательного анализа вариантов (ПАВ). Поскольку область его применения в ИТ УИТИ не определена, возникает задача разработки и исследования эффективности применения новых модификаций ПАВ для решения задач размещения ВМ или приложений на серверах в ЦОД. Это позволит разработать рекомендации по использованию ПАВ для важного класса задач.

Рассмотрим задачу распределения ВМ в ЦОД (10.1)–(10.48). Напомним, что здесь для формулирования математической модели введены понятия, которым соответствуют выражения (10.1)–(10.6), ограничениям – выражения (10.9)–(10.12), критерий оптимальности задается выражением (10.13), а целевая функция – выражением (10.14), (10.15). Необходимо разработать ПАВ для размещения ВМ так, чтобы выражение (10.14) или (10.15) достигало минимума.

10.7.2. Алгоритм последовательного анализа вариантов для распределения виртуальных машин

Алгоритм ПАВ относится к комбинаторным методам решения задач [133], основная идея которых состоит в использовании конечности множества допустимых решений и замене их полного перебора сокращенным – направленным перебором. Общая схема ПАВ разработана В.С. Михалевичем [74]. Методы ПАВ основаны на последовательном уточнении значений компонент решения и отсеве в процессе конструирования решений, которые не могут быть достроены до оптимальных.

Пусть имеется задача минимизации:

$$\min f(x_1,...,x_m) = \min s\sum_{i=1}^{n} y_i, \tag{10.49}$$

при ограничениях вида

$$\left(\sum_{j=1}^{s}(r_{j1}\omega_j)y_1,...,\sum_{j=1}^{s}(r_{ji}\omega_j)y_n\right) \le 1, \quad j=\overline{1,m}, \quad i=\overline{1,n}, \tag{10.50}$$

$$\left(\sum_{j=1}^{s}(r_{j1}\gamma_j)y_1,...,\sum_{j=1}^{s}(r_{ji}\gamma_j)y_n\right) \le 1, \quad j=\overline{1,m}, \quad i=\overline{1,n}, \tag{10.51}$$

$$x_j \in Q, \quad j=\overline{1,m}, \tag{10.52}$$

где $Q=\left\{q_{1_1 j},...,q_{\alpha_m j}\right\}$ – заданные конечные множества [134]. Каждая переменная $x=(j;i)$ для всех $j=\overline{1,m}$ выражает размещение *j*-й ВМ на *i*-м сервере.

Вектор $\overline{x}_{(m)}=\left(x_1,...,x_m\right)$ является решением, если его компоненты $x_j \in Q$, $j=\overline{1,m}$. Каждый элемент множества $q_{\alpha_m i}$ представляет собой одну из возможных компонент вектора $\overline{x}_{(m)}$.

Множество всех решений обозначим через *Z*. Решение называется допустимым, если оно удовлетворяет неравенствам (10.50) и (10.51). Множество всех допустимых решений обозначаем через Z_f. Вектор $\overline{x}_{(p)}=\left(x_1,...,x_p\right)$, $p < m$ называется частичным решением [134], если $\overline{x}_{(p)} \in Q$. Если при этом вектор $\overline{x}_{(p)}$ может быть достроен до допустимого решения $\left(x_1,...,x_p,x_{p+1},...,x_m\right)$, $\overline{x}_{(m)} \in Z_f$, то он называется допустимым частичным решением.

Множество всех решений задачи изображается в виде дерева, где пути от корня к вершинам дерева соответствуют частичным решениям, а пути из вершины x_0 в вершины x_m соответствуют полным решениям (рис. 10.18).

АУГСКП

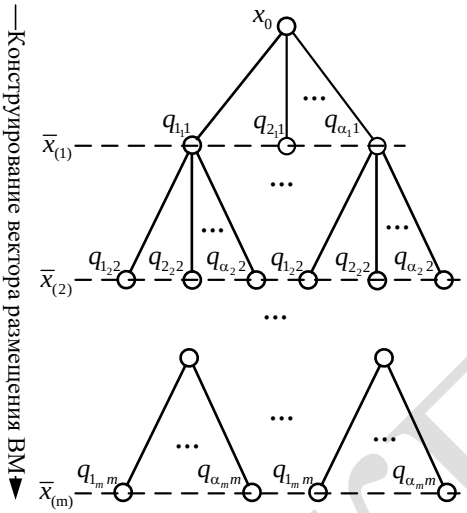


Рис. 10.18. Дерево решений для алгоритма ПАВ

Для задачи, когда все серверы одинаковы, целесообразно исключить из рассмотрения зеркальные решения, поскольку важно не абсолютное размещение ВМ на этих серверах, а их расположение ВМ относительно друг друга (рис. 10.19). При увеличении количества серверов количество зеркальных комбинаций значительно возрастает от $(m!-1)$ для случая размещения всех ВМ на одном сервере до $(m\times n-1)!$ при размещении ВМ на разных серверах.

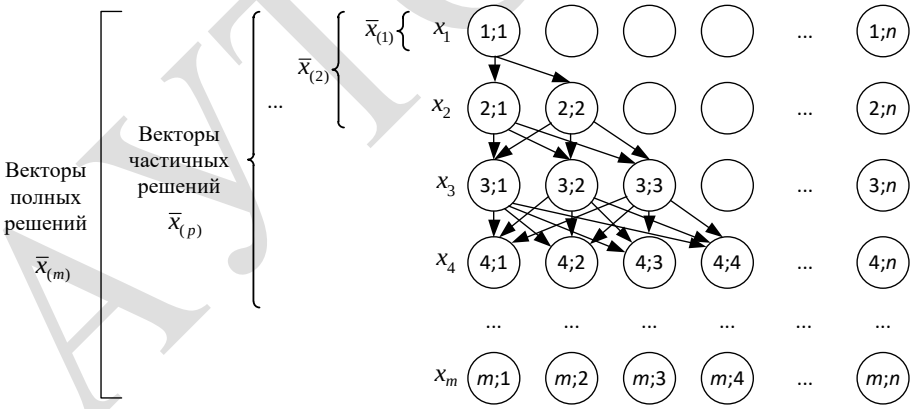


Рис. 10.19. Размещение ВМ на серверах с помощью алгоритма ПАВ

Необходимо определить набор правил выбора решений, которые будут развиваться на каждом шаге. Правила задаются в виде элиминирующих тестов, определяющих стратегию развития частичных решений. Определение набора $\xi=\{\xi_l\}$ элиминирующих тестов необходимо для отсева частичных решений, которые не могут быть достроены ни до допустимых, ни до оптимальных [134].

АУТСКП

Пусть h – некоторое множество частичных решений. Тогда в упорядоченном множестве $\sigma(h) = \{\xi_0(h), \xi_1(h), ..., \xi_l(h)\}$ через $\xi_l(h)$ обозначается множество частичных решений, исключаемых тестом ξ_l.

Введем следующий набор элиминирующих тестов: ξ_0 – анализ допустимости решений; ξ_1 – сравнение допустимых решений по значению ЦФ; ξ_2 – оценка оптимального значения γ_f ЦФ на каждом из множеств (вычисление нижней границы).

Тест ξ_0 позволяет исключить из рассмотрения недопустимые решения. Тест ξ_1 призван находить оптимальное распределение ВМ и исключать из рассмотрения неоптимальные допустимые решения. Тест ξ_2 и применение правила отсеивания зеркальных решений позволяют избежать полного перебора допустимых решений.

Определим элиминирующие тесты ξ_0, ξ_1, ξ_2 следующим образом:

$$\xi_0(h) = \{\bar{x}_{(p)} \mid \bar{x}_{(p)} \in h,\ \sum_{j=1}^{p} x_{ji}\omega_j > 1,\ \sum_{j=1}^{p} x_{ji}\gamma_j > 1,\ \overline{i = 1, n}\}, \tag{10.53}$$

$$\xi_1(h) = \{\bar{x}_{(m)} \mid \bar{x}_{(m)} \in Z_f, f(x) \le f(\bar{x}_{(m)})\}, \tag{10.54}$$

$$\xi_2(h) = \{\bar{x}_{(p)} \mid \bar{x}_{(p)} \in h, \gamma_f(\bar{x}_{(p)}) > f^*\}, \tag{10.55}$$

где f^* – верхняя граница минимального значения (10.49). Тест ξ_2 вычисляет нижнюю границу для оптимального значения ЦФ $f(\bar{x}^0_{(p)})$ на каждом из множеств $\mathrm{H}(\bar{x}^0_{(p)})$, элементами которых являются решения, продолжающие частичное решение $\bar{x}^0_{(p)}$.

На результаты ПАВ оказывает влияние порядок применения элиминирующих тестов и правил $U = (u_1, ..., u_q)$ выбора частичных решений. В алгоритме ПАВ для задачи распределения ВМ множество правил U задается правилами u_1, u_2, u_3.

Правило u_1. На каждом шаге алгоритма развиваются все частичные решения, построенные на предыдущем шаге, кроме решений, которые являются зеркальными отражениями уже рассмотренных решений.

Правило u_2. На каждом шаге из частичных решений, полученных на предыдущих шагах, развиваются те, для которых достигается минимум ЦФ.

Правило u_3. Выбор кандидата на каждом шаге схемы осуществляется из множества $h_d = h_k - h_{k-1}$ частичных решений, полученных на предыдущем шаге, и только, когда h_d не содержит частичных решений, необходимо обратиться ко всему списку.

Множества правил $U = (u_1, u_2, u_3)$ и тестов $\xi = \{\xi_0, \xi_1, \xi_2\}$ определяют стратегию перебора решений. Порядок применения правил и элиминирующих тестов соответствует индексам. Правила и тесты упорядочены по возрастанию сложности.

10.7.3. Результаты экспериментальных исследований алгоритма последовательного анализа вариантов

Количество запрашиваемых ресурсов для ВМ выбирается случайным образом. Соотношение требуемой ОЗУ и процессорного времени распределено равномерно.

При проведении экспериментов генерировалось задаваемое количество множеств ВМ. Потребности ВМ в вычислительных ресурсах по оперативной памяти и процессорному времени сервера генерировались случайным образом в пределах 0,05–0,6 от возможностей сервера.

Эффективность работы алгоритма ПАВ оценивалась по времени, необходимому для нахождения оптимального решения в зависимости от размерности задачи *m*. Результаты работы алгоритма приведены на рис. 10.20. Для сравнения показано время работы УГА, а также эвристических алгоритмов «Т» и «М».

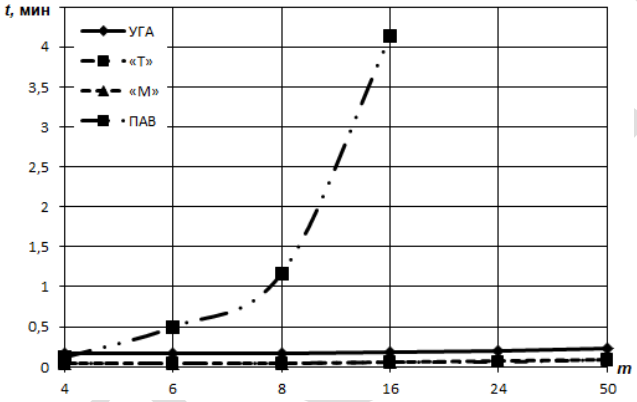


Рис. 10.20. Зависимость времени работы алгоритмов от размерности задачи

На рис. 10.20 видно, что алгоритм ПАВ показывает приемлемые результаты времени размещения на задачах малой размерности, когда количество ВМ не превышает 4–16. При дальнейшем увеличении количества ВМ время работы алгоритмом ПАВ возрастает по экспоненциальной зависимости, в то время как эвристические и ГА демонстрируют линейное увеличение затрат времени. Большое различие во времени работы алгоритма ПАВ и времени эвристических алгоритмов объясняется тем, что ПАВ является комбинаторным алгоритмом, а эвристические алгоритмы – модификацией сортировочных алгоритмов.

Проведен анализ погрешности работы алгоритмов. Значение ЦФ S_T, S_M, $S_{УГА}$, полученное в результате работы эвристических алгоритмов «Т», «М» и УГА соответственно, сравнивалось с оптимальным решением, обеспечиваемым ПАВ. На рис. 10.21 по оси ординат отложена ошибка распределения ВМ, которая показывает, насколько больше серверов задействуется при распределении ВМ по алгоритмам «Т», «М» и УГА, чем при оптимальном размещении. По оси абсцисс откладывается количество ВМ, подлежащих распределению на серверах ЦОД.

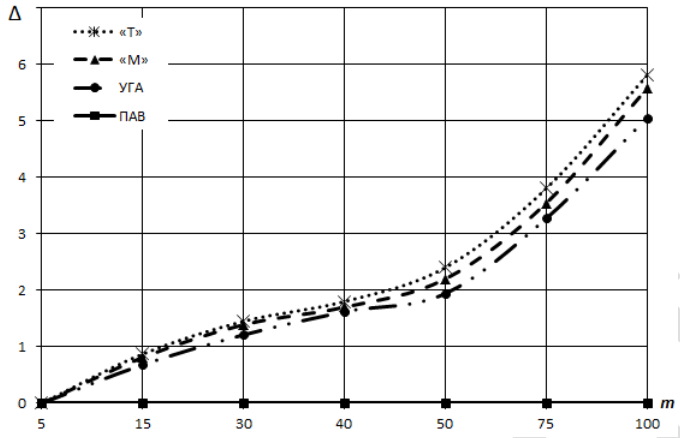


Рис. 10.21. Зависимость ошибки распределения ВМ Δ относительно оптимального решения от размерности задачи

Алгоритм ПАВ дает незначительный выигрыш по точности даже на задачах малой размерности. При увеличении размерности ошибка результатов эвристических и УГА возрастает, что свидетельствует о зависимости погрешности результатов работы этих алгоритмов от размерности задачи.

Для применения алгоритма ПАВ необходимо располагать значением количества серверов $N_{\min}$, для которых точно известно, что на них могут быть размещены все ВМ из набора K_j, $\overline{j=1,m}$. Значение $N_{\min}$ может быть получено экспертным путем или с помощью одного из эвристических алгоритмов.

Результаты исследований позволили сделать следующие рекомендации по выбору алгоритма. Для задач, где необходима высокая точность распределения при малом количестве ВМ, необходимо использовать алгоритм ПАВ. ПАВ можно применять и тогда, когда время получения результата не является критичным. Скорость работы эвристических и эволюционных алгоритмов мало зависит от размерности задачи, однако качество решения уступает алгоритму ПАВ.

Таким образом, разработан алгоритм дискретной оптимизации на основе схемы ПАВ для решения задачи размещения ВМ по серверам ЦОД. Показано, что эвристические алгоритмы позволяют быстро получать решение, благодаря которому можно снизить затраты на поддержание серверов. Алгоритм ПАВ требует существенных затрат времени, но позволяет находить наилучшие решения.

ВЫВОДЫ К ГЛАВЕ 10

1. Разработаны эвристические алгоритмы типа «Т» и «М» решения задачи распределения ВМ в ЦОД с учетом существующих ограничений. Показано, что эвристические алгоритмы позволяют получать решения быстрее, чем ГА, но с меньшим приближением к оптимуму. Кроме эвристических методов предложены три варианта подхода к реализации генетического метода поиска: классический ГА, УГА и АГА.

2. Разработан класс управляемых генетических алгоритмов, отличающийся получением стабильных популяций на каждой из эпох, что не позволяет скатиться к локальному оптимуму. Алгоритмы класса управляют последовательностью процессов в ходе решения задачи на основе предложенной системы правил и условий применения операторов кроссовера и мутации, обеспечивая высокий уровень качества получаемого решения и достаточную скорость сходимости процессов поиска.

3. Предложен класс адаптивных генетических алгоритмов, основанный на циклическом применении двух генетических алгоритмов, решающих задачи параметрической и алгоритмической адаптации, что позволяет определить стратегию выбора значений параметров операторов генетических алгоритмов и вероятность их применения в процессе эволюции ГА. Алгоритмы предложенного класса отличаются одновременным применением параметрической и алгоритмической адаптации, что позволяет за равное количество эпох получать лучшие результаты по сравнению с классическим и управляемым вариантами генетического алгоритма. Главное отличие АГА от УГА состоит в полной автономности работы и возможности адаптации к решаемой задаче и условиям протекания процесса решения задачи.

4. Предложена реализация метода последовательного анализа вариантов для размещения ВМ в ЦОД, который, в отличие от ГА, позволяет получить оптимальный вариант решения задачи, однако время работы алгоритма существенно увеличивается при увеличении размерности задач, что позволяет рекомендовать его только для размещения ВМ, количество которых не более двадцати.

5. Проведены исследования эффективности эвристических алгоритмов, управляемых и адаптивных ГА для задач распределения нагрузки и ресурсов ЦОД в зависимости от характеристик серверов, приложений и запросов пользователей. На основе этих исследований определены области, где проявляются преимущества и недостатки этих алгоритмов.

6. Проведены исследования сравнительной эффективности эвристических алгоритмов, управляемых и адаптивных ГА с алгоритмами ПАВ, что позволило определить условия применения последних.

7. Разработаны рекомендации по применению эвристических алгоритмов, алгоритмов классов УГА и АГА, алгоритмов ПАВ для различных вариантов задач распределения ресурсов с учетом особенностей ресурсов, приложений и запросов пользователей.

Антиплагиат

ГЛАВА 11

РАЗРАБОТКА И РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ И СИСТЕМЫ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ

На основе предложенных в предыдущих главах моделей и методов разрабатывается архитектура и принципы функционирования ИТ УИТИ. Рассматриваются технические и технологические аспекты реализации ИТ УИТИ и разработки на ее основе СУИ SmartBase.ITS.Control.

11.1. КОНЦЕПТУАЛЬНЫЕ И МЕТОДИЧЕСКИЕ ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ И ВНЕДРЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ

11.1.1. Значимость системы управления ИТ-инфраструктурой

В настоящее время показатели эффективности работы практически всех предприятий и организаций существенно зависят от эффективности функционирования корпоративных ИТ-инфраструктур, которые представляет собой совокупность информационной системы и телекоммуникационной сети и являются базисом ИУС предприятий или организаций. ИУС предоставляет средства для автоматизации бизнес-процессов, обеспечивает информационное и телекоммуникационное взаимодействие между функциональными и обеспечивающими подсистемами ИУС, подразделениями, сотрудниками и др.

Для управления корпоративными ИТ-инфраструктурами используются универсальные проприетарные или разрабатываются собственные адаптированные под потребности предприятия СУИ [120, 154, 155]. СУИ представляет собой специализированную ИУС, часть ИУС предприятия (корпорации, организации), предназначенную для управления ИТ-инфраструктурой. Ее создание требует разработки системного решения, видов обеспечения (алгоритмического, программного, технического, информационного), реализацию соответствующих этапов и стадий жизненного цикла, связанных с анализом, проектированием, реализацией и внедрением системы. В этой главе основное внимание уделено разработке программного обеспеченья СУИ, реализующей комплекс функций, связанных с мониторингом состояния и анализом работоспособности как отдельных элементов ИТ-инфраструктур и ИУС, так и подсистем их управления. Как организационно-техническая система СУИ проводит мониторинг сетевого оборудования и решение задач сетевого управления, проверку функционирования и контроль работоспособности сетевых сервисов и приложений, управление аппаратными и программными элементами ИУС и реализацию других функций.

Увеличение стоимости информационно-телекоммуникационных ресурсов и технологий, существенная зависимость бизнеса от эффективности работы ИУС, постоянная модернизация и появление новых ИТ, ужесточение требований к эф-

фективности и бесперебойности работы ИУС обуславливают актуальность задачи совершенствования существующих или разработки новых перспективных СУИ.

В то же время эффективность функционирования ИТ-инфраструктуры может быть обеспечена только путем комплексного взаимосвязанного управления всеми иерархическими уровнями ИТ-инфраструктуры. При всем разнообразии предприятий, организаций, корпораций, их ИТ-инфраструктура имеет очень много однотипных компонентов и зачастую создается на основе одного решения. Поэтому реализацию функций СУИ можно осуществлять на основе единого комплекса моделей, методов и средств, параметры в критериях, ограничениях которых и сам выбор критериев и ограничений позволяют учесть особенности конкретной ИТ-инфраструктуры. Эту общность задач управления целесообразно реализовать в универсальной ИТ управления ИТ-инфраструктурой. Эту ИТ ИУТИ можно использовать как «кирпичик» при создании, эксплуатации и развитии специализированных СУИ. При создании ИТ ИУТИ и интегрированной системы управления ИТ-инфраструктурой необходимо сделать упор на решение вопросов управления, которые недостаточно проработаны международными стандартами. Для решения этой задачи необходимо прежде всего определить требования к концепции управления ИТ-инфраструктурой, после чего создать концепцию управления как систему путей решения поставленной задачи. Эти задачи решены во второй и последующих главах данной работы.

В работе [154] определены требования, которым должна соответствовать перспективная СУИ. Разработана структура серверной части, но не предложены принцип построения агентов и схема взаимодействия серверной и агентской частей СУИ. В [123] методы мониторинга элементов ИТ-инфраструктуры рассматриваются без привязки к структуре СУИ. В [117] рассматривается агентская схема построения СУИ, определена концепция мобильных агентов и мета-агентов, однако отсутствуют принципы построения и структура агентов, а также не представлены механизмы взаимодействия агентов с сервером. Работа [176] посвящена интеграции методов активного и пассивного мониторинга элементов коммуникационной сети в GRID системах. В ней также не рассматриваются принципы построения агентов и вопросы их реализации.

Целью главы является разработка СУИ на основе агентского подхода с интеграцией активного и пассивного мониторинга и консолидацией методов централизованного и децентрализованного управления.

11.1.2. Основные задачи и функции систем управления ИТ-инфраструктурой

Современные СУИ представляют собой организационно-аппаратно-программные комплексы, предназначенные для мониторинга состояния, анализа работоспособности и управления аппаратными и программными средствами ИТ-инфраструктуры [8, 44, 75, 120, 154, 236, 267, 334, 356, 357, 358, 360, 376, 389].

Основные задачи и функции СУИ заключаются в следующем:

- непрерывный мониторинг и анализ функционирования элементов ИТ-инфраструктуры, включая контроль технического состояния и работоспособности аппаратного и программного обеспечения серверов, средств подсистем, сети и рабочих станций;
- накопление и обработка данных мониторинга, анализ данных о функционировании и отображение результатов контроля состояния и работоспособности

элементов ИТ-инфраструктуры, функциональных и технологических подсистем ИУС и распределенных приложений;

• дистанционное управление программно-техническими средствами ИТ-инфраструктуры;

• консолидация информации о функционировании ИТ-инфраструктуры с последующим анализом для проактивного управления, планирования развития ИТ-инфраструктуры;

• осуществление централизованного управления с децентрализованным выполнением некоторых функций управления;

• организационное, автоматизированное и автоматическое управление функционированием программно-технических элементов ИТ-инфраструктуры, программного обеспечения функциональных и технологических подсистем ИУС;

• контроль и анализ качества предоставляемых сервисов;

• интеграция средств администрирования приложений, функциональных и технологических подсистем;

• рациональное распределение ресурсов ИТ-инфраструктуры между бизнес-процессами с учетом их значимости;

• регистрация и анализ событий в ИТ-инфраструктуре;

• прогнозирование критических состояний ИТ-инфраструктуры и предотвращение возникновения отказов в ее работе на ранней стадии;

• своевременное обнаружение, локализация и управление устранением неисправностей в ИТ-инфраструктуре и ИУС;

• контроль изменений в ИТ-инфраструктуре, в том числе несанкционированных;

• контроль технического состояния и работоспособности, как отдельных узлов, так и всей телекоммуникационной сети, анализ трафика, контроль абонентских подключений, запуска сетевых сервисов, в том числе несанкционированных;

• выработка команд и сигналов управления техническими и программными средствами подсистем в автоматическом и ручном режиме для принятия решения системным администратором;

• документирование действий администраторов по управлению функциональными и технологическими подсистемами;

• автоматизация планирования деятельности персонала при техническом обслуживании и других регламентных работах с учетом загруженности ресурсов ИТ-инфраструктуры;

• автоматизация управления операциями: разработка и планирование, ввод в эксплуатацию, обслуживание и техническая поддержка компонентов ИУС.

Для решения этих и множества других задач в СУИ используются централизованное или децентрализованное управление, активный или пассивный мониторинг. Как правило, СУИ, построенная по централизованному принципу, использует метод активного мониторинга, когда сервер СУИ получает информацию от элементов ИУС путем запроса, и метод непосредственного управления элементами путем отправки соответствующих команд. Децентрализованная СУИ реализуется на интеллектуальных агентах, при этом основная часть задач мониторинга и управления реализуется агентом, а сервер собирает информацию о событиях, генерируемых агентами, и вырабатывает политику управления, обязательную для всех агентов.

11.1.3. Принципы построения перспективных систем управления ИТ-инфраструктурой

В работах [73, 106, 120, 154, 166] определены требования к свойствам, принципам построения и проектирования СУИ.

Основные принципы построения СУИ в самом общем виде можно свести к следующему:

- организационно СУИ больших корпоративных ИТ-инфраструктур должна представлять собой территориально-распределенную иерархическую структуру. Топология СУИ в пределах зоны ответственности отдельного административного региона, размещение узлов управления, число уровней иерархии СУИ определяются с учетом организационной структуры подразделения управления ИТ-инфраструктурой;
- вместо совокупности отдельных фирменных и специализированных СУ, осуществляющих раздельное управление разнообразными информационными, телекоммуникационными и компьютерными технологиями, целесообразно наличие единой интегрированной системы управления, включающей в себя единую базу: единую базу данных ОМУ, содержащую параметры конфигурации, данные мониторинга и другую информацию, исключающую дублирование и обеспечивающую целостность информации, а также ее доступность для различных приложений СУИ; модули сопряжения и взаимодействия с фирменными системами управления оборудованием коммуникационных и информационных составляющих ИУС. СУИ должна выступать интегратором этих СУ, получая от них обобщенную информацию о состоянии ОМУ и выдавая им макрокоманды, вырабатываемые с учетом приоритетов бизнеса, известных СУИ и неизвестных СУ. Макрокоманды предназначены для координации взаимодействия отдельных СУ с целью эффективного решения бизнес-задач;
- при использовании для управления ИТ-инфраструктурой множества общефункциональных универсальных и специализированных СУ отдельными технологиями и компонентами, СУИ должна не подменять эти СУ, а выступать их интегратором, получая от них обобщенную информацию о состоянии ОМУ и выдавая им макрокоманды, вырабатываемые с учетом приоритетов бизнеса, известных СУИ и неизвестных СУ. Таким образом, СУИ должна предоставлять единую среду интегрированного управления разнообразными ИТ, ресурсами и сетями;
- декомпозиция функций управления ИТ-инфраструктурой и распределение их выполнения между подсистемами СУИ с целью возможности автономной и поэтапной разработки и внедрения функциональных подсистем СУИ и их последующего наращивания и модернизации без существенных изменений в смежных или зависимых подсистемах;
- создание иерархии задач управления, на верхнем уровне которой находятся задачи управления поддержкой бизнес-процессов, на нижнем – управление ОМУ, а на промежуточных – управление функциональными и технологическими подсистемами, отдельными информационными и коммуникационными технологиями и другими компонентами ИУС и ее ИТ-инфраструктуры;
- СУИ, с точки зрения политики управления ИТ-инфраструктурой и принятия решений, должна быть централизованной, с автономным локальным выполнением делегированных функций. Причем наибольший интерес представляет принцип построения СУИ, основанный на рассмотрении в тесной взаимосвязи методов централизованного и децентрализованного управления, активного и

пассивного мониторинга. При этом проводится рациональное разделение задач централизованного и децентрализованного управления и интегрированное использование активного и пассивного мониторинга. Именно такой принцип и реализован в разработанной в данной работе СУИ;

• должна быть реализована возможность быстрого внедрения новых ИТ-услуг и создания новых приложений по управлению ИТ-инфраструктурой без внесения существенных изменений в имеющиеся приложения СУИ;

• в основе архитектуры СУИ должна лежать методология построения открытых систем, обеспечивающая возможности быстрой реконфигурации и дальнейшего развития СУИ благодаря стандартизации компонентов СУИ, использованию международных стандартов и рекомендаций, касающихся протоколов и интерфейсов взаимодействия информационных и телекоммуникационных элементов ИТ-инфраструктуры, а также регламентирующих управление ИТ-инфраструктурой, информационными и телекоммуникационными системами и сетями;

• СУИ должна иметь иерархическую структуру, включающую в себя центральный, региональные и периферийные пункты управления и соответствующие аппаратно-программные компоненты интегрированной СУИ;

• структура СУИ на аппаратном и программном уровнях должна быть модульной и иерархичной, что даст возможность гибко комплектовать, разворачивать и настраивать модули СУИ в зависимости от их уровня и от требований, которым они должны соответствовать, а также позволять легко наращивать функциональность.

Кроме того, для повышения общей эффективности управления функциональными, обеспечивающими подсистемами и ТКС, повышения ответственности администраторов на всех уровнях за принятие и исполнение управляющих решений организация перспективных СУИ должна удовлетворять следующим требованиям:

• обеспечивать рациональное управление ресурсами за счет централизации и систематизации информации о состоянии всех элементов ИТ-инфраструктуры;

• повышать надежность функционирования ИТ-инфраструктуры за счет своевременного получения и анализа информации о состоянии ОМУ, а также оперативного устранения неисправностей;

• содержать механизмы, обеспечивающие единое видение при управлении ИУС в целом у администраторов сети, менеджеров ИТ и руководителей предприятия, позволяющие иметь единое представление проблем управления;

• осуществлять планирование развития ИТ-инфраструктуры благодаря сбору, консолидации и анализу информации о производительности и степени использования системотехнических ресурсов;

• предоставлять единую универсальную среду интегрированного управления разнообразными ИТ, ресурсами и сетями;

• включать в себя центр управления и единую БД для учета различных ИТ, информационных и сетевых ресурсов, персонала и отделов, а также выполняемых ими функций, призванный согласовать действия менеджеров по решению проблем, управления ИТ и ИУС и обеспечить эффективное использование ресурсов;

• осуществлять консолидацию информации о функционировании ИТ-инфраструктуры и приложений;

• обеспечивать представление информации в удобном виде для каждой группы пользователей ИТ-ресурсов (бизнес-подразделений предприятия, ИТ-служб и др.) в соответствии с их обоснованными требованиями;

• давать возможность постепенного наращивания функциональности для упрощения разработки ПО управления ИТ-инфраструктурой, позволяя переделывать не всю систему, а только очередную группу функций, наиболее важную на текущем этапе решения возможных задач;

• осуществлять локализацию родственных функций в определенных компонентах для упрощения внесения изменений в функции других элементов;

• обеспечивать взаимодействие между всеми компонентами распределенной СУИ с возможностью передачи на верхние уровни иерархии обобщенной информации о работе ОМУ в зоне ответственности администраторов СУИ нижнего уровня иерархии и возможностью вмешательства администраторов верхних уровней иерархии в работу ОМУ нижних уровней;

• обеспечивать постоянную автоматизированную связь между администраторами, менеджерами и пользователями, позволяющую быстро выявлять и решать проблемы между людьми при управлении ИТ;

• при централизованном управлении должна быть возможность децентрализации функций управления;

• содержать меньшее количество уровней иерархии, чем организационная структура;

• минимизировать количество администраторов;

• ориентация администраторов на исполнение небольшого количества простых функций;

• использовать универсальные механизмы действий администраторов различных уровней и сервисов с использованием общей информатизации и автоматизации процессов управления;

• использовать единую систему стандартов по техническому, информационному и математическому обеспечению на базе отечественных и зарубежных стандартов и рекомендаций.

Основная специфика интегральной СУИ состоит в том, что, во-первых, ее можно разрабатывать и внедрять как на существующие, так и на вновь разрабатываемые ИУС, а, во-вторых, разработка и внедрение СУИ может проводиться поэтапно. Как правило, на первом этапе создается и внедряется служба помощи (Help Desk), а затем и другие компоненты СУИ, например, система мониторинга телекоммуникационной сети, затем система управления сетью, системы управления серверами и др. Поэтапность создания и внедрения СУИ предполагает использование на каждом этапе соответствующих международных стандартов.

В настоящее время не известны СУИ, удовлетворяющие всем вышеперечисленным требованиям и позволяющие централизованно управлять всеми функциональными и технологическими подсистемами ИУС, а также ИУС, и представляющие инструментарий для ИТ-подразделений, для эффективного автоматизированного управления всеми распределенными ресурсами и приложениями ИУС.

Некоторым из вышеперечисленных требований соответствует СУИ SmartBase.ITS.Control, позволяющая централизованно управлять ИТ-инфраструктурой и представляющая ИТ-персоналу инструментарий эффективного автоматизированного управления ресурсами ИТ-инфраструктуры.

11.1.4. Этапы создания и внедрения системы управления ИТ-инфраструктурой

Создание СУИ как централизованной СУ, исключающей остановку предоставления сервисов пользователям со стороны ИТ-инфраструктуры, учитывая непрерывность работы корпоративных ИТ-инфраструктур, внедрение СУИ и ее последующий запуск предполагает и требует:

- применения методики проектирования, ориентированной на решение задач многокритериальной оптимизации, поскольку СУИ являются системами многоцелевого назначения, которые характеризуются рядом противоречивых показателей эффективности и работают в различных режимах;
- поэтапного внедрения функционально законченных подсистем СУИ;
- обеспечения интеграции как имеющихся, так и перспективных автономных подсистем управления в единую СУ.

Процесс проектирования и внедрения СУИ должен отличаться гибкостью, непрерывностью и поэтапностью. Причем гибкость должна проявляться как на этапах планирования и проектирования ИТ-инфраструктуры, так и при ее практической реализации. Кроме того, СУИ должна иметь гибкую архитектуру, позволяющую легко изменять структурные связи и наращивать функциональность.

На *первом этапе* проводится учет всех ресурсов ИТ-инфраструктуры, включая как аппаратные средства, так и сервисы, осуществляется создание и внедрение единой системы мониторинга ресурсов и услуг. При этом работа ведется в двух направлениях:

1) создание или совершенствование существующей системы мониторинга элементов ИТ-инфраструктуры I и II уровней, а также технологических сервисов, с реализацией таких функции как сбор, обработка и отображение информации о состоянии аппаратных и программных элементов функциональных и технологических подсистем корпоративной ИУС;

2) разработка системы мониторинга приложений и бизнес-сервисов.

На первом этапе разработки и внедрения СУИ в существующие ИУС проводится учет и именование всех ресурсов ИУС, включая как аппаратные средства в виде серверов, хранилищ данных и др., так и службы, сервисы и др., выполняемые функциональными и обеспечивающими подсистемами, а также ТКС.

При этом осуществляется группирование ресурсов по целому ряду признаков, в частности, по выполняемым функциям, по взаимосвязи с другими ресурсами, по физическому расположению и другим признакам. Создается топологическая схема ресурсов, подключенных к телекоммуникационной сети, которая описывается в явном виде – в форме графа или в неявном виде – в форме таблиц или схем зависимостей. На этом этапе создается или совершенствуется единая диспетчерская служба (Service Desk).

Следующим шагом внедрения СУИ может стать процесс управления конфигурациями – серверного и сетевого оборудования, ПО и др. На этом этапе прежде всего нужно создать базу данных конфигурационных элементов и связей между ними (например CMDB) с системой учета и инвентаризации. При этом могут быть использованы различные средства автоматизации учетных операций, например, технологии штрих-кодирования.

Автоматизацию управления ИТ-инфраструктурой целесообразно начинать с автоматизации отдельных функций управления. На первой фазе формируется

стек автоматизированных функций МАУОП на всех четырех уровнях ИТ-инфраструктуры. Следующая фаза – интеграция механизмов управления, которая требует системного подхода. Последней стадией становится организация интегрированного управления ИТ-инфраструктурой в комплексе с управлением выполнением бизнес-процессов.

Проектирование и внедрение СУИ должно происходить так, чтобы ее можно было внедрять по частям, и уже на первых этапах эксплуатации только части СУИ мог быть сразу получен положительный эффект. Для этого необходимо снизить время простоя ИТ-инфраструктуры путем автоматизации управления устранением неисправностей, созданием системы анализа данных об инцидентах. При этом в составе ИТ-службы предприятия необходимо сделать организационные изменения – создать диспетчерскую службу, выполняющую функцию технической поддержки, которая осуществляет поддержку пользователей в процессе управления устранением неисправностей. С помощью средств мониторинга ИТ-инфраструктурой осуществляется автоматический сбор информации о неисправностях в аппаратных и программных элементах функциональных подсистем и автоматическое оповещение об инцидентах. Если к неисправностям отнести и отклонения технологических параметров в ИУС, то это позволит прогнозировать техническое состояние оборудования ТП и обеспечить своевременное и качественное проведение профилактических и ремонтных работ.

Для эффективной работы подсистемы управления устранением неисправностей необходимо разработать систему метрик и численных показателей для классификации отказов, часть которых необходимо прописать в SLA. Система метрик позволяет классифицировать отказы, определять их категорию, причины, время восстановления, степень влияния на бизнес-процессы, реакции на инцидент и специфицировать другие показатели. Метрики проектируются сверху вниз.

Второй этап заключается в создании и внедрении интегрированной СУ. На этом этапе разрабатываются бизнес-показатели и критерии оценки функционирования ИТ-ресурсов, обеспечивается мониторинг этих показателей.

При решении прикладных задач управления ИТ-инфраструктурой, разработке, инсталляции и настройке приложений управления широко используются математическое, имитационное, натурное и другие методы моделирования процессов управления элементами ИТ-инфраструктуры, ФПС, ТПС и т.п., необходимые для определения пороговых значений состояний и параметров работы ОМУ, превышение которых требует от СУИ формирования управляющих воздействий.

Третьим этапом является автоматизация процесса управления качеством услуг. При этом необходимо сформировать каталог услуг, определить требования к ним, задать параметры контроля, разработать SLA, построить адекватную организационную структуру.

На *этапе эксплуатации* проводится модернизация совершенствование СУИ. При этом осуществляется:

- анализ изменения характеристик и свойств бизнес-процессов;
- анализ динамики параметров ИТ-инфраструктуры и ИУС;
- проверка возможности использования перспективных ИТ-технологий, парадигмы облачных вычислений и др.;
- поддержка принятия решений по модернизации ИТ-инфраструктуры с учетом информации о прогрессивных ИТ, данных о производителях и поставщиках

технических и программных средств, сравнительных технических и экономических характеристиках этих средств и др.

Поэтапное внедрение СУИ позволит более эффективно использовать информационные и коммуникационные ресурсы, без перерывов в работе ИУС, а в последствии осуществлять управление аппаратно-программными средствами в единой интегрированной среде.

11.2. СТРУКТУРА И ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ ИТ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ

Как уже отмечалось, создание СУИ целесообразно осуществлять на основе ИТ УИТИ. Несмотря на их общность, связанную с реализацией функционала управления объектами широкого класса, СУИ и ИТ УИТИ являются принципиально различными сущностями. ИТ УИТИ – это только основа, на которой создаются СУИ. Они имеют много общего в функционале, но при этом важные для СУИ организационная структура, комплекс технических средств, база данных и другие составляющие для ИТ УИТИ существуют только в виде требований, которым она должна соответствовать.

Однако если не определить точно эти требования, то созданную ИТ УИТИ нельзя будет сделать компонентом, «кирпичиком» СУИ. При этом наиболее важные требования ИТ УИТИ должна учитывать на уровне архитектуры и принципов функционирования ИТ УИТИ как составляющей СУИ.

Поскольку архитектура инструментальных средств ИТ УИТИ построена на принципах трехуровневой клиент-серверной технологии, сосредоточимся на ее функциональности.

Функциональная структура ИТ УИТИ. Функциональная структура ИТ УИТИ, реализующая вся пять процессов МАУОП, приведена на рис. 11.1.

В ИТ УИТИ, функциональная структура которой приведена на рис. 11.1, каждый из процессов МАУОП реализуется программным модулем, выполняющим соответствующие функции. Модульность структуры позволяет легко наращивать функционал отдельных модулей без внесения существенных изменений в другие модули, а также создавать на основе ИТ УИТИ облегченные версии СУИ, выполняющие только часть процессов МАУОП, например, мониторинга и управления или только мониторинга, с сокращенным функционалом. Цель такого подхода – превратить создание СУИ из трудно контролируемой совокупности хаотических действий по закупке компонентов ИТ-управления в управляемый бизнес-процесс создания интегрированной СУИ.

Исходя из пяти процессов МАУОП, жизненный цикл управления ИТ-инфраструктурой целесообразно рассматривать как последовательное прохождение пяти соответствующих фаз – мониторинга, анализа, управления, оптимизации и планирования. Управление жизненным циклом позволяет планировать развитие ИУС, выявляя и анализируя тенденции, сопоставляя их с планом развития ИТ-инфраструктурой для выявления отклонений. При этом осуществляется переход к непрерывному процессу управления в СУИ.

В соответствии с функциональной структурой (см. рис. 11.1) подсистемы СУИ выполняют следующие основные функции.

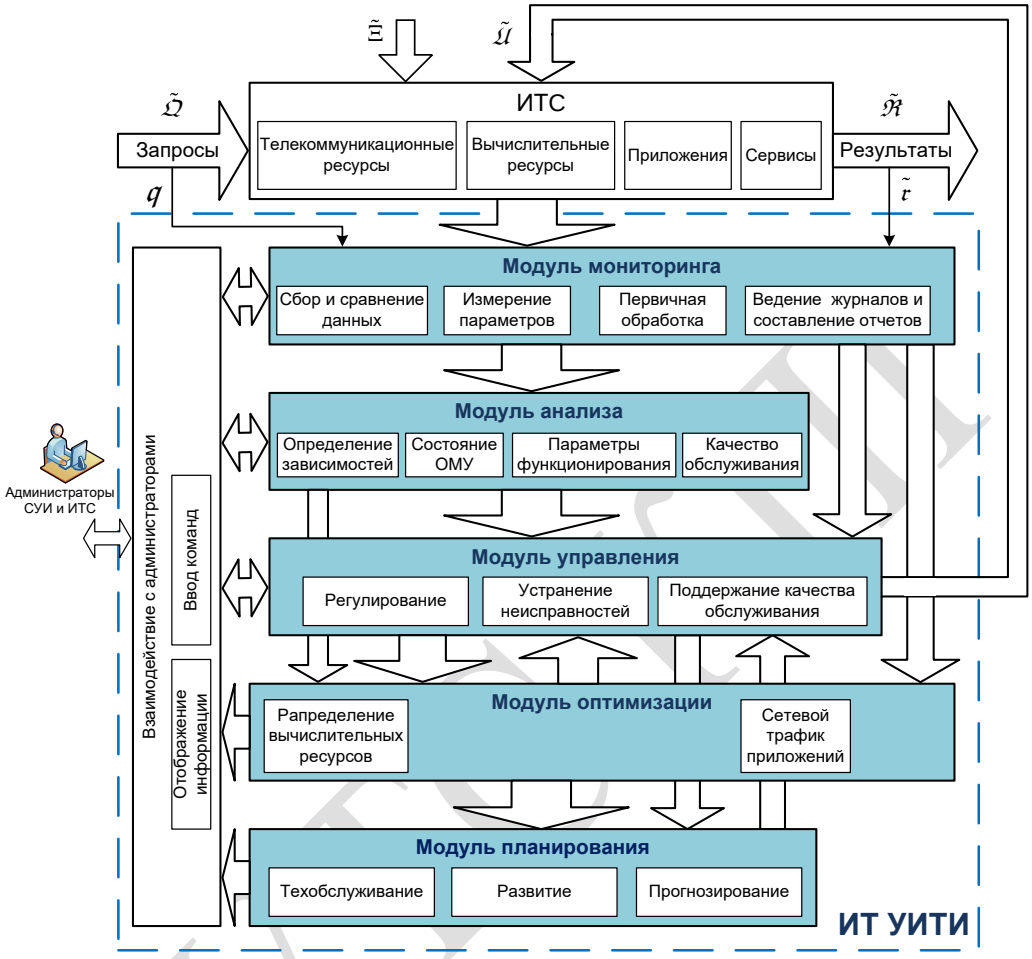


Рис. 11.1. Функциональная структура ИТ УИТИ

Модуль мониторинга определяет состояние элементов ИТ-инфраструктуры путем сбора, измерения, первичной обработки и сравнения данных с пороговыми значениями, формирует отчетность путем ведения журналов о функционировании ИТ-инфраструктуры и работе ИТ-подразделения, отображает результаты мониторинга в виде карт, диаграмм, информационных панелей и других форматов представления данных, с предоставлением механизмов фильтрации с целью обеспечивая для пользователей возможности многовариантного выбора показателей. Кроме того, осуществляется идентификации ресурсов и проблем в ИТ-инфраструктуре.

Модуль анализа проводит анализ результатов мониторинга для определения состояния ОМУ. Модуль задействуется при управлении устранением неисправностей для осуществления оценки и анализа данных, необходимых для выявления проблемных областей при определении первопричин возникновения неисправностей, выполнения импакт-анализа.

Модуль управления выбирает управляющие воздействия из множества $\tilde{\mathcal{U}}$ в соответствии с (2.3) на основе политик управления с учетом существующих ограничений. В модуле реализованы функции координатора.

Модуль оптимизации решает задачи оптимального распределения и перераспределения ресурсов ИТ-инфраструктуры, а также задачи системной оптимизации.

Модуль планирования решает задачи прогнозирования поведения ИТ-инфраструктуры, выявления тенденций в использовании ресурсов, отслеживания проведения регламентных работ и планирования развития: определения динамики роста, оптимизации закупок при масштабировании или модернизации ИТ-инфраструктуры, выработки политики развития и др.

Каждый из модулей реализует свои функции для всех уровней иерархической модели ИТ-инфраструктуры и содержит блоки, реализуемые в виде совокупности приложений, среди которых есть приложения, решающие, например, такие задачи как управление распределением ресурсов, управление техническим обслуживанием и др.

Распределение функций между компонентами рассредоточенной СУИ определяется топологией и масштабами ИТ-инфраструктуры. Так, применение централизованной подсистемы мониторинга может оказаться неприемлемой из-за большого объема генерируемого трафика, в то время как использование региональных серверов СУИ и агентской технологии позволяет сократить трафик мониторинга. Кроме того, централизация управления сетью, вычислительными ресурсами, сервисами и приложениями позволяет существенно уменьшить эксплуатационные расходы.

Все особенности построения СУИ учитываются на этапе проектирования СУИ с учетом топологии, масштаба и назначения ИТ-инфраструктуры.

Функциональная схема модуля мониторинга. Сбор значений параметров $p_l^{(i,j)}$, $l=\overline{1,n}$, $i=\overline{1,I_j}$, $j=\overline{1,J}$, осуществляет модуль мониторинга, функциональная схема которого приведена на рис. 11.2.

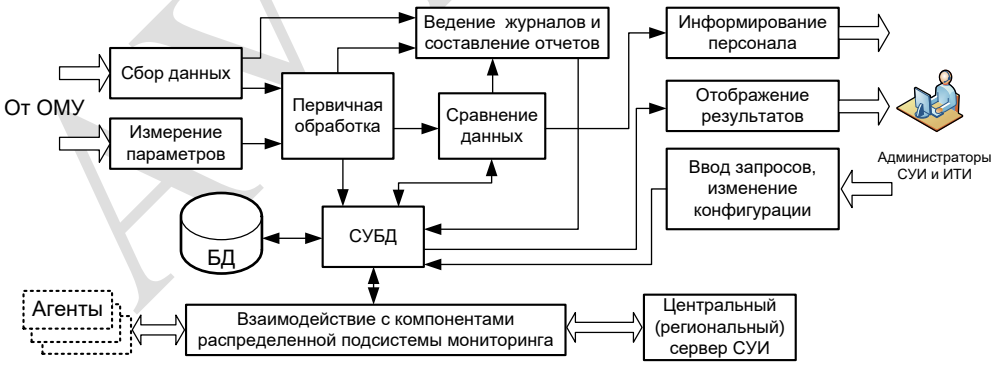


Рис. 11.2. Функциональная схема модуля мониторинга

Функциональная схема модуля анализа. Анализ состояний всех ОМУ $O_{i,j}$, $l=\overline{1,n}$, $i=\overline{1,I_j}$, осуществляет модуль анализа, функциональная схема которого приведена на рис. 11.3.

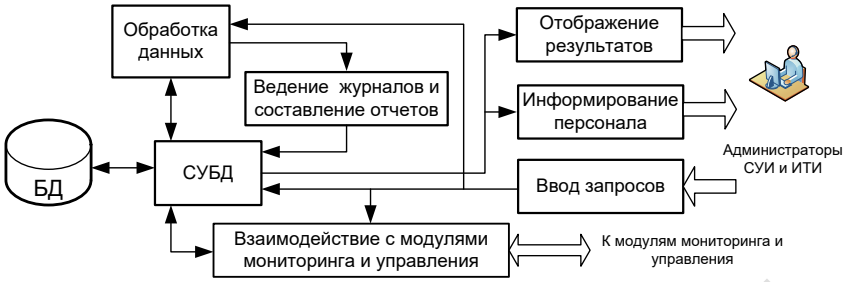


Рис. 11.3. Функциональная схема модуля анализа

Функциональная схема модуля управления. Разработка модуля управления, функциональная схема которого приведена на рис. 11.4, проведена с помощью эффективного инструментария на основе платформы SmartBase [162]. При разработке модуля управления использованы современные подходы к разработке программных систем [6, 15, 23, 62, 75, 97, 98, 127, 166], а также методологии проектирования ИУС [4, 16, 53, 106, 128].

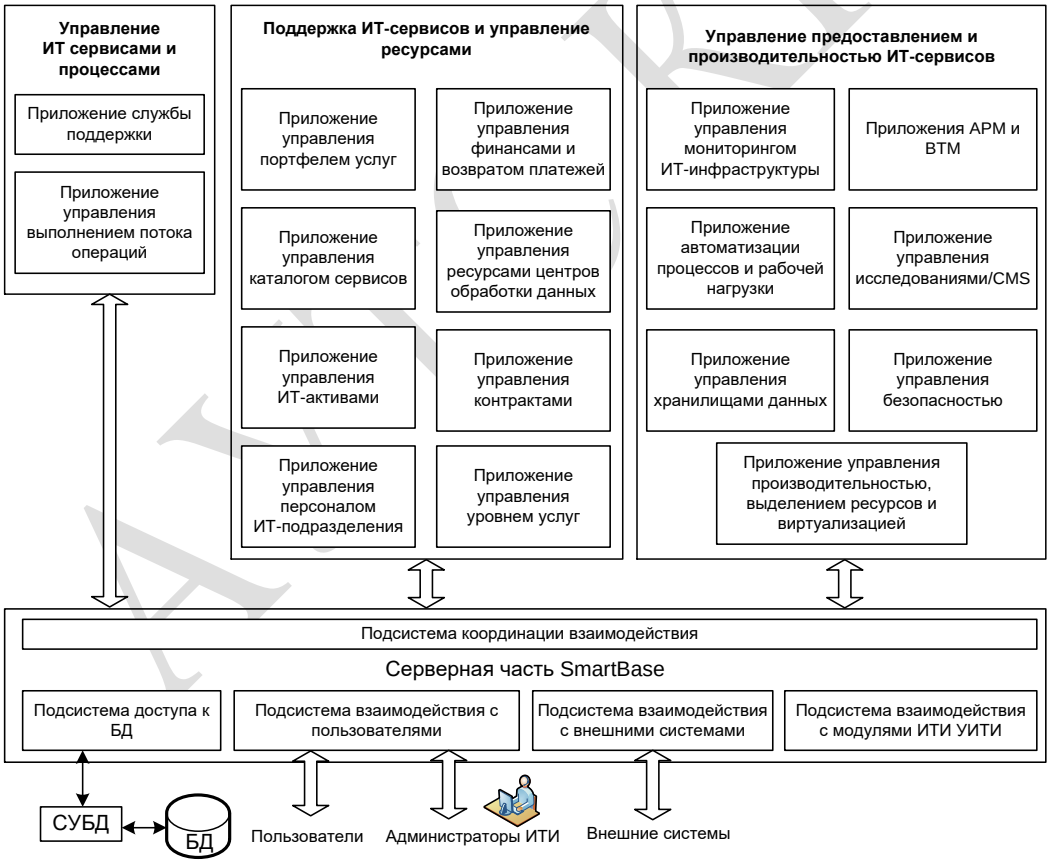


Рис. 11.4. Схема модуля управления на базе платформы SmartBase

В модуле управления реализуются методы и алгоритмы, предложенные в данной работе, в виде программных компонентов приложений управления ИТ-инфраструктурой. Комплекс приложений управления ИТ-инфраструктурой соответствует дифференциации ПО ITMS2.

Функциональная схема модуля оптимизации. Модуль оптимизации призван решать оптимизационные задачи при выполнении процессов МАУОП, прежде всего, задачи оптимального распределения вычислительных ресурсов ИТ-инфраструктуры. Функциональная схема модуля оптимизации приведена на рис. 11.5.

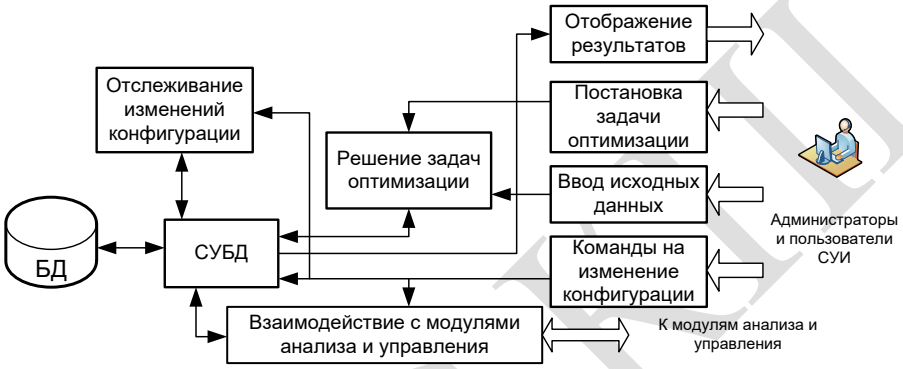


Рис. 11.5. Функциональная схема модуля оптимизации

Модуль оптимизации использует два режима работы: автоматизированный и автоматический.

Инициализация запуска решения задач проводится по сигналам модуля управления или инициируется администратором СУИ. В автоматизированном режиме администратор выбирает задачу оптимизации и задает исходные данные. Наблюдая за ходом решения задачи, он корректирует процесс получения близкого к оптимальному результата. Реализация решения проводится соответствующим изменением конфигурации ИТ-инфраструктуры.

Процесс решения задач оптимизации может запускаться и реализовываться автоматически при поступлении информации, например, об изменении приоритетов бизнес-процессов, возникновении критичных отказов и других важных событиях. В этом случае модуль анализа сигнализирует о необходимости запуска задачи оптимизации, а в модуль управления поступает новый план распределения ресурсов.

Функциональная схема модуля планирования. Модуль планирования, функциональная схема которого приведена на рис. 11.6, предназначен для автоматизации решения задач оптимального развития ИТ-инфраструктуры.

Модуль планирования отслеживает проведение регламентных работ, прогнозирует поведение ИТ-инфраструктуры, выявляет тенденции использования ресурсов, определяет динамику роста ИТ-инфраструктуры, вырабатывает политики и планирует развитие, оптимизирует закупки при масштабировании или модернизации ИТ-инфраструктуры и другие функции.

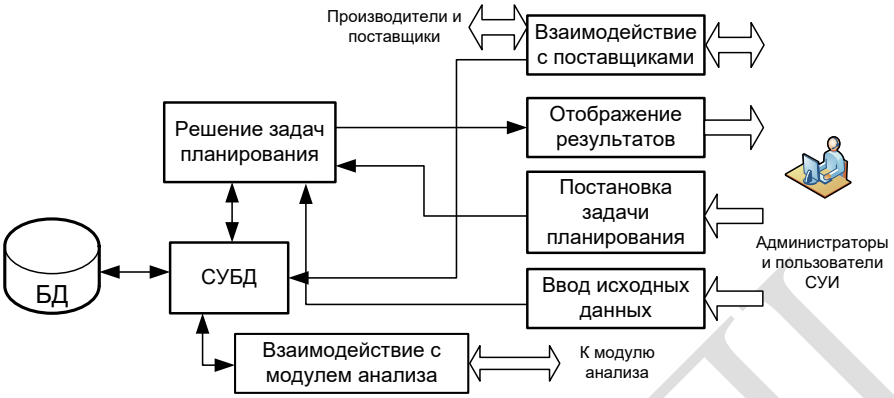


Рис. 11.6. Функциональная схема модуля планирования

Для решения задач планирования обычно используется ПО управления закупками, автоматизации проведения работ по обслуживанию, пакеты прикладного анализа данных и другие средства.

11.3. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ SMARTBASE.ITS.CONTROL

СУИ SmartBase.ITS.Control создана на основе разработанной ИТ УИТИ, инструментальные средства которой в виде программных модулей охватывают базовую функциональность СУИ. Функции управления ИТ-инфраструктурой обеспечиваются за счет организационных решений и действий персонала, а также инструментальных средств ИТ УИТИ и технических средств СУИ. Организационные решения включают формирование организационной структуры управления ИТ-инфраструктурой на основе типовой, определение перечня служебных обязательств и порядок взаимодействий администраторов, а также снабжения их необходимой информацией.

Технические средства СУИ состоят из серверов СУИ и АРМ администраторов. ПО СУИ состоит из серверной, клиентской и агентской частей.

11.3.1. Разработка структуры системы управления ИТ-инфраструктурой

В крупных, территориально распределенных на существенных расстояниях ИУС, содержащих большое количество элементов, СУИ целесообразно строить по распределенному иерархическому принципу (рис. 11.7), когда серверные компоненты объединяются в связанные структуры, образуя иерархию. В этом случае на верхнем уровне располагается главный или центральный сервер СУИ, определяющий состояние крупных составляющих ИУС – функциональных и технологических подсистем, телекоммуникационной сети и других составляющих, а на нижних уровнях – региональные или периферийные серверы СУИ.

Периферийные серверы непосредственно или с помощью агентов осуществляют мониторинг, анализ функционирования и управление элементами ИТ-инфраструктуры в своей географической или административной зоне ответственности.

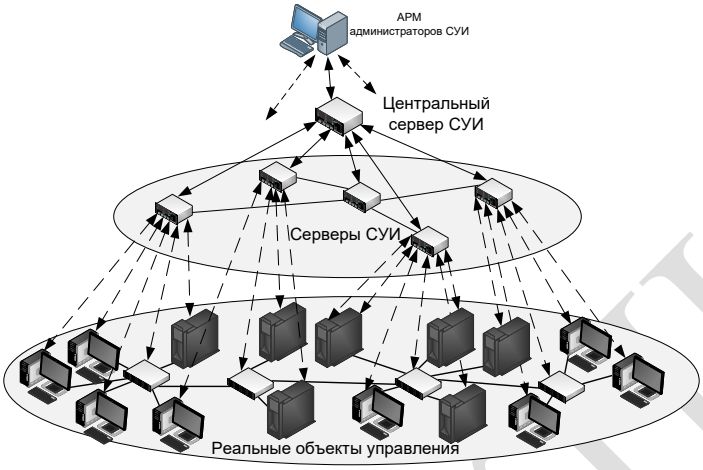


Рис. 11.7. Иерархия серверов СУИ

С нижних уровней на верхние поступают обобщенные данные мониторинга и анализа о функционировании элементов ИТ-инфраструктуры, расположенных в зоне ответственности периферийных серверов СУИ. С верхних уровней на нижние распространяются политики управления и макрокоманды. Все серверы СУИ управляются администраторами, которые находятся в иерархической соподчиненности в соответствии с организационной структурой департамента, обеспечивающего функционирование ИТ-инфраструктуры.

Для определения состояния ИТ-инфраструктуры и ИУС, оценки качества функционирования элементов ИТ-инфраструктуры и управления всеми ОМУ сервера СУИ от РОУ получают значения параметров, обрабатывают эти данные, выносят решение о состоянии РОУ, определяют необходимое управление для поддержания функционирования РОУ в соответствии с заданными требованиями и реализуют управляющие воздействия.

Обобщенная информация о функционировании элементов ИТ-инфраструктуры поступает администратору, который наблюдает за ходом процесса управления ИТ-инфраструктурой и принимает решения по управлению РОУ.

Взаимодействие и совместная работа компонентов СУИ осуществляется через телекоммуникационную сеть. Через сеть осуществляется взаимодействие между серверами СУИ и элементами функциональных и технологических подсистем ИУС, АРМ администраторов СУИ с администраторами функциональных, технологических подсистем и телекоммуникационной сети.

Для эффективного использования предложенной в данной работе ИТ УИТИ при разработке, эксплуатации и развитии СУИ SmartBase.ITS.Control, специальное ПО разработанной СУИ, состоящее из трех отдельных программных компонентов – серверного, клиентского и агентского программных модулей, как это показано на рис. 11.8, должно быть реализовано в виде компонентов ИТ УИТИ.

Серверный программный модуль (сервер) обеспечивает взаимодействие между клиентами, предоставляет доступ к данным, обеспечивает их целостность и защиту, уведомляет администратора о критических или интересующих его со-

бытиях в ИУС, осуществляет планирование действий по мониторингу, осуществляет анализ и управление элементами ИТ-инфраструктуры. Непосредственно мониторинг и реализацию управляющих воздействий серверный модуль осуществляет совместно с агентскими модулями.

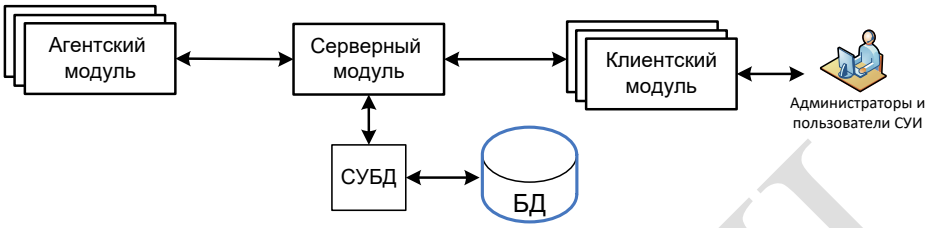


Рис. 11.8. Структура СУИ SmartBase.ITS.Control

Для сбора данных мониторинга и реализации управляющих воздействий сервер взаимодействует с элементами ИУС через агентские модули. Важной функцией сервера является синхронизация и обработка данных, а также уведомление подключенных агентов и клиентов о произошедших изменениях.

В СУИ должны быть реализованы режимы автоматического, автоматизированного и ручного управления ОМУ. В автоматизированном и ручном управлении участвует администратор СУИ, АРМ которого реализуется на основе клиентского модуля. Для эффективного управления администратора в СУИ осуществляется поддержка принятия решения администраторами.

Клиентский программный модуль (клиент) выполняется на АРМ администраторов и реализует интерфейс администратора для настройки СУИ, отслеживания работоспособности и производительности элементов ИУС путем просмотра в удобном виде данных мониторинга, результатов анализа, разных журналов действий и других данных, контроля состояния функциональных и технологических подсистем и ИУС в целом; контроля проведения регламентных и ремонтных работ, локализации и устранения неисправностей; удаленного управления конфигурацией и работой ИУС; ввода команд управления при реконфигурации или проведении восстановительных мероприятий. Непосредственно мониторинг элементов ИУС и реализация управляющих воздействий осуществляется агентскими и серверным модулями, в то время как основная функция клиентского модуля – предоставление графического интерфейса для взаимодействия администратора с системой. Присутствие модуля анализа позволяет определить негативные тенденции, например, выявить деградацию отдельных важных параметров во времени, и своевременно сообщать об этом администраторам, что позволяет устранять некоторые проблемы еще до их возникновения.

Агентский программный модуль (агент) контролирует ОМУ, анализирует их состояние и поведение. Кроме того, агенты используются при моделировании перспективного состояния как отдельных элементов, так и всей ИУС. Это позволяет предвосхищать и предотвращать функциональные отказы еще при первых признаках их проявления, что, в свою очередь, повышает надежность функционирования ИУС.

Агент устанавливается на РОУ и запускается в виде сервиса, специализирующегося на выполнении индивидуального набора функций мониторинга и управления, конфигурируемых в зависимости от задач, решаемых агентом и объектами, которые он обслуживает.

Собираемые агентом данные могут обрабатываться как на этом же, так и на других агентах, и вместе с результатами их обработки хранятся в локальной БД или отправляются на сервер. По запросу или по расписанию агент передает на сервер СУИ обобщенные данные мониторинга, что позволяет уменьшить количество запросов сервера СУИ к компонентам ИУС, а также загруженность телекоммуникационной сети.

11.3.2. Программный сервер СУИ

Структура сервера СУИ SmartBase.ITS.Control приведена на рис. 11.9.

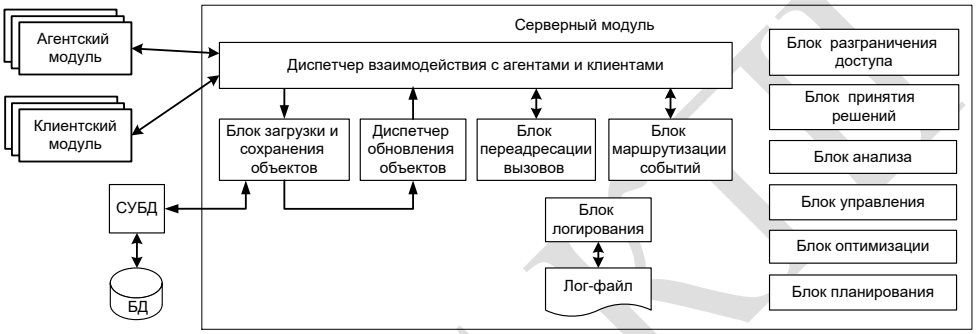


Рис. 11.9. Структура сервера СУИ SmartBase.ITS.Control

Сервер СУИ осуществляет сбор данных мониторинга и управление ОМУ посредством агентских модулей, реализует взаимодействие с клиентскими модулями, предоставляет доступ к БД, обеспечивает защиту и целостность данных, осуществляет поддержку принятия решений по управлению ИТ-инфраструктурой, генерирует команды управления ОМУ, уведомляет администратора о критических событиях и т.п.

Сервер состоит из диспетчера взаимодействия с агентами и клиентами, а также блоков: загрузки и сохранения объектов; диспетчера обновления объектов; переадресации вызовов; маршрутизации событий; разграничения доступа; принятия решений; анализа; управления; оптимизации; планирования; логирования.

Диспетчер взаимодействия с агентами и клиентами осуществляет маршрутизацию информационных потоков между компонентами СУИ, проводит корректное подключение и отключение агентов и клиентов с рассылкой соответствующих уведомлений. Для получения доступа к серверу клиенты проходят процедуру регистрации, при которой проводится аутентификация и проверка прав. Любые запросы незарегистрированных клиентов, кроме запроса на регистрацию, автоматически отклоняются. Для поддержания связи подключенный клиент с заданной частотой путем передачи соответствующего запроса проверяет доступность сервера. В случае недоступности сервера клиент осуществляет попытки повторного подключения. Сервер проводит аналогичную проверку только в случае отсутствия в течение заданного времени запросов от клиента. При недоступности клиента сеанс прекращается. Для сокращения загруженности телекоммуникационной сети запрос доступности не проводится при наличии активного информационного обмена между агентами и клиентами с сервером.

Блок загрузки и сохранения объектов осуществляет обработку запросов, кэширование ЛОУ и запросов, а также реализует механизм синхронизации. Этот

блок вместе с диспетчером обновления объектов выполняет загрузку и сохранение логических отображений ОМУ – ЛОУ, а также вместе с диспетчером обновления объектов обеспечивают целостность данных и своевременное уведомление клиентов о новых версиях объектов.

При получении некоторого запроса на загрузку ЛОУ проверяется его наличие в кэше и, в случае отсутствия, запрос перенаправляется в БД. При этом предоставляются находящиеся в кэше готовые ЛОУ, а отсутствующие конструируются из объектов, находящихся в БД, с последующим добавлением в кэш. Благодаря этому значительно повышается скорость выполнения запросов и обеспечивается гарантированное наличие в системе не более одной копии каждого ЛОУ. Механизм синхронизации используется для получения от клиента обновленных версий ЛОУ, а для применения изменений, внесенных данным клиентом, отслеживаются версии атрибутов ЛОУ и его связей. Данный механизм позволяет редактировать один и тот же объект на разных клиентах. После выполнения синхронизации все затронутые изменениями ЛОУ сохраняются в БД и передаются диспетчеру обновления объектов для оповещения подключенных клиентов.

Диспетчер обновления объектов обеспечивает целостность и актуальность данных. Для этого реализуется механизм, отвечающий за своевременное оповещение клиентов о произошедших изменениях структуры дерева ЛОУ и его составляющих. Для каждого подключенного клиента диспетчер строит выборки объектов из списка обновленных. Построение выборок сводится к решению NP-полной задачи, так как для решения задачи определения необходимости оповещения клиента об обновлении некоторого ЛОУ необходимо решить аналогичную задачу для каждого из связанных с ним объектов. Затраты на определение необходимости оповещения клиента минимальны для объектов, связанных непосредственно с клиентом, и велики для объектов, привязанных к клиентам через цепь других объектов. Поэтому нецелесообразно проводить выборку объектов некоторых типов, отправляя все измененные объекты клиенту. Для снижения сетевого трафика клиентам передаются не измененные объекты, а список их идентификаторов, а клиент запрашивает обновление только тех объектов, которые им используются. Эти механизмы позволяют нескольким клиентам одновременно работать с удаленными объектами, своевременно получая уведомления об изменениях их состояния и структуры.

Блок переадресации вызовов и маршрутизации событий, а также диспетчер обновления объектов позволяют клиентам работать с удаленными объектами, своевременно получая уведомления об изменениях их состояния и структуры.

При построении систем клиент-серверного взаимодействия, при котором клиенты взаимодействуют друг с другом только через сервер, возникает проблема обращения к ЛОУ, месторасположение которого неизвестно. ЛОУ может находиться только на одном агенте, а все его воплощения (копии) на других машинах являются заместителями, поэтому необходимо неявно перенаправлять вызов некоторой функции объекта-копии реальному объекту. При этом обращение к удаленным объектам осуществляется как к локальным, а все механизмы, осуществляющие переадресацию, скрываются от администратора СУИ. Это обеспечивает взаимную анонимность вызывающего и конечного клиентов, поскольку принятие решения о правомерности такого вызова и нахождение конечного клиента осуществляет сервер. Если при обращении к объекту выясняется, что он является копией, то задействуется встроенный механизм переадресации, который

обращается к серверу, где происходит определение конечной машины, на которой находится объект, и ее доступность, после чего запрос перенаправляется конечному клиенту. Конечный клиент возвращает результаты запроса на сервер, который, в свою очередь, перенаправляет результаты запросившему клиенту. В случае недоступности сервера или конечного клиента, ошибки или отказа в доступе клиенту возвращается описание причины неудачного выполнения запроса.

Для обеспечения работы механизма реактивного обновления состояний ЛОУ вне рамок одного агента должна осуществляться своевременная передача уведомлений о событиях, произошедших на данном агенте, другим агентам, имеющим зависящие ЛОУ. Для подписки на события и их передачи агенты не обязаны знать о местонахождении конкретных ЛОУ, обращаясь с находящимися у них копиями, как с реальными ЛОУ. Реализация процесса подписки показана на рис. 11.10.

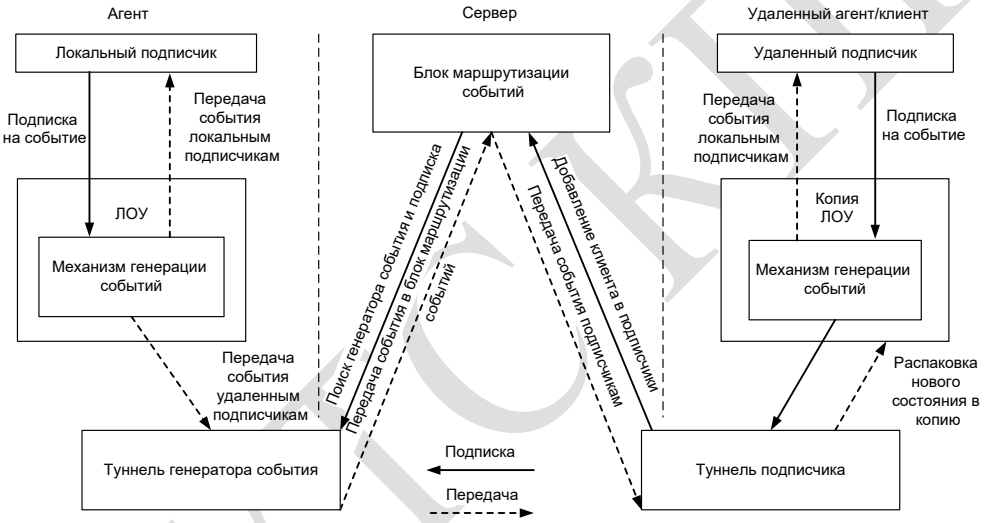


Рис. 11.10. Процесс подписки на удаленное событие и его вызов

Если при подписке на событие определяется, что данный объект является копией (заместителем), проводится подписка на получение данного события на локальном туннеле. Туннель представляет собой общий для данного клиента механизм подписки, отписки и маршрутизации событий. Туннель переадресует подписку на серверный блок маршрутизации событий, где существует реестр событий и клиентов – подписчиков и источников. Сервер определяет клиента, которому принадлежит генерирующий событие объект, и добавляет кортеж (объект–клиент-источник–клиент-подписчик) в реестр событий. Далее диспетчер взаимодействия с агентами и клиентами определяет, подключен ли клиент-источник к серверу, и отправляет ему запрос на отсылку подписываемого события.

Подписка автоматически восстанавливается после разрывов сеансов связи. При отключении клиента-подписчика блок маршрутизации событий автоматически аннулирует все его подписки, а после восстановления подключения клиент заново осуществляет подписку. При отключении клиента-источника подписки на его события остаются в реестре, и при повторном подключении клиента к сети

блок маршрутизации событий сообщает об ожидающих его подписках. При возникновении события обновления состояния агент делает рассылку локальным подписчикам и, в случае наличия удаленных подписок, происходит отправка этого события и сопутствующих атрибутов на серверный блок маршрутизации событий, который выполняет его ретрансляцию всем клиентам-подписчикам. Получив событие, клиентский туннель распаковывает его атрибуты в объект-копию, после чего генерирует аналогичное событие на локальной машине от имени объекта-копии. Обычно в качестве атрибутов события отправляется текущее значение или обновленное состояние объекта. Таким образом, использованные механизмы и подходы позволяют реализовать поддержку реактивного обновления состояний ОМУ и своевременное обновление состояний копий ОМУ на всех подписанных клиентах.

Блок разграничения доступа проверяет права агентов и клиентов при попытках получения доступа, редактирования или удаления объектов и связей, редактирования прав, вызова удаленных функций некоторого объекта или подписки на события об изменении его состояния.

Серверный блок анализа в отличие от агентского, который анализирует состояния отдельных ОМУ, анализирует состояние ИТ-инфраструктуры и ИУС в целом. Блок анализирует полученные от агентов данные мониторинга и анализа параметров с использованием различных статистических методов. В частности, выявляются тренды поведения, проводится работа по архивации данных мониторинга и др. Проактивный мониторинг осуществляется путем анализа и определения тенденции динамики изменения состояния определенных параметров функционирования системы и прогнозирования возможных будущих состояний.

На основании результатов анализа блок принятия решений вырабатывает варианты воздействия на ИТ-инфраструктуру в соответствии с используемыми критериями, политиками и алгоритмами. Изначально администратор вручную задает алгоритмы поведения системы при возникновении определенных видов неисправностей, а в процессе обучения, запоминая и анализируя действия администратора, блок принятия решений отбирает наиболее часто использующиеся решения и предлагает их администратору, который в свою очередь может на основании этих рекомендаций совершенствовать алгоритмы поведения системы. Принятые решения могут быть следующих типов: уведомление администратора о существующей или возможной в будущем неисправности; рекомендации администратору по устранению неисправности; автоматизированное принятие решения или предложение выбора из готовых вариантов решения; автоматическое устранение неисправности или реконфигурация ИТ-инфраструктуры.

Блок управления воздействует на РОУ посредством передачи управляющих запросов соответствующим им ЛОУ. При этом блок переадресации вызовов перенаправляет запрос агенту, который непосредственно воздействует на данный ЛОУ. Блок управления работает в ручном и автоматическом режимах, что позволяет управлять ИТ-инфраструктурой посредством клиентского интерфейса или путем автоматического исполнения действий по реконфигурации ИТ-инфраструктуры, перераспределению ресурсов [114], управлению информационными потоками [119] и другими средствами.

Блоки оптимизации и планирования выполняют соответствующие задачи с заданным периодом или в заданное время, а блок логирования сохраняет историю системных событий.

Обслуживаемые сервером модули. К обслуживаемым сервером модулям относятся агентский и клиентский программные модули. Эти модули устанавливаются на отдельные серверы и АРМ операторов СУИ, взаимодействуют с сервером СУИ и друг с другом посредством сервера.

Обслуживаемые сервером модули обязательно должны содержать следующие компоненты, обеспечивающие общение с сервером и входящие в блоки взаимодействия агента и клиента:

- компонент конфигурирования и мониторинга подключения;
- компонент загрузки и сохранения объектов;
- компонент переадресации вызовов;
- компонент маршрутизации событий;

Компонент конфигурирования и мониторинга подключения выполняет следующие функции:

- настройка стартовой конфигурации обслуживаемого модуля;
- инициализация механизма логирования;
- проверка прав подключения;
- непосредственное подключения к серверному модулю;
- отслеживание состояния подключения (выявление потерь соединения с сервером и других событий);
- повторное подключение (в случае разрыва соединения).

Компонент загрузки и сохранения объектов взаимодействует с соответствующим серверным блоком. Компонент организовывает запросы на получение, сохранение или обновление объектов, с которыми работает обслуживаемый модуль. Эти запросы обрабатываются серверным блоком загрузки и сохранения объектов. Результатом взаимодействия блока обслуживаемого модуля и серверного блока является автоматическое обновление данных всех модулей СУИ, выполняемых в них при инициализации любым из модулей СУИ.

Компонент переадресации вызовов взаимодействует с соответствующим серверным модулем, отправляя ему удаленные вызовы и запросы удаленным ОМУ. Он также принимает удаленные вызовы и перенаправляет запросы от удаленных ОМУ соответствующим локальным.

Компонент маршрутизации событий осуществляет подписку на удаленные события и выдачу произошедших локальных событий удаленным подписчикам.

11.3.3. Программный агент СУИ

В мониторинге ОМУ основную роль играют программные агенты СУИ. Такие агенты представляют собой универсальные, гибкие инструменты с модифицируемым набором функций, позволяющие контролировать ОМУ, анализировать их поведение и состояние. Каждый агент специализируется на выполнении определенного набора функций МАУОП, задаваемого администратором в зависимости от возлагаемых на агента задач.

Мониторинг элементов ИТ-инфраструктуры осуществляют программные агенты СУИ. Эти агенты СУИ также представляют собой универсальные, гибкие инструменты с модифицируемым набором функций. Но они используются для контроля элементов ИТ-инфраструктуры, анализа их поведения и состояния.

Модель программного агента базируется на пяти основных элементах классов: сущность, действия, отношения, сообщения и проверочные функции. Эти элементы обеспечивают функционирование агента, а также его взаимодействие

с другими агентами и сервером или клиентами. Каждый агент содержит *сущность* – логический объект, активный элемент модели агента, выполняющий некоторое подмножество его функций. Для него соответствующий набором свойств определяется и шаблон поведения, наделяющий его способностью взаимодействовать с другими сущностями, в роли которых могут выступать логические объекты агентов, клиентов или сервера СУИ.

Поведение сущностей определяется *действиями* – набором функций, которые определяют работу сущности и протокол взаимодействия с другими сущностями. В первом случае инициализируется библиотека функций, которая контролирует и определяет внутреннее поведение агента. Во втором – определяется стратегия внешнего взаимодействия и правила передачи и обмена информацией. Взаимодействие различных сущностей возможно только после установления доверительных отношений между ними.

Отношения определяют возможность взаимодействия между собой агентов и могут быть двух видов: доверительные и отношения контроля.

Доверительные отношения устанавливаются при необходимости взаимного обмена информацией. Такие отношения используются между группой агентов в случае общего мониторинга и анализа состояния комплексных объектов ИТ-инфраструктуры.

Отношения контроля устанавливаются между двумя сущностями и позволяют одной из них (контролеру) запрашивать и получать необходимую информацию от второй (подконтрольной) сущности. Такие отношения, в первую очередь, устанавливаются между сервером и агентами СУИ (рис. 11.11) при централизованном управлении.

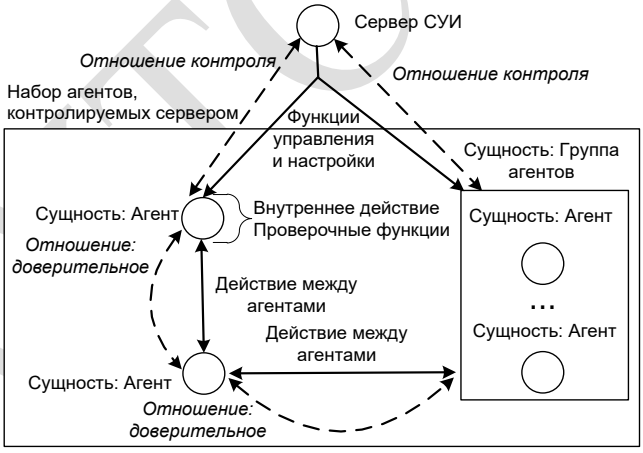


Рис. 11.11. Связи между сущностями агентов

Агент обменивается информацией с другими агентами, а также с сервером посредством *сообщений*. Сообщения позволяют передавать необходимые данные и осуществлять удаленное управление. Они могут быть двух видов: служебные и системные.

Системные сообщения генерируются СУИ для автоматического или планового обмена системной информацией и конфигурирования отдельных сущностей. Они используются для мониторинга элементов ИУС с возможностью последующего анализа.

Служебные сообщения генерируются администратором СУИ или планировщиком для удаленного управления ОМУ или получения информации о состоянии ОМУ. С помощью сообщений реализован механизм оперативного реагирования на возможные неисправности.

Для контроля функционирования отдельного агента или группы агентов используются *проверочные функции*. Эти функции могут запускаться агентом для самоконтроля или вызываться сервером. Посредством проверочных функций строятся графики работы как отдельных элементов, так и всей ИУС. Это позволяет в режиме реального времени автоматически определять возможные «узкие места» в производительности системы и перераспределять ресурсы для оперативного решения возникающих проблем. Для этого на основе статистической обработки результатов выполнения проверочных функций строятся графики изменения значений ключевых показателей работоспособности как отдельных элементов, так и ИУС в целом. Данные об изменении значений показателей позволяют автоматически выявлять неполадки в работе ИУС. Выявление неполадок приводит к подключению механизмов перераспределения ресурсов для оперативного реагирования на возникающие проблемы.

Структура агентского модуля СУИ приведена на рис. 11.12.

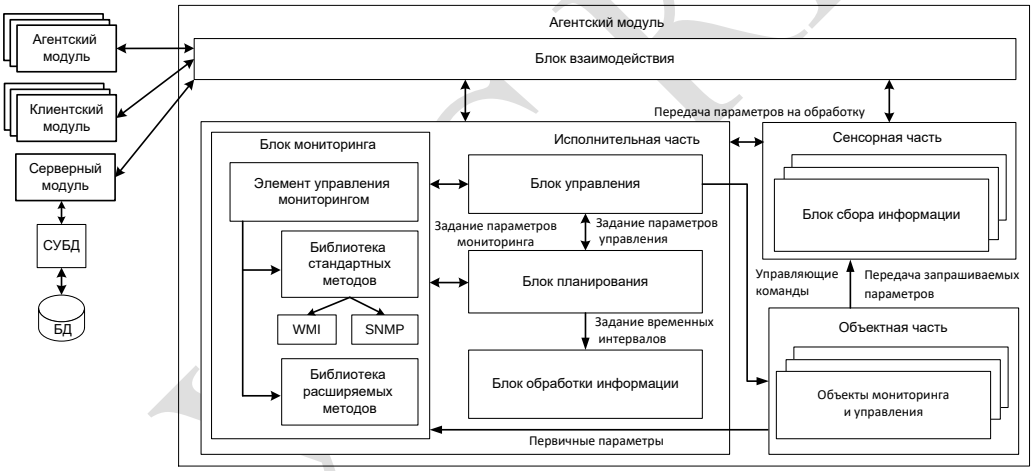


Рис. 11.12. Структура агентского модуля

Агентский модуль состоит из исполнительной, сенсорной, объектной частей и блока взаимодействия. Исполнительная часть обеспечивает штатное функционирование агента, осуществляет управление агентом, планирование действий, вызов функций мониторинга. В свою очередь, она состоит из блоков мониторинга, управления, планирования и обработки информации.

Блок мониторинга включает в себя элемент управления мониторингом, библиотеку стандартных методов и библиотеку расширяемых методов. Он обеспечивает мониторинг ЛОУ, находящихся в зоне ответственности агента. Мониторинг может осуществляться стандартными методами с использованием протоколов SNMP или ICMP, инструментария WMI или с помощью предложенной и реализованной в предлагаемой СУИ технологии DALL (Dynamic Auto Link Library) функций, использующей рефлексию типов. Стандартные средства мони-

торинга агенты СУИ используют для получения информации с локального элемента ИУС, на который установлен агент, и, при наличии доверительных отношений, с удаленных ЛОУ. Расширяемые методы мониторинга, представленные в виде DALL-функций, выбираются администратором для каждого агента из библиотеки шаблонов поведения. Каждый элемент шаблона соответствует определенному типу агента в зависимости от возложенных на него функций и состоит из набора DALL-функций.

Исполнительная часть обеспечивает управление агентом, планирование выполнения функций мониторинга и обработку информации, которая поступает от сенсорной части. При этом контролируются показатели качества функционирования отдельного элемента ИУС, на котором расположен агент. В случае выхода значений показателей за критические пределы проводится локализация потенциальной неисправности. Блок управления обеспечивает работу с деревом ЛОУ, к которому привязан агент, и может устранить возникшую, или потенциальную неисправность при наличии необходимых для устранения неисправности алгоритмов. Об этом информируется сервер СУИ, который может скорректировать принятые агентским блоком управления решения. При отсутствии необходимых для устранения неисправности алгоритмов агент уведомляет сервер СУИ о локализированной неисправности, и сервер принимает решение по ее устранению. При недоступности сервера агент принимает решения по устранению неисправностей самостоятельно.

Блок планирования обеспечивает контроль запускаемых функций проверки, непрерывный процесс мониторинга и распределяет ресурсы между всеми решаемыми агентом задачами. Кроме того, для реализации механизмов пассивного мониторинга можно подписывать DALL-функции на внешние системные события.

Блок обработки информации осуществляет обработку данных, поступающих из сенсорной части и от блока мониторинга. На основе различных модулей, в частности статистического анализа, выявляются закономерности изменения значений измеряемых параметров и определяется соответствие состояний ЛОУ предопределенным политикам, что необходимо для децентрализованного управления. Блок проводит обработку данных с учетом временных интервалов, формируемых блоком планирования. Кроме того, блок планирования формирует параметры мониторинга, которые влияют на выбор методов мониторинга ОМУ на агентском модуле. Результатом работы блока управления исполнительной части являются параметры управления, обработку которых осуществляет блок планирования.

Объектная часть состоит из дерева ЛОУ, которые закреплены за агентом. Состояние каждого ЛОУ оценивается на основе ряда параметров. Блок управления передает ЛОУ управляющие команды для поддержания контролируемых параметров в заданных пределах. Расчетные значения параметров поступают на сенсорную часть.

Сенсорная часть состоит из набора датчиков, счетчиков, указателей и других элементов, которые собирают данные для определения состояний ЛОУ, контролируемых агентом, и передает полученные данные в блок обработки информации.

Блок взаимодействия осуществляет информационный обмен агентов между собой и с сервером. Связь между агентскими модулями осуществляется напрямую при установлении доверительных отношений между агентами, или через канал связи агентов в противном случае (рис. 11.13).

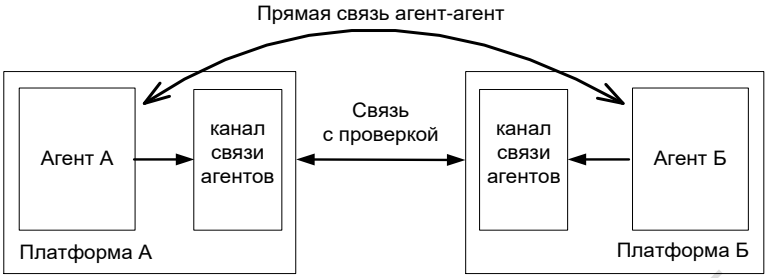


Рис. 11.13. Связи между агентами

Канал связи агентов обеспечивает связь между агентами, когда между ними нет прямых доверительных отношений. При этом создается новое сообщение для передачи, которое поступает в стек сообщений и обрабатывается планировщиком. Каждому сообщению приписывается уровень важности от 0 до 9 (0 – наивысший уровень). Сообщения с уровнем важности 0 всегда передаются напрямую независимо от уровня доверенности между агентами.

Каждый программный агент СУИ выполняет возложенные на него функции согласно этапам определенного жизненного цикла (рис. 11.14): инициализации (идентификатор 0), активности (идентификатор 1), перемещения (идентификатор 2), ожидания (идентификатор 3), паузы (идентификатор 4) и уничтожения (идентификатор 5).

Первоначальным этапом жизненного цикла программного агента является инициализация. Здесь выполняются инициализация набора функций самоконтроля, задание параметров для планировщика и загрузка дерева ЛОУ. Следующим является этап активности. Здесь агент запускает планировщик, работает с деревом ЛОУ, анализирует состояние всех составных элементов и может проводить начальный анализ полученных данных. Главной задачей этапа является контроль состояний ЛОУ и передача необходимой информации на сервер. Если за некоторый промежуток времени не происходит изменения параметров ЛОУ, агент переходит в состояние ожидания.

Планировщик отслеживает возобновление активности подконтрольных ЛОУ. Если долгое время параметры объектов не изменяются, то по указанию сервера агент может переместиться на другой РОУ, на котором ЛОУ возобновили активность, или на котором повысилась расчетная нагрузка. В этом случае агент переходит к этапу перемещения (идентификатор 2) и прекращает активность. Для этого приостанавливается внутренний планировщик и сворачивается активность дерева ЛОУ. Сервер СУИ готовит туннель для перемещения агента и служебным сообщением передает его идентификатор. По идентификатору туннеля блок управления агента находит нужный путь перемещения и мигрирует к заданному месторасположению в ИУС.

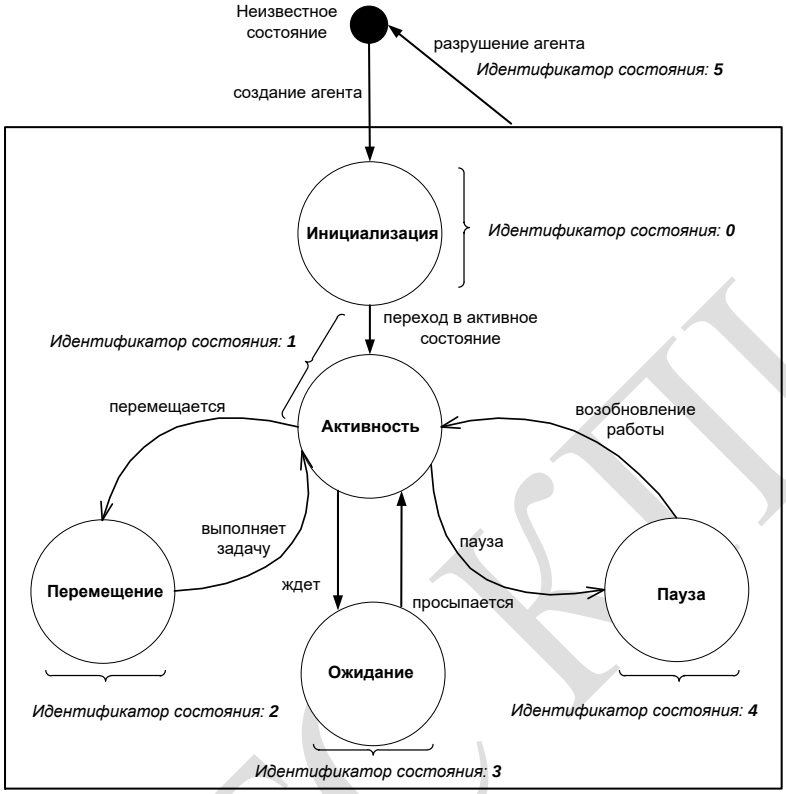


Рис. 11.14. Жизненный цикл агента

Если услуги отдельно взятого агента перестают быть полезными для СУИ, то сервер может принять решение о переходе к этапу паузы (идентификатор 4) или разрушения агента (идентификатор 5).

11.3.4. Клиентский программный модуль СУИ

Клиентский программный модуль (клиент) реализует интерфейс администратора для настройки СУИ, отслеживания работоспособности и производительности ОМУ, контроля состояния функциональных и технологических подсистем и ИУС в целом, а также для ввода команд управления при реконфигурации или проведении восстановительных мероприятий.

Непосредственно мониторинг ОМУ и реализация управляющих воздействий осуществляются агентскими и серверным модулями. Основная функция клиентского модуля состоит в предоставлении графического интерфейса для взаимодействия администратора с системой.

Структура клиентского модуля приведена на рис. 11.15.

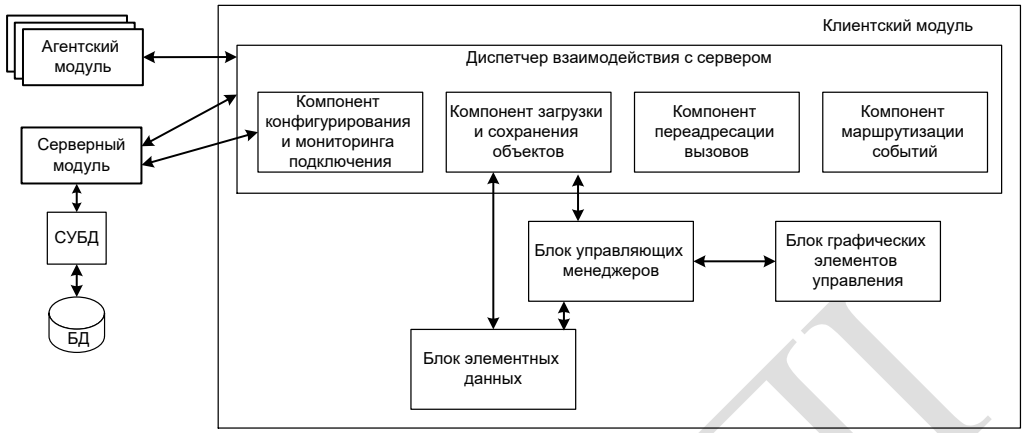


Рис. 11.15. Структура клиентского модуля

Клиент состоит из диспетчера взаимодействия с сервером, блоков графических элементов управления, управляющих менеджеров и элементных данных клиентской стороны. Диспетчер включает компоненты конфигурирования и мониторинга подключения, загрузки и сохранения объектов, переадресации вызовов и маршрутизации событий

Клиент СУИ реализован согласно шаблону программирования «Модель–Представление–Модель представления» (рис. 11.16). Этот шаблон используется для разделения модели и ее представления и позволяет отделить логику работы с данными от логики реализации графического интерфейса. Это позволяет модифицировать части ПО отдельно друг от друга, разделив работу с данными, логикой и интерфейсом программы.

Рис. 11.16. Используемый шаблон программирования

В данном случае моделью являются ОМУ, шаблоны, функции, параметры и другие объекты, с которыми работает СУИ, представлением – графический интерфейс, а моделью представления – абстракции, используемые для представления конкретных объектов.

Структурированность построения пользовательского интерфейса позволяет повторно задействовать некоторые графические элементы управления, не изменяя при этом модель данных, которая при любых обстоятельствах остается статической. Еще одним преимуществом шаблона является гибкость полученной системы в отношении тестирования качества кода. Для тестирования модели и модели представлений могут применяться классические регрессионное, поблочное тестирование и другие платформы. Таким образом, большая часть программного кода покрывается тестами, которые просты в реализации и поддержке.

Кратко рассмотрим назначение блоков клиентского модуля (см. рис. 11.15). Блок графических элементов управления является реализацией «представления» и состоит из графических компонентов, которые предназначены для отображения информации, принятой от компонента загрузки и сохранения объектов, получения ввода пользователя и передачи новых или измененных данных для их сохранения и синхронизации. Характерной особенностью этого блока является то, что он не перенасыщен программной логикой. В нем содержится только разметка внешнего вида графических элементов и методы делегирования вызовов модели представлений.

Блок управляющих менеджеров является реализацией «модели представлений». Именно этот блок отвечает за предоставление данных, которые будут отображаться в блоке предоставления графического интерфейса. От менеджера зависит, какие данные будет отображать тот или иной графический элемент и, соответственно, выбирая менеджер, можно отображать разные наборы данных в одном и том же графическом окне. Каждый менеджер связывается с компонентом загрузки и сохранения объектов для получения коллекции необходимых данных с последующем отображением на экране пользователя.

Блок элементных данных клиентской стороны вместе с компонентом загрузки и сохранения объектов реализуют «модель». Элементные данные представляют собой облегченные объекты, которые непосредственно будут отображаться на графических элементах. Этот блок необходим для представления нужных в той или иной ситуации свойств конкретного объекта данных. Компонент загрузки и сохранения объектов непосредственно предоставляет инфраструктуру для получения объектов.

11.4. ТЕХНОЛОГИЧЕСКИЕ ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИТ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ И СИСТЕМЫ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ

Реализация ИТ УИТИ потребовала воплощения технологических новаций, нацеленных на обеспечение высокого уровня требований.

11.4.1. Особенности реализации агентской технологии в СУИ SmartBase.ITS.Control

В предлагаемой СУИ мониторинг и управление ОМУ осуществляется активными агентами, являющимися отдельными программными компонентами СУИ, устанавливаемыми на РОУ. Агентом СУИ именуется вычислительный процесс, запущенный на компьютере. Агенты оперируют с ЛОУ, которые отображают состояние и параметры РОУ. Область ОМУ, которые могут обслуживаться каждым агентом, ограничивается возможностью доступа агента к РОУ.

Части дерева ЛОУ распределяются между агентами. Обычно агент обслуживает дерево ЛОУ, предоставляющее структуру ОМУ того компьютера, на котором он запущен. Кроме того, агент может обслуживать и другие ОМУ. При этом ОМУ распределяются между агентами СУИ таким образом, что каждый ОМУ обслуживается одним агентом СУИ. Для этого данный агент получает ЛОУ. ЛОУ на обслуживающем агенте посредством DALL-функций обращается непосредственно к РОУ. ЛОУ могут обрабатываться на агентах, физическое расположе-

ние которых может отличаться от физического расположения ЛОУ. К конкретным агентам привязываются только ЛОУ, управляющие и мониторинговые функции которых выполняются непосредственно на целевой машине, остальные ЛОУ могут обслуживаться любыми агентами. Это позволяет получать структуры, которые просто изменять и настраивать. Тем самым можно решить проблему поддержки слабых РОУ, которые могут не обладать достаточной производительностью и быстродействием для анализа всего дерева находящихся на них ЛОУ. Это справедливо и для проблемы безопасности, когда возникает необходимость скрыть от агента структуру дерева ЛОУ или отдельные его элементы. Другие агенты СУИ для осуществления взаимодействия между ОМУ, которые они обслуживают, и ОМУ, которые обслуживаются другими агентами, могут получить свои копии ЛОУ – ЛОУ-заместители. ЛОУ-заместители соответствуют ОМУ, обсуживающимся другими агентами, и для взаимодействия с РОУ обращаются к обслуживающим их агентам, переадресовывая им все вызовы.

Распределение ОМУ между агентами проводится в зависимости от производительности агентских РОУ и пропускной способности линий связи между ними. При этом активный ЛОУ можно перемещать между агентами не только во время конфигурирования, но и во время функционирования СУИ. Если два ЛОУ, между которыми есть функциональная связь, обслуживаются разными агентами, они являются граничными. Переопределяя граничные ОМУ, можно реализовать толстые, тонкие и мигрирующие агенты, применяя один и тот же программный модуль агента СУИ.

Для реализации мигрирующих агентов в качестве агентской платформы применяется все тот же агентский программный модуль СУИ и те же ЛОУ. Однако граница разделения в графе ОМУ может перемещаться во время работы СУИ, перенося обслуживание групп ОМУ с одного компьютера на другой. Так как в качестве ОМУ можно представить абсолютно любой процесс, то перенос его обслуживания с одного агента на другой будет означать миграцию этого процесса. При этом миграция происходит поэтапно – сначала на целевом агенте происходит загрузка и разворачивание необходимого для процесса функционала. Потом процесс сворачивает свою активность на предыдущем агенте, перемещается на целевой агент. После этого процесс вновь активизируется. Миграция может происходить как с целью разгрузки определенных агентских машин для высвобождения вычислительных мощностей, так и с целью выполнения определенных задач на определенной машине, например, диагностических, восстановительных либо обслуживающих.

Обмен данными между агентами происходит через сервер СУИ. Для обмена информацией между модулями СУИ используется технология .NET Remoting.

При подключении к серверу агентский модуль проходит процедуры установления связи с сервером и регистрации. Взаимодействие между агентами сводится к обмену служебной информацией с сервером и общению между ЛОУ.

Реализованный в СУИ механизм обновления построен на принципах реактивного мониторинга, при котором при обновлении состояния конкретного ЛОУ передаются оповещения другим ЛОУ, зависящим от него. Информационный обмен между ЛОУ проводится в виде оповещений, запросов значений параметров, передачи управляющих воздействий.

Таким образом, ЛОУ однозначно соответствует определенным РОУ. Агенты распределяются по вычислительным узлам ИУС и за каждым агентом закрепляется определенное количество ЛОУ. При этом ЛОУ, обслуживаемые данным

агентом, не обязаны ассоциироваться с теми РОУ, на котором установлен агент. Агенты, используемые в качестве органов управления и сенсоров, объединяются в распределенную вычислительную сеть по определению состояний ЛОУ. Причем каждый агент может определять состояния любого ЛОУ, и ЛОУ могут свободно перемещаться между агентами.

Агенты могут быть вынесены за пределы, например, сервера приложения, с тем, чтобы не создавать процессору сервера дополнительную нагрузку решением задач мониторинга и анализа ЛОУ. Реализация предлагаемого подхода позволяет также разделить серверный агент на несколько РОУ, что позволяет существенно разгрузить сервер, распараллелить обслуживание возложенных на него ЛОУ и тем самым повысить надежность СУИ.

Упрощенный пример распределения дерева ЛОУ между агентами отображен на рис. 11.17. Видно, что тонкий агент обслуживает только те ОМУ, для обслуживания которых необходимо их присутствие на целевой машине, а толстый агент – все логические ЛОУ своей машины. Соответственно, региональный серверный агент обслуживает ЛОУ своей сети, главный серверный агент – ЛОУ всей системы.

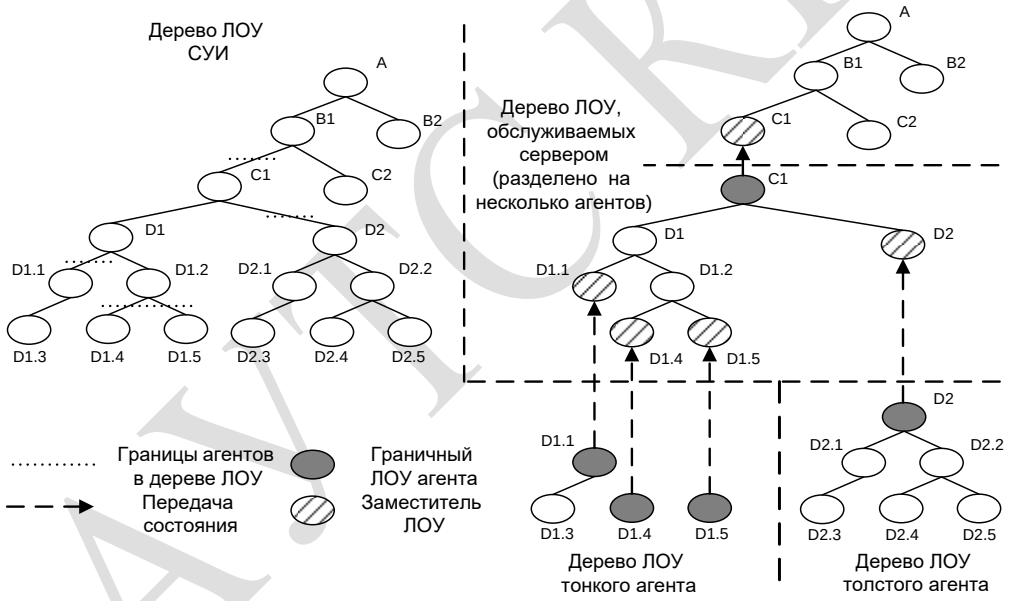


Рис. 11.17. Пример распределения дерева ЛОУ по агентам

Предлагаемый подход позволяет легко реализовать толстые, тонкие и мигрирующие агенты, применяя один и тот же программный модуль, но по-разному задавая граничные ЛОУ. Тонкий агент занимается только съемом данных мониторинга с целевой машины и выполнением управляющих функций на целевой машине, а анализ и принятие решений проводятся на сервере СУИ. В мигрирующих агентах граница разделения в дереве ЛОУ перемещается во время работы СУИ, перенося группы ЛОУ с одной машины на другую.

Для реализации предложенного подхода к распределению ЛОУ между агентами необходимо, чтобы каждый конкретный ЛОУ, независимо от его физического расположения, логически обрабатывался только одним агентом, и в то же

время обеспечивалась оперативная передача информационных и управляющих сигналов между граничными ЛОУ разных агентов. Для этого введены понятия оригинального ЛОУ и его заместителя (копии) соответственно для ЛОУ, которые обрабатываются на данном агенте, и тех, которые необходимы на данном агенте для обеспечения взаимодействия с оригинальными ЛОУ. Объекты-заместители, используя встроенный механизм переадресации вызовов, передают запросы оригинальному ЛОУ, а также синхронизируют свое состояние в соответствии с принятыми событиями.

11.4.2. Программное представление объектов мониторинга и управления

Каждый элемент ИУС представляется как ОМУ (рис. 11.18,*а*), в котором отображается информация о текущем состоянии и конфигурации данного элемента ИУС. С точки зрения мониторинга, оценки состояния и управления ОМУ ИУС удобно представлять в виде графа (рис. 11.18,*б*), узлами которого являются ЛОУ, а связи отображают функциональные зависимости.

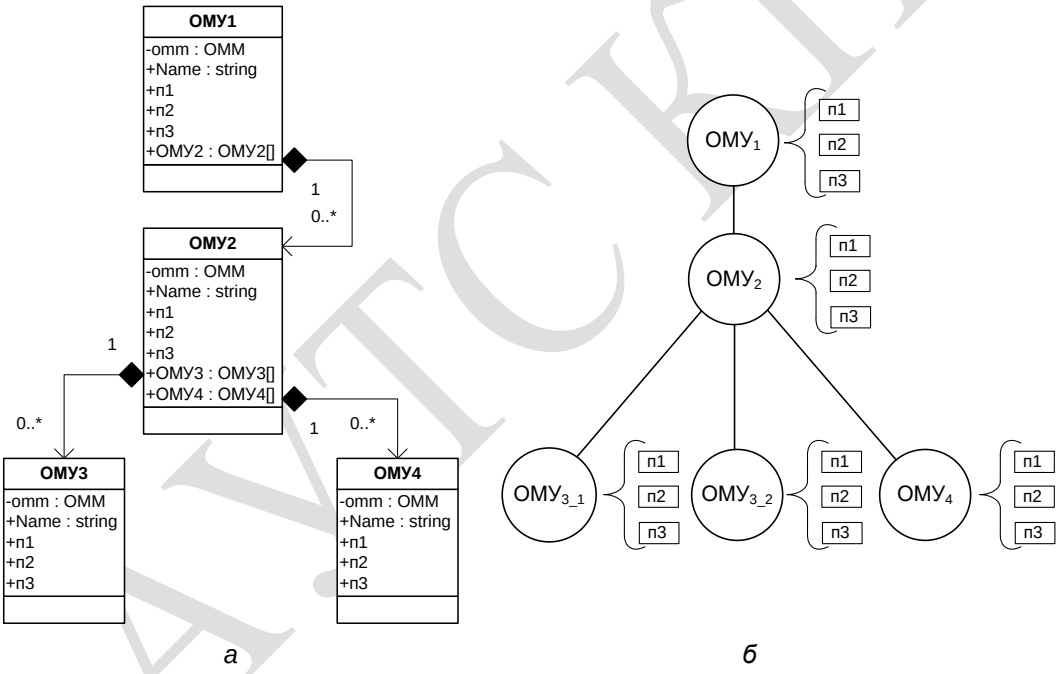


Рис. 11.18. Фрагмент иерархии ОМУ (*а*) и соответствующий фрагмент графа ОМУ (*б*)

Параметры ЛОУ характеризуют различные свойства объекта и могут быть конфигурационными и информационными. Значения конфигурационных параметров динамически не изменяются и характеризуют конфигурацию элементов и их технические характеристики. Значения информационных параметров обновляются в процессе мониторинга и могут быть использованы при анализе и расчете состояния ЛОУ.

ОМУ объединяются в иерархическую структуру и отображают связи между РОУ. Связи между ОМУ могут быть двух типов:

АУТСКИ

• Contains – связь типа родитель-потомок, один ко многим, означает непосредственное нахождения одного объекта внутри другого (только один родительский элемент по такой связи). Примерами такой связи может быть размещение устройств внутри сервера, серверов в стойке, этажей в здании и т. п.;

• Influence – описывает любую связь между ОМУ, которая не может быть описана простой иерархией (отношение родитель-потомок), представляя собой отношение многих ко многим. Например, таким видом связи представляют сетевые топологии.

В графе ЛОУ могут присутствовать одинаковые или схожие ветви. При построении дерева администратор будет тратить время на дублирование информации, поэтому целесообразно осуществлять стандартизированную обработку однотипных ЛОУ. Для этого однотипные ЛОУ объединены в шаблоны, которые определяют набор параметров, функций и функциональных связей, методов оценки состояния и управления ЛОУ данного шаблона. Шаблонный граф является прототипом графа ОМУ и он априори определяет структуру и возможные варианты его построения.

Механизм шаблонов позволяет настраивать поведение и структуру шаблонного ЛОУ и при создании новых ЛОУ использовать настройки шаблона, при необходимости уточняя и дополняя их. При этом можно легко изменять поведение и структуру группы ЛОУ путем изменения настроек их шаблона. Общие настройки для объектов, такие как зависимости логик и параметры функций, хранятся внутри шаблонов, что существенно уменьшает время переконфигурации. Система шаблонов накладывает ограничение на дерево ОМУ, не позволяя соединять между собой объекты, не предусмотренные шаблоном. Это позволяет обеспечить контроль над структурой данных. В терминах объектно-ориентированного программирования шаблон можно рассматривать как класс объектов, а ОМУ – как конкретный объект. Всегда можно заранее определить все отношения между классами объектов, которые будут повторяться в конкретных реализациях объектов этих классов. Примером может служить шаблон ЦПУ и конкретный процессор, установленный в конкретном компьютере, как ОМУ.

Ряд механизмов взаимодействия модулей СУИ инкапсулированы непосредственно в программном отображении ЛОУ. Это значительно упрощает работу с объектами в различных программных модулях и позволяет взаимодействовать с удаленными объектами посредством их локальных копий-заместителей, а также выполнять описанные ниже действия без использования отдельных блоков на каждом из программных компонентов СУИ. Структура ЛОУ приведена на рис. 11.19.

Основными составными элементами ЛОУ являются блоки функций ЛОУ, информационного обмена, параметров ЛОУ, DALL-функций.

Блок функций ЛОУ выполняет такие функции:

а) управления ЛОУ, правами на запуск которых обладают сами объекты мониторинга и управления (самоуправление), ЛОУ высшего уровня, для автоматического управления дочерними ЛОУ, администратор при ручном управлении элементами ИУС;

б) обработки данных от датчиков или дочерних ОМУ. Обработанная информация используется для мониторинга и анализа работы элементов ИУС. Это позволяет определять такие параметры как качество работы, надежность, устойчивость, различные характеристики и состояние элементов.

Блок информационного обмена обеспечивает обмен данными между ЛОУ.

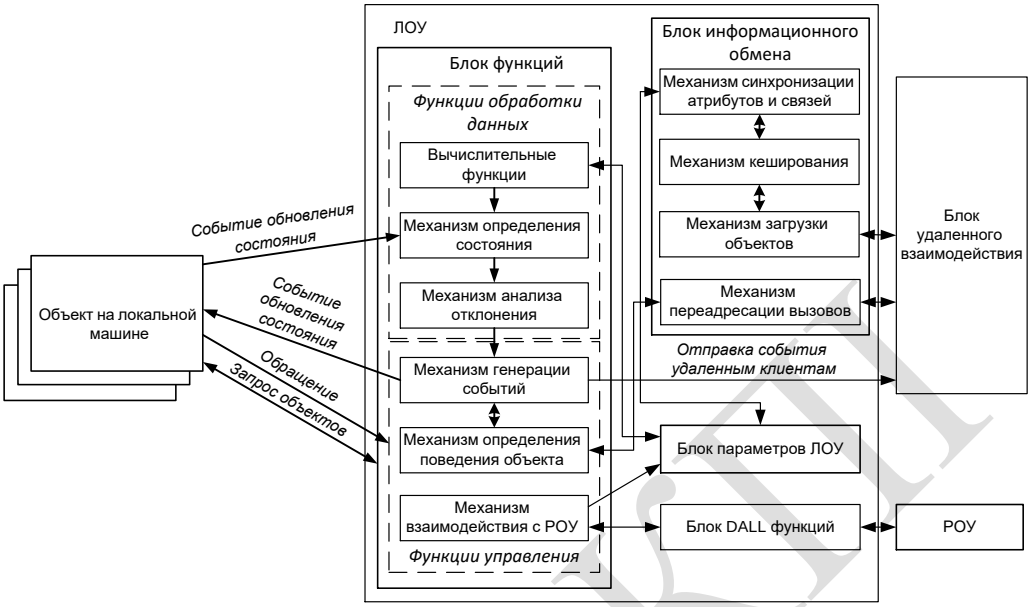


Рис. 11.19. Структура ЛОУ и инкапсулированные в ЛОУ механизмы

В программном отображении ЛОУ для обеспечения оперативного взаимодействия с другими ОМУ и контроля значений собственных параметров и состояния ЛОУ инкапсулированы следующие механизмы (см. рис. 11.19):

• синхронизации атрибутов и связей, который осуществляет синхронизацию текущей версии ЛОУ с обновленной, полученной с сервера, либо наоборот, сохраняемой клиентом, для этого отслеживает изменения параметров ОМУ и обеспечивает актуальность информации передаваемой ОМУ более высокого уровня;

• кэширования, обеспечивающий хранение используемых на данной машине ЛОУ и наличие не более одной копии объекта в системе. В случае получения новой версии уже существующего объекта происходит его синхронизация;

• определения поведения объекта, определяющий, как будет вести себя объект – как оригинальный или как заместитель. Для этого выясняется, является ли копия объекта на данной машине оригинальной (оперирующей с ЛОУ напрямую) или заместителем (переадресовывающим команды на оригинальный удаленный ЛОУ в зависимости от истинного расположения отображаемого им ЛОУ) и обеспечивается в системе наличие не более одной копии каждого объекта;

• определения состояния, который определяет текущее состояние ЛОУ в зависимости от значений мониторинговых показаний или состояний влияющих на него ЛОУ, используя различные функции из блока функций ОМУ. Это блок анализирует динамику изменения параметров ОМУ, обеспечивает оперативное реагирование на ухудшающую динамику;

• анализа отклонения, принимающий решение о необходимости уведомления зависимых ЛОУ об изменении своего состояния, при этом используется свойство гистерезиса;

• генерации событий, который сообщает зависимым ЛОУ-подписчикам об изменении своего состояния;

• переадресации вызовов, который для объектов-заместителей переадресует вызов управляющих функций на сервер, где определяется местонахождение оригинального объекта;

• загрузки объектов, который динамически подгружает необходимые объекты, связанные с данным, используя в качестве источника локальную или удаленную (серверную) базу данных.

11.4.3. Особенности программной реализации СУИ

DALL-функции. Для связи ЛОУ с РОУ используются DALL-функции, с помощью которых осуществляется как мониторинг параметров и состояния РОУ, так и управляющие воздействия на них.

DALL-функциям приписывают набор входных и выходных параметров, привязанных к параметрам ОМУ. С помощью планировщика их выполнение можно запускать по расписанию, с определенной периодичностью. Система поддерживает как набор зарегистрированных функций (имена, наборы параметров), так и возможность их добавления к шаблонам ОМУ. После добавления шаблонной функции появляется возможность привязать параметры функции к шаблонным параметрам, или явно задать значение (нет необходимости создавать параметры ОМУ для всех параметров DALL-функций). Затем настраиваются параметры запуска функций планировщиком.

Если функции определены в шаблоне, то для ОМУ создаются соответствующие функции ОМУ. Они позволяют переопределить настройки планировщика, отключить определенные параметры управления ОМУ с помощью функции или задать значение по умолчанию. Кроме того, для одной шаблонной функции можно создать несколько функций ОМУ с разной конфигурацией и потом включать/выключать их по мере необходимости.

Функции ОМУ также определяют отображение параметров ОМУ на параметры конкретных DALL-функций, осуществляют их передачу в функцию перед запуском, запуск DALL-функций и отображение результирующих параметров на параметры ОМУ.

DALL-функции назначаются шаблоном ОМУ. В его настройках задается привязка параметров функции к параметрам шаблона, условия запуска функции планировщиком, в том числе интервал запуска. Также каждый ОМУ может переопределять поведение функции, уточняя время запуска или параметры функции.

Программирование DALL-функций. Механизм DALL-функций реализован на базе технологии .Net. Для разработки DALL-функций необходимо подключить динамическую библиотеку DALL. Реализация требуемого функционала с учетом преимуществ объектно-ориентированного программирования, в частности наследования, позволяет легко учесть структурные, технологические и иные особенности объектов управления. Так, в DALL-функциях, используя соответствующие атрибуты, можно явно указать наборы входных и выходных параметров. Для подключения DALL-функцию необходимо загрузить в системную БД, после чего подключить к соответствующим ЛОУ и сконфигурировать. Поскольку DALL-функции поставляются в виде отдельных библиотек, можно использовать перезагрузку для учета особенностей взаимодействия с конкретным РОУ.

Для эффективной реализации функции можно создавать несколько классов, наследующих класс DALL_Function. В одной библиотеке может быть несколько

функций и любое количество вспомогательных классов. Полное имя функции – [*Имя_библиотеки*].[*Имя_функции*].

Пример описания функции, используемой для тестирования механизма событий:

```
   public class Random : DALL_Function
    {
        #region Params
        private int maxIntVal = Int32.MaxValue;
        private int minIntVal = 0;
        [DALL_Parameter("MaxIntVal", DALL_ParameterType.Input, "Максимальное
значение числа, которое необходимо вернуть. По умолчанию =2147483647")]
        public int MaxIntVal
        {
            get { return maxIntVal; }
            set { maxIntVal = value; }
        }
        [DALL_Parameter("MinIntVal", DALL_ParameterType.Input, "Минимальное
значение числа, которое необходимо вернуть. По умолчанию =0")]
        public int MinIntVal
        {
            get { return minIntVal; }
            set { minIntVal = value; }
        }
        [DALL_Parameter("ReturnIntVal",          DALL_ParameterType.Output,
"Случайное число в диапазоне [MinIntVal;MaxIntVal]")]
        public int ReturnIntVal { get; set; }
        [DALL_Parameter("ReturnDoubleVal",       DALL_ParameterType.Output,
"Случайное число в диапазоне [0.0;1.0]")]
        public Double ReturnDoubleVal { get; set; }
        #endregion Params
        System.Random rand = new System.Random();
        #region Overrides of DALL_Function
        public override string Name
        {
            get { return "Random.Random"; }
        }
        public override string Description
        {
            get { return "Модуль Random"; }
        }
        protected override bool _Run()
        {
            ReturnIntVal = rand.Next(maxIntVal - minIntVal) + minIntVal;
            ReturnDoubleVal = rand.NextDouble() * (maxIntVal - minIntVal) +
minIntVal;
            return true;
        }
        #endregion Overrides of DALL_Function
    }
```

Созданные библиотеки функций помещаются в папку ./DLLs внутри папки установки агента.

Тестирование DALL-функций. Для отладки и тестирования DALL-функций разработан эффективный инструментарий – программа DALL Tester. Как и агент, она загружает библиотеки из папки ./DLLs и позволяет их выполнять независимо от выполнения агента. Этот инструментарий используется для отладки функций

Аутсп

на этапах разработки и модификации СУИ. Основное окно DALL Tester приведено на рис. 11.20. Слева отображается список загруженных функций, посредине – список параметров выбранной функции и их текущие значения.

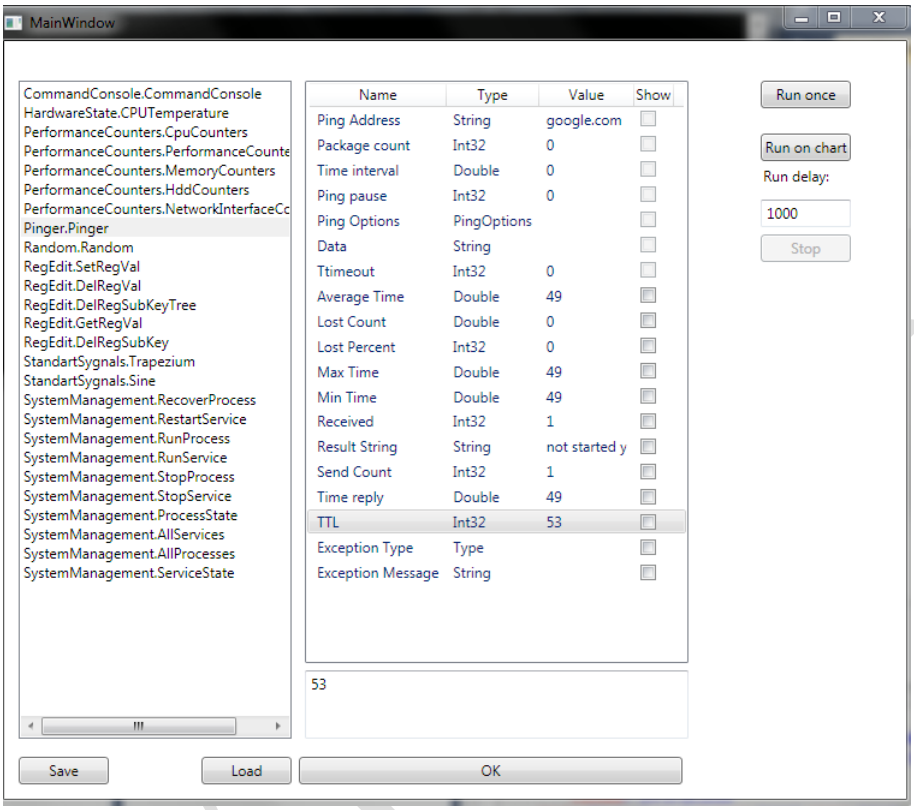


Рис. 11.20. Основное окно инструментария тестирования DALL-функций

Например, при работе с функцией ping необходимо выбрать параметр Ping Address и ввести в текстовое поле внизу адрес, потом нажать «Ok». Таким образом проводится начальная конфигурация функции при ее тестировании (рис. 11.21).

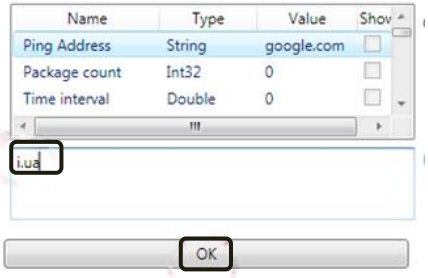


Рис. 11.21. Задание значений параметров для тестирования функций

В результате параметр для выполнения команды «ping» получит новое значение (рис. 11.22).

Name	Type	Value
Ping Address	String	i.ua
Package count	Int32	0

Рис. 11.22. Новое значение параметра Ping Address

Для единичного запуска функции и просмотра результата необходимо нажать «Run». Результат отображается в таблице значений параметров. Чтобы функция запускалась в цикле, а результат отображался в виде графика (рис. 11.23), нужно отметить параметры, которые необходимо выводить на график, затем установить время задержки между вызовами в мс, по умолчанию – 1000 мс, и нажать кнопку «Run on chart» (рис. 11.24).

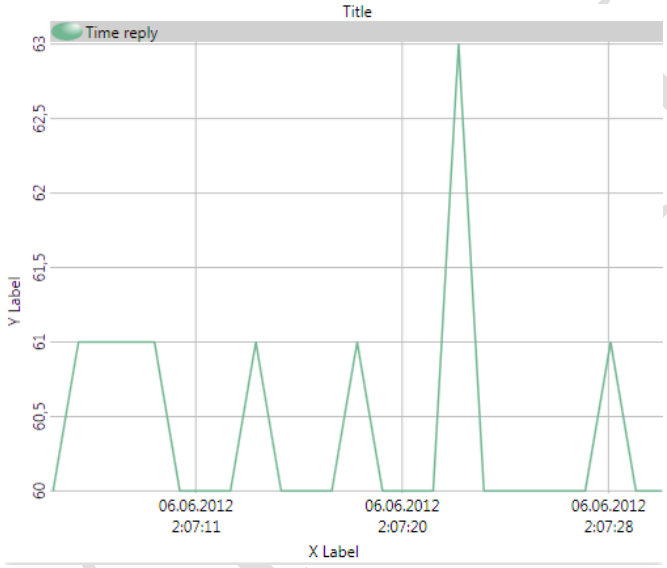


Рис. 11.23. График результатов тестового выполнения функции

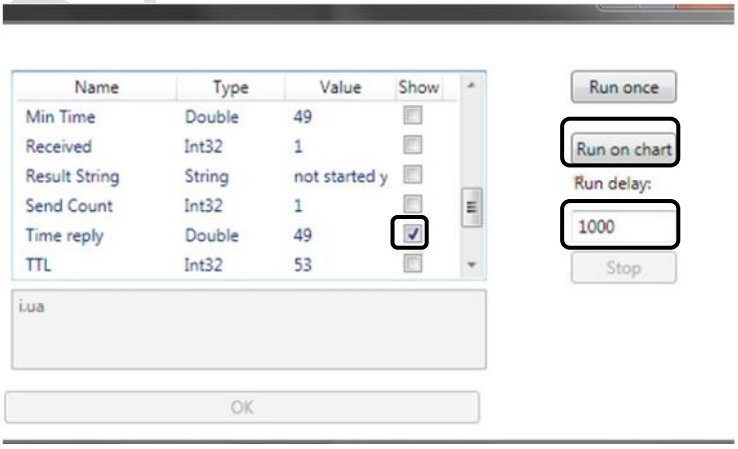


Рис. 11.24. Ввод значений параметров для отображения результатов тестирования в виде графика

11.4.4. Механизм реактивного обновления состояний ОМУ

Методы оценки состояния ОМУ. Результатом выполнения функций мониторинга и наблюдения за объектами СУИ являются числовые значения. Например, функция мониторинга загруженности процессора выдаст числовое значение доли занятости процессора в данный момент времени. Проблема заключается в том, что для управления ИТ-инфраструктурой требуется большая наглядность. Администратору сложно делать вывод о работе объекта мониторинга, основываясь только на числовом представлении его параметров.

Поэтому в СУИ вводится понятие состояния ОМУ. Состояние – это качественная оценка, морфологическая интерпретация числовых значений параметров ОМУ в данный момент времени. Администратору намного проще составить оценку работы конкретного ОМУ, руководствуясь не только числовыми значениями его параметров, а и состоянием. В СУИ реализован ряд методов оценки, которые на основе числовых значений параметров ОМУ дают качественную оценку их текущему состоянию. К ним, в первую очередь, относятся расчетные методы, механизм границ состояний и механизм копирования состояния от объекта, который находится ниже по иерархии.

Метод границ состояний. Каждое состояние характеризуется интервалом допустимых значений, которые должен принимать параметр ОМУ, чтобы находится в этом состоянии. Множество таких интервалов должно охватывать все возможные значения, которые может принимать параметр ОМУ. Каждый интервал характеризуется описанием состояния, которое соответствует этому интервалу, и верхней границей допустимых значений параметра. Нижней границей считается верхняя граница предыдущего интервала, либо минус бесконечность, если интервал первый.

Состояние определяется следующим образом: 1) по очереди выбирается каждое допустимое для данного параметра состояние, для которого конструируется интервал значений параметров ОМУ; 2) проверяется, попадает ли значение текущего параметра в полученный интервал: если да, то считается, что параметр ОМУ находится в выбранном состоянии, иначе – переходим к следующему состоянию; 3) состояние определяется путем проверки принадлежности значения текущего параметра интервалу, соответствующему каждому допустимому для данного параметра состоянию.

Опыт использования ИТ УИТИ показал, что данный метод очень удобен для оценки состояния параметров ОМУ, значения которых непосредственно снимаются с РОУ.

Расчетные методы. Эти методы приводят полученные значения к взвешенным, используя разнообразные алгоритмы расчета – среднего арифметического, среднего геометрического, взвешенной суммы, взвешенной нормированной суммы, среднего арифметического взвешенного, среднего гармонического взвешенного. Эти методы позволяют при расчете состояния для текущего параметра учитывать показатели зависимых параметров. Возможны два варианта расчета – по числовым значениям и состояниям параметров.

При первом варианте выбирается все множество зависимых параметров и к их значениям применяется один из приведенных выше алгоритмов. Каждое значение, которое используется для расчета средневзвешенного, имеет свой вес, который показывает степень влияния каждого параметра на определение резуль-

тата. К результирующему средневзвешенному значению применяется метод границ состояний. Таким образом, результатом является состояние текущего параметра ОМУ.

Второй вариант предусматривает учет состояний зависимых параметров для расчета состояния текущего параметра (метод кодирования состояния предложен в главе 5). С этой целью каждому состоянию присваивается условное значение –так называемый уровень приоритета данного состояния. Тогда средневзвешенное значение считается по уровням приоритета текущих состояний зависимых параметров ОМУ. Полученное значение соотносится с одним из состояний из множества допустимых для данного параметра. Если вычисленное значение находится на границе раздела двух состояний, используется один из режимов разрешения неоднозначности определения состояния. Оптимистичный режим выберет состояние с лучшими показателями, пессимистичный – состояние с худшими показателями.

Метод копирования состояния у объекта, находящегося ниже по иерархии. Этот метод используется для ОМУ, у которых есть только один дочерний параметр. Он переносит состояние зависимого объекта на текущий объект.

Метод расчета состояния для параметра ОМУ может выбираться для конкретного параметра ОМУ, либо может быть использован механизм, заданный в шаблоне параметра ОМУ.

Определение момента времени перерасчета состояния. Поскольку в ИУС состояние ЛОУ N-го иерархического уровня зависит от состояния ЛОУ $(N-1)$-го уровня, необходимо обновлять состояние ЛОУ вышерасположенного уровня при изменении состояния ЛОУ нижерасположенных уровней. При этом возникает задача определения необходимой частоты обновления состояний всех ЛОУ. Существует два похода к решению этой задачи:

- пересчет состояний ОМУ по указанию планировщика;
- реактивное обновление состояний.

Планировщик задач. Как отдельный блок агентского/серверного модуля СУИ планировщик осуществляет запуск процессов мониторинга и других задач, выполняемых различными модулями по расписанию.

Задание, отслеживаемое планировщиком, представляется в СУИ в виде записи, которая определяет периодичность запуска задания и требуемое количество запусков. Задание может быть настроено так, чтобы запускаться на выполнение один раз в заданное время, определенное количество раз с указанием конкретных значений времени выполнения задачи или промежутка времени между выполнениями, либо выполняться непрерывно (настраивается как «–1 раз») с заданной периодичностью. В процессе работы планировщика вычисляется и запоминается время последнего запуска задания и ожидаемое время следующего запуска.

Поведение планировщика зависит от значения периода запуска задания:

- для небольших промежутков (меньше 15 мин) для каждого задания создается новый поток выполнения, в котором выполняется сама задача, после чего проверяется значение следующего времени запуска, и поток ожидает нужный промежуток времени;
- для больших промежутков времени задания помещаются в отдельный список, который обслуживается специальным потоком редко запускаемых заданий,

при этом анализируется время следующего запуска ближайшего задания, и ожидается необходимый промежуток времени, после чего задание запускается.

Параметры заданий, обслуживаемых планировщиком, можно изменять, а также задачи можно запускать вручную.

При пересчете состояний ЛОУ по указаниям планировщика каждому ЛОУ N-го уровня необходимо задавать период обновления, а при пересчете использовать текущие значения состояний ЛОУ $(N-1)$-го уровня, частота обновления которых может отличаться от частоты опроса ЛОУ N-го уровня. Недостатки такого метода очевидны – сложность синхронизации обновления сложных иерархий во времени, достаточно медленное реагирование на изменение состояния ЛОУ $(N-1)$-го уровня и использование устаревших значений при низкой частоте опроса. Последний недостаток особенно критичен, поскольку потенциально может привести к выдаче ложной информации конечному пользователю. Для компенсации этого недостатка осуществляют повышение частоты опроса. Однако это приводит к перезагруженности каналов связи служебной информацией, повышению уровня требований к вычислительным ресурсам. Поэтому для СУИ разработан и реализован механизм реактивного обновления состояний ЛОУ.

Определение периода опроса ОМУ. В зависимости от периода опроса ОМУ можно разделить на три категории: с заданным периодом; с периодом, определяемым экспериментально; с адаптивным периодом.

В первом случае период опроса ОМУ задается администратором в виде двух значений: первое определяет период опроса для нормальной работы ОМУ; второе – при аварийном или граничном значении состояния ОМУ. При этом администратор руководствуется своими знаниями и опытом. Оба значения периода могут изменяться в процессе работы администратором. При таком задании периода опроса данные мониторинга создают существенный сетевой трафик.

Во втором случае период опроса подбирается СУИ на основе результатов анализа динамики изменения состояния ОМУ. Сначала проводятся измерения с большой частотой и определяется динамика изменения состояния ОМУ. После этого частота опросов уменьшается до минимального значения, не искажающего тренды изменения состояния ОМУ. Механизм запускается по команде администратора. Данный способ существенно усложняет реализацию ПО планировщика СУИ, а также требует дополнительных вычислений в процессе его работы.

В третьем случае СУИ адаптируется под динамику изменения состояния ОМУ и в автоматическом режиме определяет и модифицирует значения периода опроса. Так, если к ОМУ длительное время отсутствуют обращения (такая ситуация возможна, например, когда ОМУ является резервной системой хранения данных), то период опроса параметров ОМУ уменьшается. Если же к ОМУ начинают обращаться часто, то период опроса автоматически увеличивается.

Поскольку рассмотренные подходы не перекрывают все разнообразие условий функционирования ОМУ, предложен еще один способ обновления состояний, основанный на реактивном обновлении.

Реактивное обновление. Суть реактивного обновления заключается в управлении планировщиком периодом обновления состояний только для ЛОУ нижних уровней, состояние которых не зависит от состояний других ЛОУ, а

определяется только значениями параметров, определяемых при мониторинге. При этом ЛОУ верхних уровней реагируют на поток событий об изменении состояний ЛОУ нижних уровней, влияющих на состояние рассматриваемого ЛОУ, и не нуждаются в использовании планировщика. При обновлении состояния ЛОУ (N-1)-го уровня генерирует событие, на уведомление о котором подписываются зависящие от него ЛОУ N-го уровня. Механизм анализа отклонения ЛОУ обладает свойством гистерезиса [123]. Это предотвращает генерацию события, излишне частые срабатывания и остановку волны перерасчета состояний ЛОУ вверх по дереву, если состояние иерархически нижерасположенных ЛОУ изменилось незначительно. Кроме того, для пассивного мониторинга в СУИ реализована возможность подписки функций ЛОУ на внешние системные события.

Механизм реактивного обновления обеспечивает своевременное обновление состояний всех ЛОУ, уменьшает количество избыточных пересчетов состояния верхних уровней иерархии ЛОУ, позволяет гибко настраивать чувствительность системы к изменениям состояний ЛОУ.

11.4.5. Протоколы взаимосвязи, используемые в СУИ

Взаимодействие агентских, клиентских и серверных модулей СУИ обеспечивает реализацию функций прикладного и представительного уровней модели OSI. Непосредственно это обеспечивается компонентами загрузки и сохранением объектов, переадресацией вызовов и маршрутизацией событий, находящихся в модулях СУИ, а также серверным блоком разграничения доступа.

Взаимосвязь между модулями СУИ осуществляется посредством программных компонентов .NET Remoting, которые, в свою очередь, используют для транспорта данных потоки TCP. При этом запросы к удаленным объектам в среде .NET Remoting не отличаются от локальных вызовов. Операции преобразования структур данных объектов в среде между клиентскими и серверными частями приложения выполняются автоматически.

Протокол взаимосвязи между модулями СУИ и ЛОУ. Контролируемые агентами ЛОУ являются самостоятельными активными элементами СУИ. Для общения между собой они используют агентские модули. В свою очередь, агентские модули составляют исполнительную платформу для ЛОУ. ЛОУ, находящиеся на клиентском и серверном модулях, не являются активными элементами СУИ. Они могут лишь принимать обновленные данные от соответствующих активных ЛОУ и передавать им задаваемые пользователем системы управляющие воздействия. Для общения между собой они используют необходимые блоки представительного уровня соответствующих модулей. Иерархия компонентов, обеспечивающих взаимосвязь между модулями СУИ, приведена на рис. 11.25.

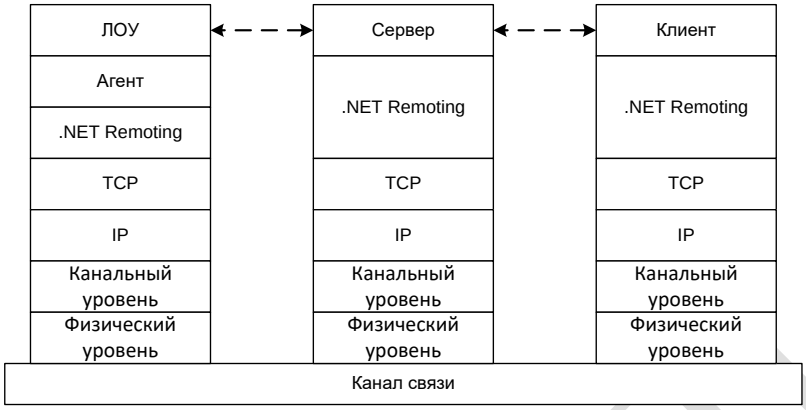


Рис. 11.25. Иерархия компонентов взаимосвязи в СУИ

Стек обработчиков сообщений. Технология .NET Remoting использует стек обработчиков сообщений для выполнения ряда действий при передаче сообщений, связанных с авторизацией клиентов, упаковкой передаваемых данных и др.

Передача запроса с обслуживаемого модуля на сервер. Передача запроса с обслуживаемого модуля на сервер осуществляется с помощью следующих обработчиков:

Client	**Server**
ClientThreadChannelSink	ServerThreadChannelSink
ClientChanelSink1	ServerUpdateChanelSink
ClientChanelSink2	ServerChanelSink
BinaryClientFormatterSinkProvider	BinaryServerFormatterSinkProvider

Кратко опишем назначение этих обработчиков:

ClientThreadChannelSink обеспечивает создание отдельного потока для вызова;

ClientChanelSink1 упаковывает данные перед отправкой запроса, распаковывает и обеспечивает синхронизацию данных при получении ответа от сервера;

ClientChanelSink2 передает серверу данные, касающиеся авторизации клиента;

BinaryClientFormatterSinkProvider сериализирует данные для отправки;

BinaryServerFormatterSinkProvider десериализирует полученные данные;

ServerChanelSink проверяет данные авторизации клиента при получении запроса, осуществляет отклонение запросов от незарегистрированных клиентов;

ServerUpdateChanelSink распаковывает принятые данные, собирает данные для уведомления всех клиентов об изменениях в удаленных ОМУ, вызывает диспетчера обновления ОМУ, запаковывает данные контекста запроса для отправки клиенту;

ServerThreadChannelSink получает информацию о клиентском потоке, выполняющем запрос. Серверный менеджер потоков позволяет клиенту узнавать информацию о процессе работы запущенных им потоков, а также останавливать их.

Передача запроса с сервера на обслуживаемый модуль выполняется с помощью следующих обработчиков:

Server	**Client**
Server.ClientChanelSink	
ClientChanelSink1	CallbackChanelSink
BinaryClientFormatterSinkProvider	BinaryServerFormatterSinkProvider

Кратко опишем их назначение:

Server.ClientChanelSink детектирует ошибки при обращении к клиенту, в случае недоступности клиента помечает его как отключенного;

ClientChanelSink1 упаковывает данные перед отправкой в контекст запроса, распаковывает и синхронизирует данные при получении ответа;

BinaryClientFormatterSinkProvider сериализирует данные для отправки;

BinaryServerFormatterSinkProvider десериализирует полученные данные;

CallbackChanelSink распаковывает и синхронизирует данные при получении запроса от сервера.

Использование контекста запроса. При получении любого запроса обработчик сообщений (ServerUpdateChanelSink) инициализирует контекст запроса. Тем самым обеспечивается возможность использования идентификатора запрашивающего модуля для определения прав доступа, списков возвращаемых, измененных и удаленных запросом объектов, а также имя вызывавшего потока.

После окончания выполнения запроса обработчик сообщений отфильтровывает из списка возвращаемых объектов те, на которые у запрашивавшего модуля СУИ отсутствуют права доступа. Таким образом, объекты будут выданы только авторизированным модулям СУИ и упрощаются проверки в блоках загрузки объектов, таких как ОМУ, параметры, шаблоны, функции и др.

На клиентском либо агентском модулях СУИ обработчик сообщений ClientChanelSink1 при получении ответа на запрос, который содержит список возвращаемых объектов, добавляет эти объекты в кэш. Те из объектов, которые присутствовали в кэше ранее, синхронизируются со старыми копиями.

Автоматизация отложенной инициализации атрибутов. Модуль СУИ может обратиться к серверу с запросом на инициализацию атрибута. При этом сервер (ServerUpdateChanelSink) извлекает копию объекта и, если атрибут не инициализирован, инициализирует его. После этого значение инициализированного атрибута возвращается запрашивавшему модулю СУИ. Интерфейсы функций-запросов на инициализацию атрибутов ОМУ помечаются атрибутом LazyProretyInitializerAttribute с указанием индекса соответствующего атрибута ОМУ. Поэтому запросы перехватываются обработчиком сообщений.

Обработка запросов, изменяющих данные. Запросы, выполнение которых приводит к изменению данных, помечается атрибутом Modifying. После их получения и выполнения обработчик сообщений запускает работу диспетчера обновления объектов, который рассылает уведомления об измененных объектах соответствующим модулям СУИ. Запросы на удаление объектов отфильтровываются обработчиком запросов.

Протокол взаимодействия между ЛОУ и РОУ. ЛОУ, обслуживаемый текущим агентом, обращается к РОУ посредством DALL-функций. Чтобы эта функция имела возможность выполнить свою задачу, она должна иметь доступ к РОУ. Если ЛОУ для текущего агента является заместителем, то обращение передается реальному ЛОУ с использованием протокола общения между модулями СУИ и ЛОУ. Взаимодействие происходит по следующей цепи:

ЛОУ-заместитель ↔ ЛОУ ↔ DALL-функции ↔ РОУ.

Протокол взаимодействия между сервером и РОУ. Обобщая изложенное выше, схему взаимодействия сервера с РОУ при мониторинге и управлении представим в виде, изображенном на рис. 11.26.

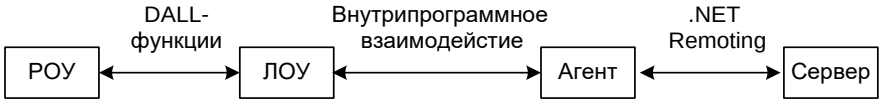


Рис. 11.26. Схема взаимодействия РОУ–сервер

Для взаимодействия компонентов СУИ кроме .NET Remoting в СУИ используется WCF, являющийся частью .NET Framework 3.0.

11.4.6. Синхронизация данных и совместная обработка

Синхронизация данных в системе делает возможной их согласованную обработку разными модулями СУИ. Для того, чтобы в СУИ существовала только одна копия объекта, при внесении изменений необходимо своевременно уведомлять об этом все модули СУИ. Синхронизация должна вызывать минимальные задержки и создавать минимальную дополнительную нагрузку на коммуникационную сеть.

В системе осуществляется два вида синхронизации:

- на уровне объектов, где выполняется локальная синхронизация двух версий объекта – хранимой в данном модуле СУИ и вновь полученной. При этом отслеживаются и сравниваются версии атрибутов ОМУ и его связей;
- на уровне системы, где серверные блоки маршрутизации событий и диспетчер обновления объектов обеспечивают комплексную согласованную обработку изменений.

Схематично работа механизма синхронизации показана на рис. 11.27.

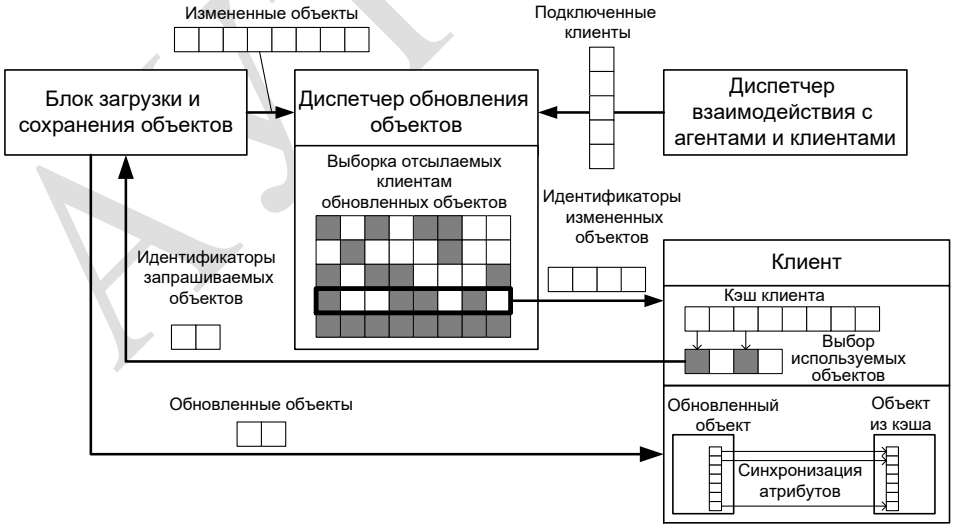


Рис. 11.27. Работа механизма синхронизации

Механизм кэширования. Механизм кэширования хранит используемые данной машиной объекты и обеспечивает наличие не более одной копии объекта в СУИ. При обращении к ОМУ сначала проверяется его наличие в кэше. И запрос направляется на сервер, либо в БД только в случае отсутствия. Благодаря этому значительно повышается скорость выполнения запросов и обеспечивается гарантированное наличие в системе не более одной копии каждого ОМУ. В случае получения новой версии уже существующего объекта выполняется синхронизация.

Синхронизация объектов. Для обеспечения целостности и актуальности данных реализован механизм, отвечающий за своевременное оповещение клиентов о произошедших изменениях структуры дерева ОМУ и его составляющих.

С целью согласования обрабатываемых данных после сохранения или удаления объектов необходимо уведомить модули СУИ об изменении данных. Для этого идентификаторы изменяемых объектов вносятся в соответствующие списки сохраненных, обновленных и удаленных объектов. Используя эти списки, хранимые в контексте запроса, диспетчер обновления объектов определяет необходимость уведомления каждого из подключенных к серверу модулей о выполненных изменениях объектов. Для принятия решения об уведомлении модулей об обновлении объектов также необходима информация о возможности доступа модулей к объектам для обработки. Например, агенты СУИ не уведомляются об обновлении шаблонов ОМУ, так как обработкой шаблонов занимаются только клиенты СУИ, агенты уведомляются только об обновлении ОМУ и их компонентов, находящихся в зоне их ответственности.

Для каждого подключенного клиента серверный диспетчер обновления объектов строит выборки объектов из списка обновленных. Построение выборок сводится к решению NP-полной задачи, поскольку решение задачи определения необходимости оповещения клиента об обновлении некоторого ОМУ требует решения аналогичной задачи для каждого из связанных с ним объектов. Чем длиннее цепи опосредованно связанных объектов, тем больше требуется вычислений. Однако необходимо заметить, что в некоторых случаях количество операций является постоянным и зависит только от типа обновленного объекта. Например, ОМУ может быть привязан к конкретному агенту и к клиентам определенного региона. Поскольку функции ОМУ принадлежат конкретному ОМУ, и чтобы определить, какие из модулей СУИ уведомляются об обновлении функций ОМУ, необходимо определить то же для их ОМУ. Это же замечание касается и параметров функций ОМУ, принадлежащих конкретным функциям ОМУ.

Определение необходимости уведомления каждого из подключенных модулей СУИ об обновлении определенных объектов, с одной стороны, порождает дополнительную нагрузку на вычислительные ресурсы сервера СУИ. С другой стороны, снижается нагрузка на коммуникационную сеть, поскольку большая часть объектов отсеивается при построении выборок обновленных объектов для каждого из модулей СУИ. Для балансировки загруженности вычислительных ресурсов и сети можно гибко настраивать механизм отбора необходимых для уведомления объектов, выполняя минимальные проверки для объектов некоторых типов (например, только проверку доступа модуля СУИ к данному объекту), либо не выполняя ее вовсе.

Для снижения сетевого трафика обслуживаемым модулям передаются не сами измененные объекты, а список их идентификаторов, после чего модуль запрашивает обновление только тех объектов, которые им используются.

Синхронизация атрибутов. Клиентский модуль, получив от сервера обновленную копию объекта, с которым он работал, вносит изменения во все атрибуты и связи объекта, кроме тех, которые он сам изменял.

Поскольку агент не имеет прав на изменение структуры объектов мониторинга и их связей, он не может вносить изменения в объекты. Поэтому при синхронизации принимается вновь полученная с сервера версия объекта.

На сервере, при получении от клиента обновленных версий ОМУ, в процессе синхронизации атрибутов и связей объектов применяются только изменения, внесенные данным клиентом. Такой механизм позволяет редактировать один и тот же объект на разных клиентах. После синхронизации все измененные ОМУ сохраняются в БД и передаются диспетчеру обновления объектов для оповещения подключенных клиентов.

Описанные механизмы синхронизации данных позволяют нескольким клиентам одновременно работать с удаленными ОМУ.

11.4.7. Обработка истории данные мониторинга

Сбор исторических данных мониторинга. Одной из основных задач агента СУИ является мониторинг состояния ОМУ. Все данные, собранные агентом, сначала хранятся в локальной БД, периодически они пересылаются на сервер, где обрабатываются и хранятся уже в серверной БД. Механизм и периодичность отправки информации с агента на сервер СУИ определяется системным планировщиком. В дальнейшем такая информация может использоваться СУИ при анализе работы системы за определенные промежутки времени. Информация мониторинга формирует большие массивы данных. Чтобы сделать ее полезной администратору, на сервере СУИ реализован механизм ее обработки.

Обработка исторических данных мониторинга. Обработка исторических данных мониторинга осуществляется сервером каждый раз при поступлении новых данных. Для того чтобы сократить время получения обработанных данных в тот момент, когда они нужны для формирования управления, данные предварительно обрабатываются при их поступлении.

Исходные данные агрегируются по уровням, на каждом из которых накапливаются данные с большей степенью усредненности. Для повышения эффективности обработки данных они хранятся в двух таблицах БД: в первой находятся непосредственные данные мониторинга, которые усредняются; во второй – информация о выделенных уровнях. Расчет значений на уровнях основывается на дате и значении параметра в записи из первой таблице, а также на информации о соответствующем уровне из второй таблицы, в частности, номер уровня, промежуток усреднения и т. д.

Рекурсивный алгоритм расчета усредненных значений.

Шаг 1. Начиная от первого необработанного за промежуток времени, указанного в соответствующей записи из таблицы уровней, берутся значения усредняемого уровня и усредняются по формуле

$$b_j = \frac{a_i + a_{i+1} + \ldots + a_{i+n-1}}{n},$$

где n – количество значений в промежутке, a_i – соответствующие значения. Полученное значение b_j записывается в БД с датой, равной началу промежутка плюс половина его значения.

Шаг 2. Обрабатываются данные следующего промежутка.

Шаг 3. Усреднение уровня заканчивается, когда в промежутке нет данных для обработки.

Полученные данные формируют структуру, похожую на пирамиду: ее ординатой является номер уровня, а абсциссой – дата.

Шаг 4. Обработка заканчивается в тот момент, когда на очередном уровне в промежутке необработанных данных окажется меньше, чем длина промежутка усреднения, либо когда алгоритм достигает верхнего уровня пирамиды. Если оставшееся количество данных на уровне меньше длины периода усреднения, на уровне остается «хвост» необработанных данных, которые используются при появлении новых данных.

Теперь при получении запроса на усредненное значение определенного параметра за определенный период быстро находится уровень, количество значений данного параметра на котором максимально приближено к оптимальному. Это позволят администратору принять взвешенное решение и задать достаточно эффективное значение.

Таким образом, обработка сохраненных данных мониторинга осуществляется с меньшими временными и ресурсными затратами. Это объясняется уменьшением предаваемых данных и принятой концепцией обработки, при которой усреднение происходит один раз при добавлении данных, а не при каждом запросе на усредненные параметры. При этом обеспечивается необходимая для корректного отображения и обработки данных точность.

Уровни усреднения могут настраиваться как автоматически, так и вручную администратором для каждого отдельного шаблонного параметра или параметра ОМУ. Настраиваемыми являются максимальное количество уровней усреднения для данного параметра, длина периода усреднения, оптимальное количество выдаваемых по запросу значений параметров.

11.5. ИНТЕРФЕЙС АДМИНИСТРАТОРА

Вход в систему. В приложении управления ИТ-инфраструктурой реализована полнофункциональный модуль авторизации, который поддерживает cookie authentication, token authentication, role-based authentication. Таким образом предоставляется авторизация на основе cookies, токенов, а также разграничение прав с помощью ролей пользователей. Главная форма входа в приложение изображена на рис. 11.28.

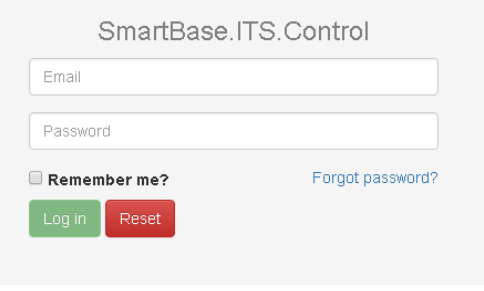


Рис. 11.28. Форма входа в систему

Если пользователь вводит электронный адрес в некорректном формате, то система отобразит ошибку ввода данных после обработки правил валидации на стороне клиента (рис. 11.29). Если введен пароль, который не соответствует базовым требованиям, то также отразится ошибка (рис. 11.30).

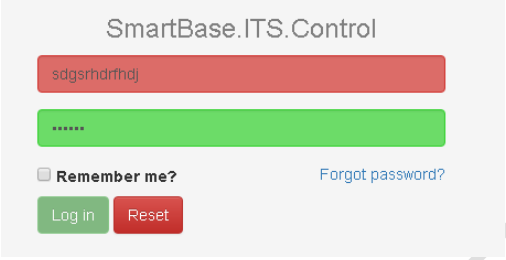


Рис. 11.29. Валидация электронной почты

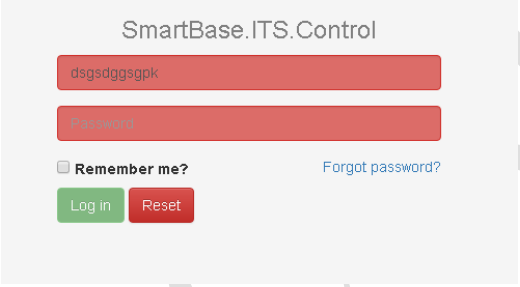


Рис. 11.30. Валидация пароля

При корректном заполнении всех полей запрос отправляется на сервер для авторизации, которая осуществляется на основе сличения введенных данных авторизации с данными, находящимися на сервере (рис. 11.31).

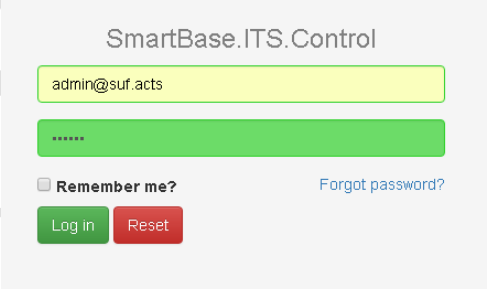


Рис. 11.31. Вид окна при корректном введение данных авторизации

Реализован механизм, позволяющий пользователю восстановить пароль (рис. 11.32).

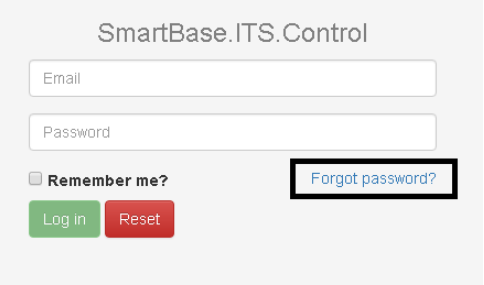


Рис. 11.32. Кнопка восстановления пароля

На рис. 11.33 изображена форма отправки запроса на восстановление пароля. После ввода адреса электронной почты осуществляется проверка соответствия введенного адреса существующему адресу, хранящемуся в базе данных. Если адрес найден, то специальный модуль системы сформирует письмо со специальной ссылкой для восстановления пароля и отправит его на почту пользователя.

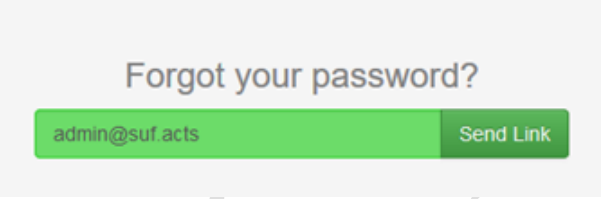


Рис. 11.33. Форма восстановления пароля

На всех вышеприведенных формах реализована многоступенчатая валидация пользовательского ввода (на уровнях frontend, backend и базы данных), что позволяет обеспечить защиту от таких типов хакерских атак, как SQL-injection, XSS (англ. Cross-Site Scripting) и др.

Управление агентами и ОМУ. Сразу после входа в систему пользователь в роли администратора получает доступ ко всем зарегистрированным агентам (рис. 11.34).

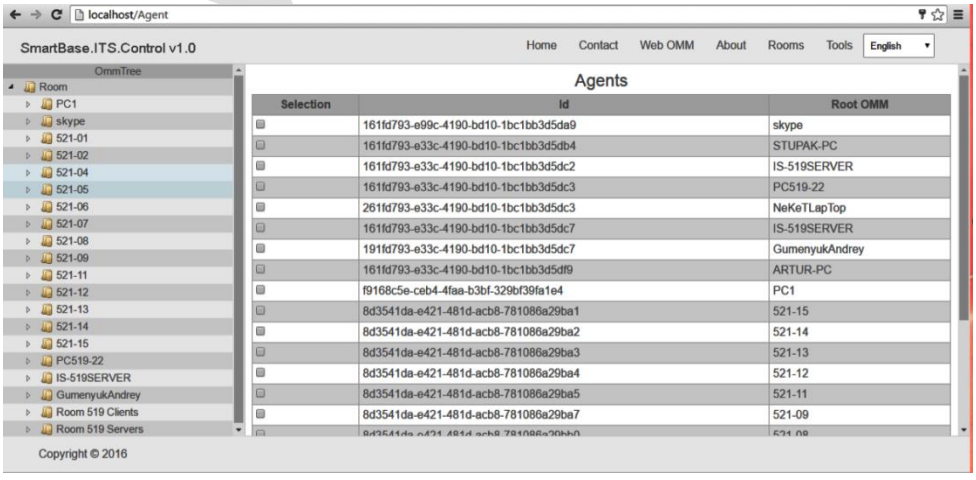


Рис. 11.34. Список агентов

Для каждого агента система хранит определенный набор метаданных, включая имя и идентификатор агента. В первой колонке содержатся поля, позволяющие отмечать агентов, которые выбраны для исполнения команд. Во второй колонке отображается идентификатор агента, который можно использовать при ручной настройке агента на целевой машине. В третьей колонке отображается название корневого элемента иерархии объектов, которые отслеживаются агентом (в данном случае оно соответствует имени машины, на которой запущен агент).

Администратор отмечает выбранные агенты, задает команду управления и значения ее параметров (рис. 11.35).

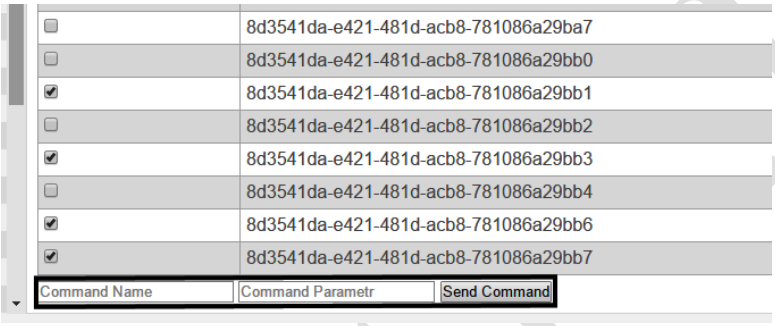


Рис. 11.35. Массовое выполнение команд

Форма для связи с разработчиками. В ПО СУИ реализован механизм, который позволяет администраторам СУИ подержать связь с разработчиками ПО СУИ, для того чтобы проинформировать последних о выявленных недостатках и высказать предложения по улучшению системы (рис. 11.36).

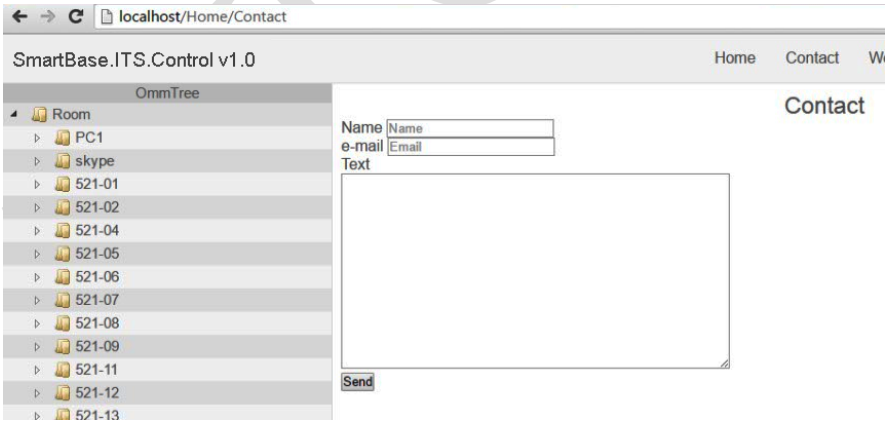


Рис. 11.36. Форма для связи с разработчиками

Форма редактирования размещения. Форма на рис. 11.37 позволяет создавать и редактировать группы агентов, исходя из физического расположении машин, на которых установлены эти агенты.

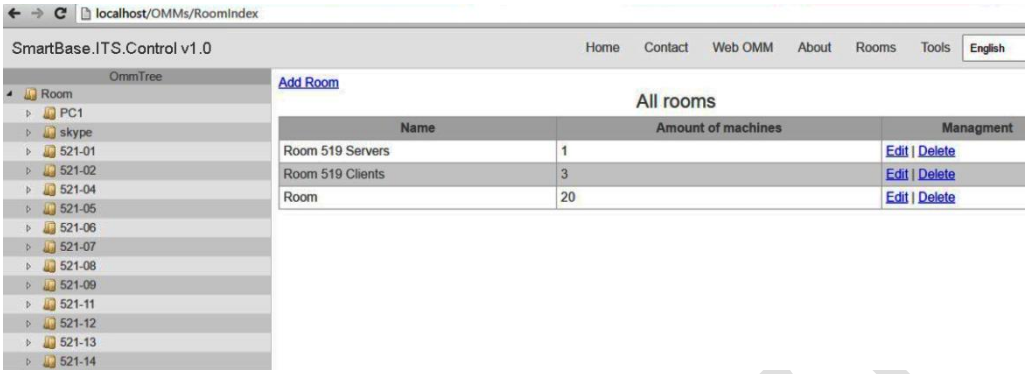


Рис. 11.37. Форма редактирования расположения

Для создания новой группы расположения следует нажать кнопку «Add Room» и ввести название группы (рис. 11.38).

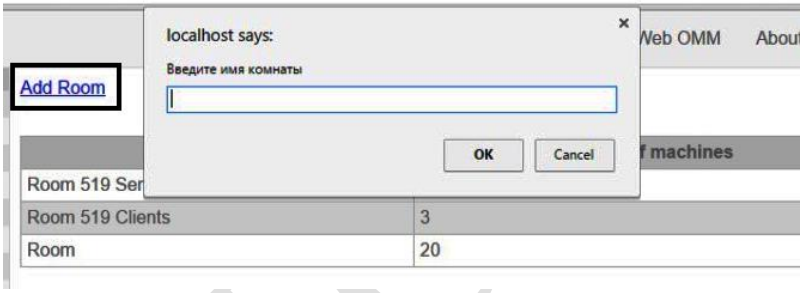


Рис. 11.38. Создание новой группы расположения

Форма отражения ОМУ. При выборе объекта в дереве ОМУ (область 1 на рис. 11.39) система отображает список параметров объекта мониторинга (область 2 на рис. 11.39). В форме (область 3 на рис. 11.39) можно задать или изменить родительский объект.

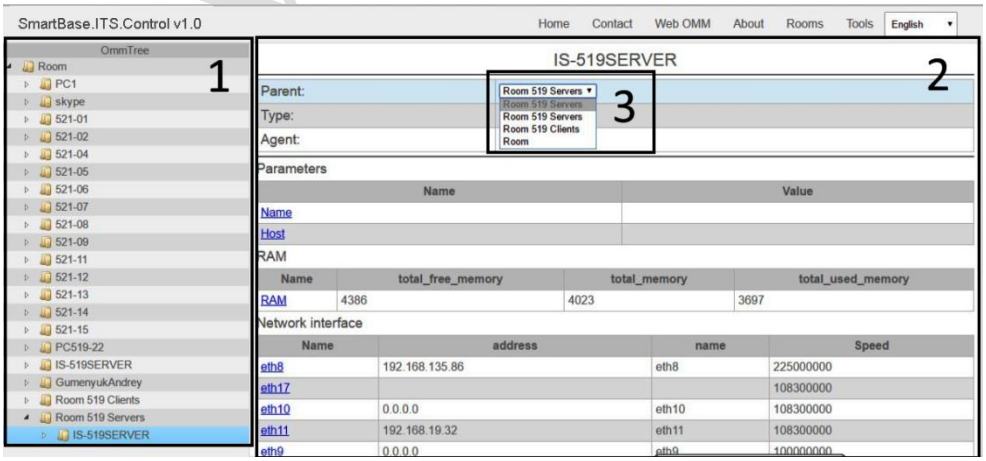


Рис. 11.39. Форма отображения ОМУ

На форме параметров ОМУ (область 1 на рис. 11.40) администратор может видеть список значений параметров ОМУ. В области 2 на рис. 11.40 отображается данные, поступившие от дочерних объектов мониторинга. В сводной таблице отображается значения не всех параметров, а только тех, которые отмечены администратором как важные.

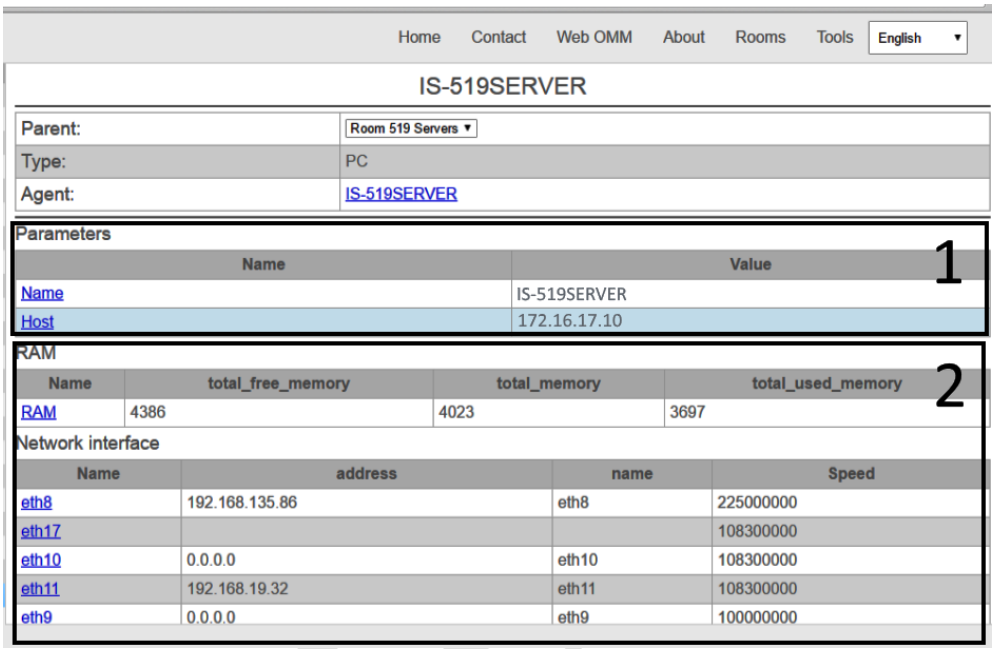


Рис. 11.40. Параметры ОМУ

Для каждого ОМУ, независимо от уровня его вложенности в иерархическом дереве ОМУ, отображается ссылка на связанного с ним агента (рис. 11.41).

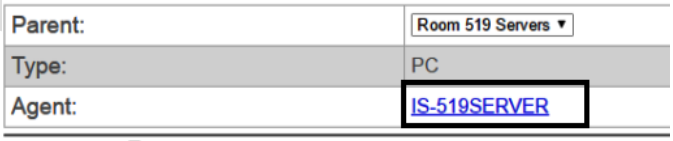


Рис. 11.41. Ссылка на агент

Форма управления агентом. Посредством следующей формы (область 1 на рис. 11.42) администратор может выполнять команды на текущем агенте. Агентские приложения осуществляют программную реализацию набора команд, среди которых есть как универсальные, так и спроектированные для конкретной операционной системы. Исполняемый код может принимать аргумент в виде JSON-объекта, а также содержит механизм асинхронного выполнения как по запросу администратора, так и в автоматическом режиме в процессе функционирования СУИ.

Рис. 11.42. Форма управления агентом

После выполнения команды появляется окно с сообщением о ее успешном выполнении (рис. 11.43). В отличии от группового выполнения команд такой способ имеет расширенные поля для ввода значений параметров и позволяет увидеть результат выполнения отдельной команды.

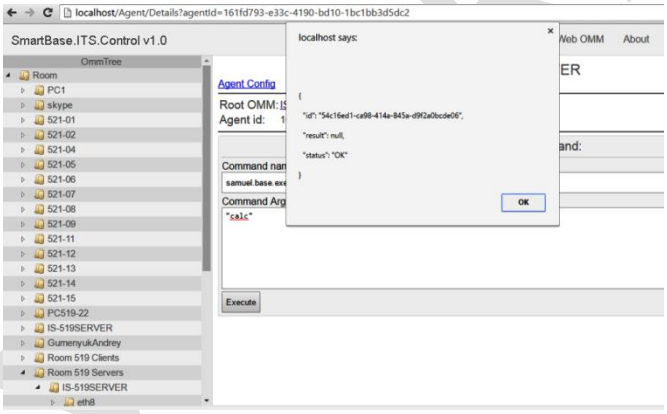


Рис. 11.43. Результат выполнения команды

При нажатии на кнопку «Agent Config» (область 2 на рис. 11.42) появляется форма настройки конфигурации агента, с помощью которой можно изменить параметры текущего агента, не имея прямого доступа к компьютеру, на котором запущен агент.

На рис. 11.44 изображена форма настройки агента. Настройки делятся на несколько частей.

Сеть. Предоставляет возможность настроить адрес текущей сети, задавать имя очереди и интервал переподключения. Также имеется возможность задавать имя хоста и IP-адрес.

Сохранение данных мониторинга. Позволяет выбирать место сохранения промежуточных данных мониторинга агентом: в оперативной памяти, во встроенной базе данных (sqlite), в удаленной базе данных (postgresql). Агент буфери-

зует данные мониторинга для того, чтобы не отправлять данные на сервер слишком часто или в случае невозможности отправки данных на сервер из-за неполадок сети.

Настройка периода мониторинга. С помощью этой строки можно настраивать время и период мониторинга текущей системы и отдельных ресурсов.

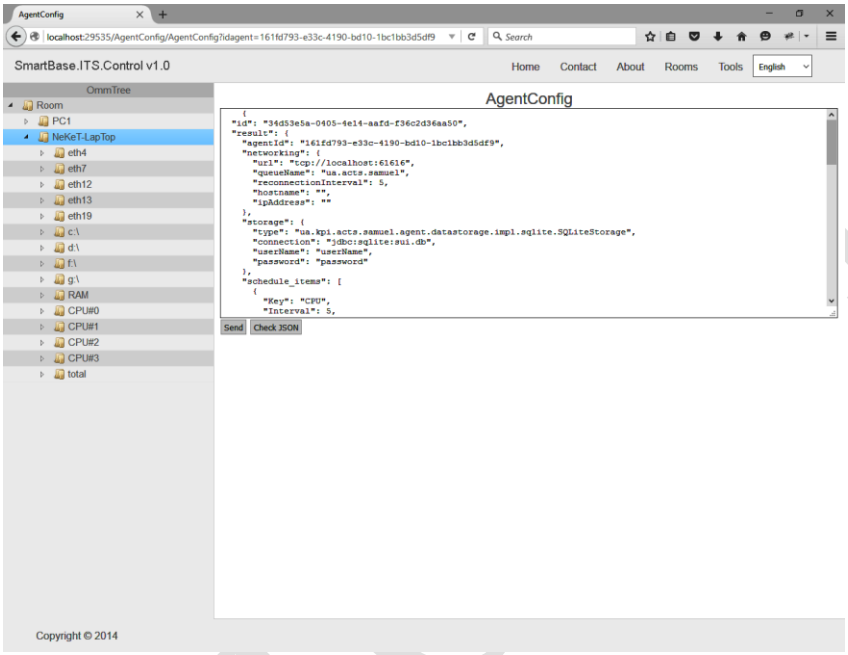


Рис. 11.44. Форма редактирования конфигурации агента

Подробная информация о значениях параметров ОМУ. При выборе конкретного ОМУ появляется список параметров и их значений. В качестве примера на рис. 11.45 отражены параметры ядра центрального процессора.

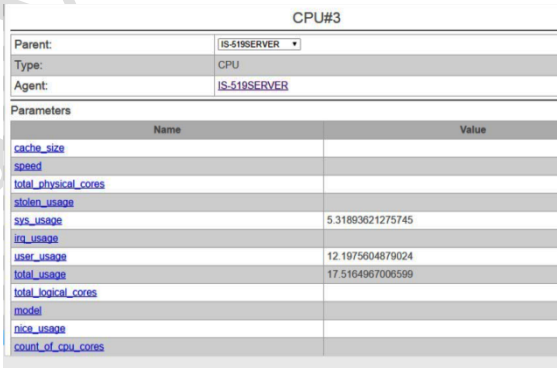


Рис. 11.45. Детализация параметров CPU

На рис. 11.46 приведена форма отображения детальной информации о рабочей станции. Для удобства восприятия информации пользователем отображаются только основные данные о процессоре.

eth8	192.168.135.86	eth8	225000000
eth17			108300000
eth10	0.0.0.0	eth10	108300000
eth11	192.168.19.32	eth11	108300000
eth9	0.0.0.0	eth9	100000000

File system

Name	free_memory	used_memory	total_memory
D:\	92.5518455505371	18.1327209472656	
c:\	75	41	116
d:\	31	0	31
C:\	42.0151977539063	70.3305015563965	

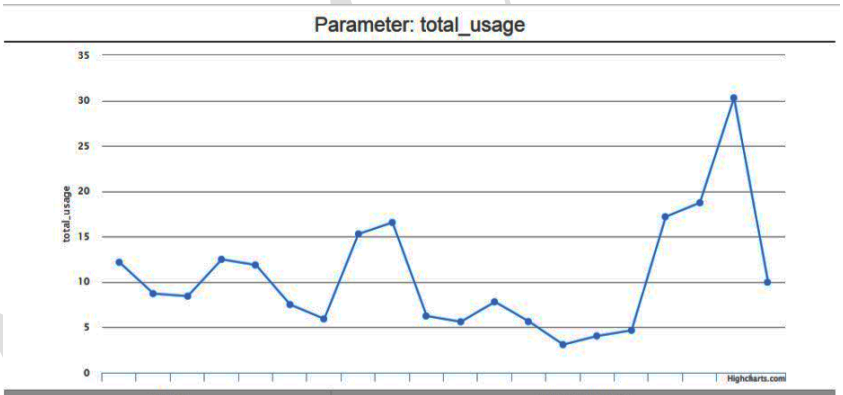
CPU

Name	sys_usage	user_usage	total_usage
CPU#3	5.31893621275745	12.1975604879024	17.5164967006599
total	5.13062950982832	7.23563075391889	12.3662602637472
CPU#2	4.68	4.06	8.74
CPU#1	4.06	8.44	12.5
CPU#0	6.55868826234753	4.37912417516497	10.9378124375125

Processes

Рис. 11.46. Форма детальной информации о рабочей станции

Для принятия актуальных решений пользователем системы реализован механизм визуализации данных мониторинга, которые доступны не только в виде данных, но и в виде графиков. При выборе отдельного параметра можно перейти на форму истории изменения его значений, которая отображается в графическом или табличном виде (рис. 11.47). При переходе на текущее представление при включенном агенте данные и график изменяются в реальном времени. Перевод агента в режим мониторинга в реальном времени осуществляется посредством команды «BeginRealTime». При этом отправка данных проводится через заданный в конфигурации промежуток времени, который отличается от штатного режима работы агента.



Value	Date Collected
12.18	11/18/2015 5:37:51 PM +02:00
8.7234893957583	11/18/2015 5:37:53 PM +02:00
8.44	11/18/2015 5:37:56 PM +02:00

Рис. 11.47. График изменения значений параметров

На странице с данными мониторинга реализован механизм постраничного отображения, с помощью которого пользователь может получать данные порциями. При этом уменьшается нагрузка на серверную и клиентскую части ПО. Пользователь может выбирать данные, которые он будет просматривать (рис. 11.48).

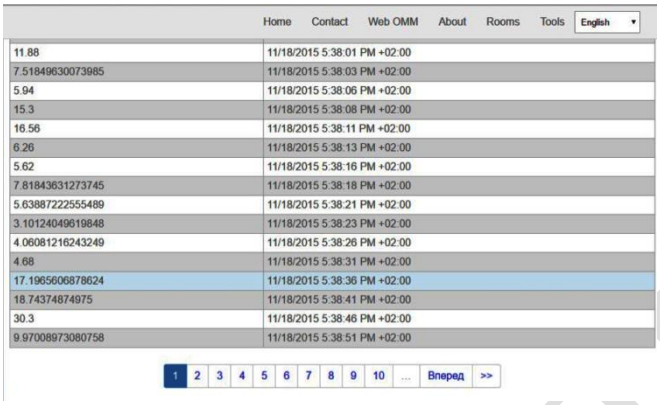


Рис. 11.48. Представление значений параметров в виде таблицы

Дерево ОМУ. В связи с большим количеством объектов мониторинга дерево ОМУ строится динамично и дочерние элементы загружаются с сервера в момент развертывания узла. Дерево ОМУ позволяет осуществлять быструю навигацию между объектами мониторинга (рис. 11.49).

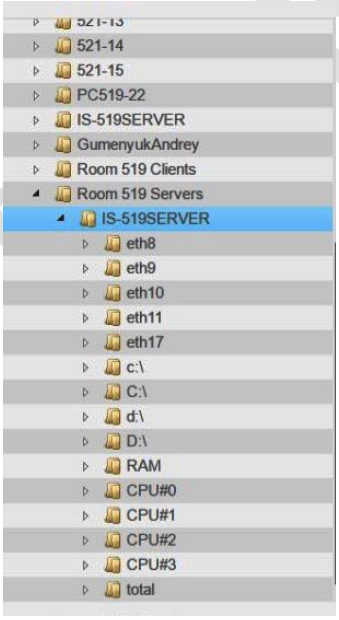


Рис. 11.49. Дерево ОМУ

Управление процессами на рабочих станциях. Для перехода на форму управления процессами нужно нажать на кнопку «Processes» в окне просмотра параметров ОМУ текущего агента (рис. 11.50).

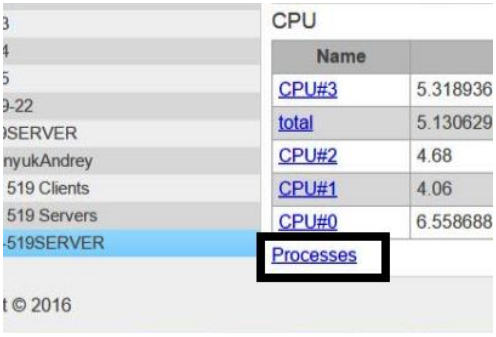
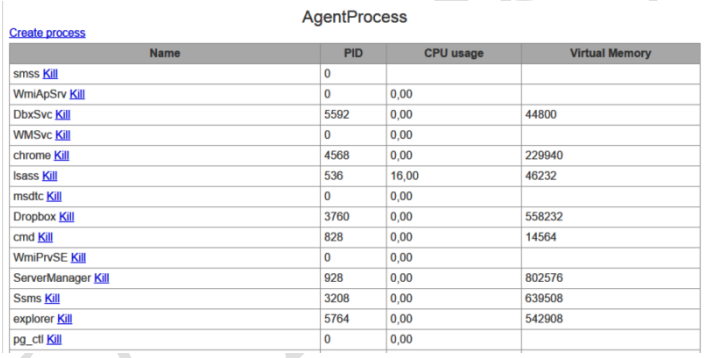


Рис. 11.50. Переход к списку процессов

Появится таблица с перечнем всех запущенных на данной машине процессов (рис. 11.51). В таблице отображается имя образа процесса, его идентификатор, количество задействованного ресурса процессора и использованного объема ОЗУ. Эта информация доступна не для всех процессов, поскольку некоторые из процессов являются критическими, системными или процессами уровня ядра операционной системы.



AgentProcess

Create process

Name	PID	CPU usage	Virtual Memory
smss Kill	0		
WmiApSrv Kill	0	0,00	
DbxSvc Kill	5592	0,00	44800
WMSvc Kill	0	0,00	
chrome Kill	4568	0,00	229940
lsass Kill	536	16,00	46232
msdtc Kill	0	0,00	
Dropbox Kill	3760	0,00	558232
cmd Kill	828	0,00	14564
WmiPrvSE Kill	0	0,00	
ServerManager Kill	928	0,00	802576
Ssms Kill	3208	0,00	639508
explorer Kill	5764	0,00	542908
pg_ctl Kill	0	0,00	

Рис. 11.51. Список запущенных процессов

Рядом с каждым процессом имеется кнопка «Kill», при нажатии на которую агент, при наличии соответствующих прав, попытается завершить выбранный процесс.

При нажатии на строку таблицы в любом месте, кроме кнопки «Kill», непосредственно в таблице отобразится развернутая информация (рис. 11.52) для данного процесса, содержащая следующие поля:

- Id – идентификатор текущего процесса;
- Used percent – процент использования процессора;
- Owner user/group name – информация о пользователе, которому принадлежит процесс;
- информация о текущем рабочем каталоге;
- имя исполнимого модуля процесса;
- статистика операций ввода/вывода по данному процессу;
- информация о виртуальной памяти процесса: количество ошибок, при страничной адресации, т. е. восстановление оперативной памяти из файла подкачки.

Этот параметр показывает, насколько активно используется файл подкачки; количество зарезервированной памяти; количество разделенной памяти, т. е. памяти, которую процесс разделяет с другими процессами; объем виртуальной памяти процесса;

- Nice value – приоритет процесса в Unix системах;
- идентификатор родительского процесса, создавшего данный процесс;
- информация о планировщике ядра, по сути являющаяся эффективным приоритетом процесса, определенном на основе значений различных параметров;
- Process state – JSON-объект, который более подробно характеризует процесс;
- количество запущенных потоков в контексте данного процесса;
- идентификатор текущей консоли для процессов, являющихся консольными.

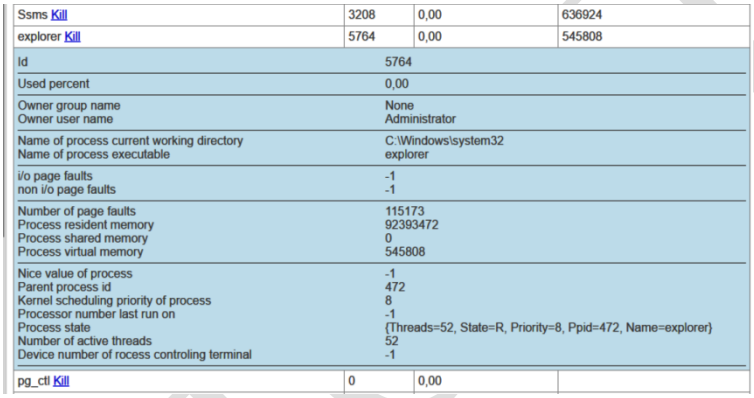


Рис. 11.52. развернутая информация по процесса

При нажатии на кнопку «Create process» (рис. 11.53) появится форма создания процесса.

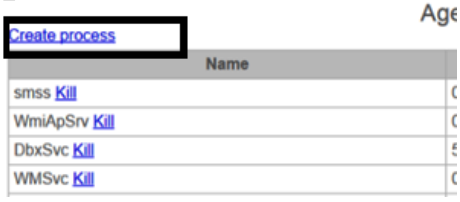


Рис. 11.53. Кнопка создания нового процесса

В форме на рис. 11.54 администратор водит название процесса и путь к рабочему каталогу. Аргументы командной строки заносятся в отдельное поле. Администратор имеет возможность задавать переменные окружения и присваивать им нужны значения.

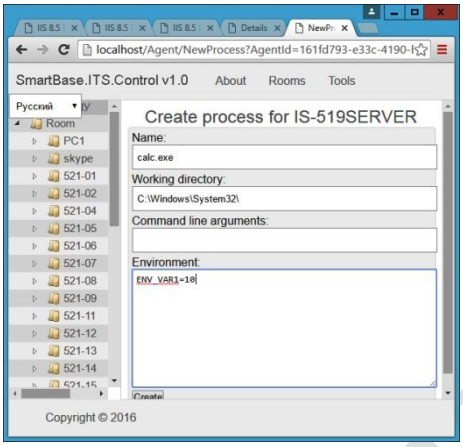


Рис. 11.54. Создание процесса

Многоязычный интерфейс. В разработанной системе реализован многоязычный пользовательский интерфейс. Для переключения языка используется выпадающий список в правом верхнем углу на всех формах интерфейса администратора (рис. 11.55).

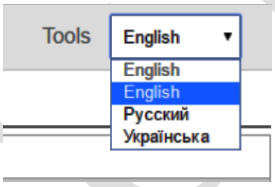


Рис. 11.55. Переключатель языка интерфейса

Мониторинг веб-ресурсов. СУИ может осуществлять мониторинг веб-ресурсов. Для этого администратор задает перечень сайтов, работоспособность которых необходимо отслеживать. После чего СУИ опрашивает заданный ресурс с определенным интервалом и в случае его неработоспособности уведомляет администраторов этих ресурсов.

Форма создания задачи мониторинга веб-ресурса приведена на рис. 11.56.

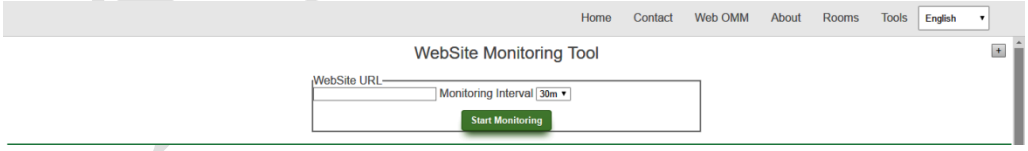


Рис. 11.56. Создание новой задачи мониторинга веб-ресурса

Администратор вводит адрес сайта и выбирает интервал опроса. Это создает в СУИ специальную сущность – Web-ОМУ и инициирует специальный планировщик, который опрашивает данный ресурс с указанным интервалом.

СУИ осуществляет мониторинг, проверяет HTTP статус, который возвращает запрошенная страница, сохраняет в базе данных время отклика удаленного узла и тип сервера, который обрабатывал запрос (рис. 11.57).

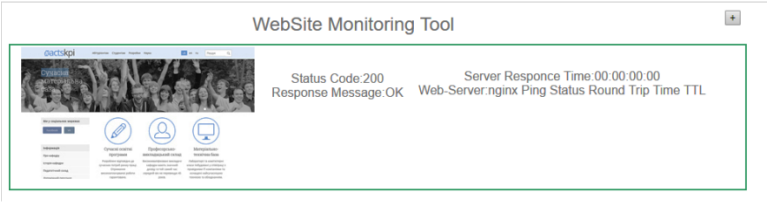


Рис. 11.57. Отображение информации о веб-сайте

В случае добавления нескольких сайтов для мониторинга (символ плюс в правом верхнем углу на рис. 11.57) будет отображаться информация о всех добавленных сайтах. Для каждого из них будет отображать последний снимок экрана, время отклика и статусный код (рис. 11.58).

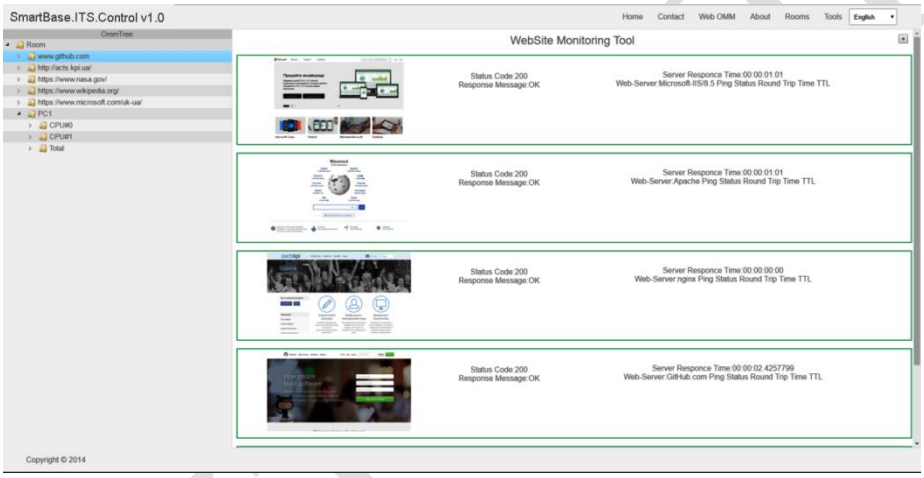


Рис. 11.58. Общий вид окна мониторинга веб-ресурсов, встроенного в главный интерфейс программы

Редактирования типов данных. В СУИ реализована гибкая мета-модель, в которой администратор системы может создавать новые типы ОМУ. Как видно с левой стороны на рис. 11.59, типы ОМУ объединены в иерархию наследования, которая отображается в виде дерева. Вершиной дерева является тип «Object» как базовый тип для всех ОМУ. От него наследуются объекты-контейнеры и место их расположения (их конкретная реализация) – здание, этаж, комната. Также созданы объекты для компьютеров и их компонентов. В случае добавления нового типа ОМУ отпадает необходимость изменять программный код, достаточно воспользоваться интерфейсом программы администратора и создать новый тип. Интерфейс позволяет не только создавать новые типы, но и редактировать существующие, в том числе изменять и добавлять новые атрибуты к типам ОМУ, которые являются аналогом колонок в базе данных, причем объектные типы поддерживают наследование.

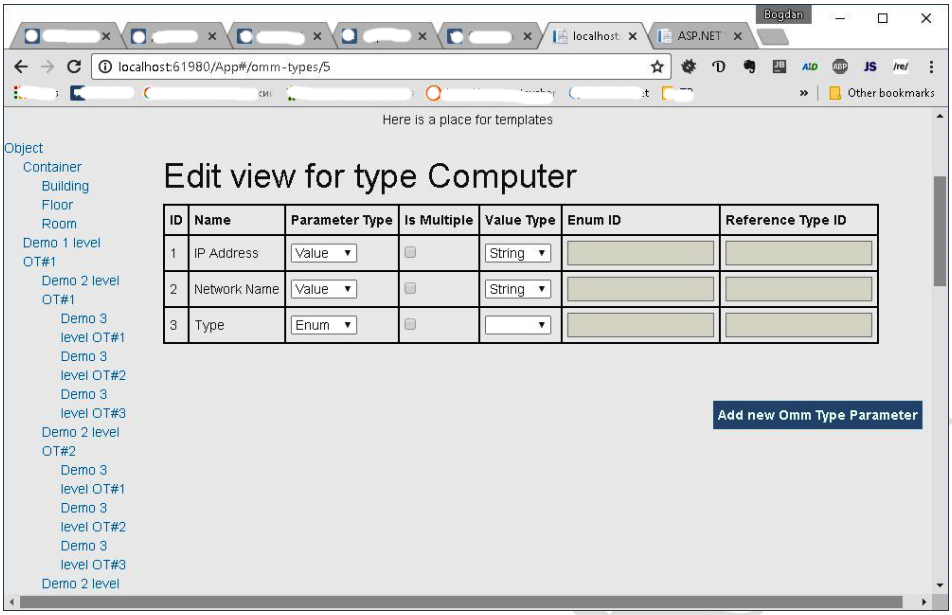


Рис. 11.59. Редактирование типов ОМУ

Первой колонкой редактора атрибутов является имя атрибута, далее следует тип атрибута. Возможны следующие значения:

- Value – простой атрибут-значение. ОМУ с таким значением параметра может хранить текущее значение заданных параметров. После выбора такого типа параметра администратор может в 4-й колонке задать конкретный тип значения: текст, целое число, двоичное число, бинарные данные и др.;
- Enum – перечисляемые типы данных. Дискретные значения (да/нет, вкл./выкл., хорошо/удовлетворительно/плохо). Администратор в отдельной форме может создавать свои перечисляемые типы. Количество значений в перечислении не ограничено;
- ссылка на другой. Например, бизнес-процесс может содержать ссылки на серверы приложений;
- ссылка на тип – мета-информация, применяется при создании конфигурационных типов, которые описывают процессы внутри системы. Может использоваться для конфигурирования процессов мониторинга и автоматизации создания объектов;
- ссылка на мета-атрибут – позволяет в ОМУ иметь ссылки на шаблонные параметры, описания атрибутов. Дает возможность задавать конфигурацию мониторинга и указывать, в какое «поле» объекта добавлять новые данные.

С целью облегчения модульного тестирования и тестирования пользовательского интерфейса программно сгенерировано дерево типов с наследованием и большим количеством параметров (рис. 11.60).

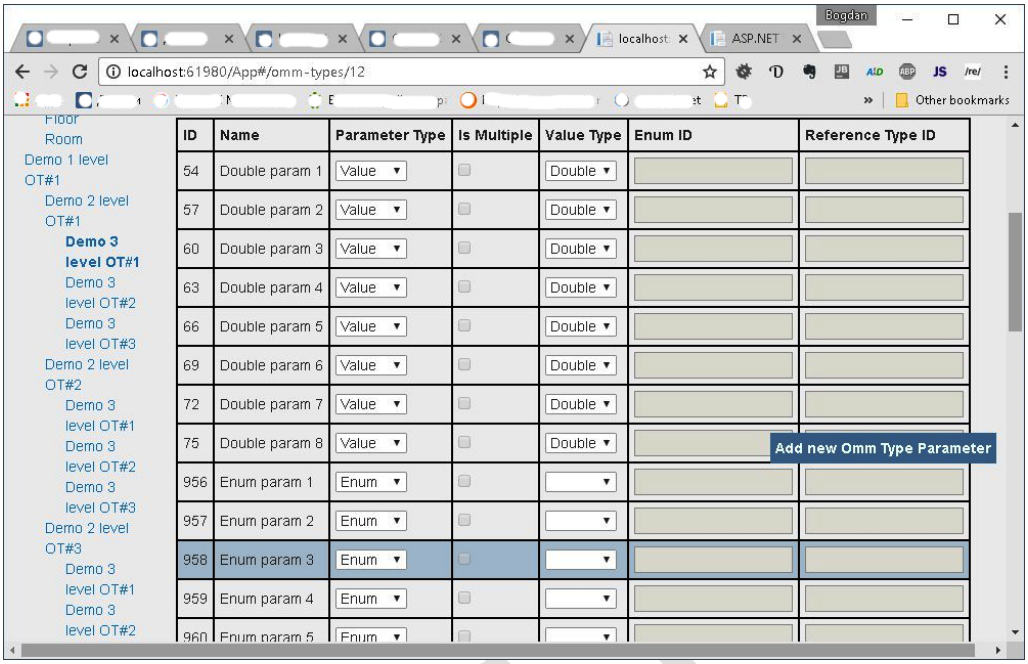


Рис. 11.60. Тестовый тип с заданным большим количеством параметров

Агентское приложение. Агентское приложение является консольным приложением, которое можно запустить в консольном окне, как это изображено на рис. 11.61.

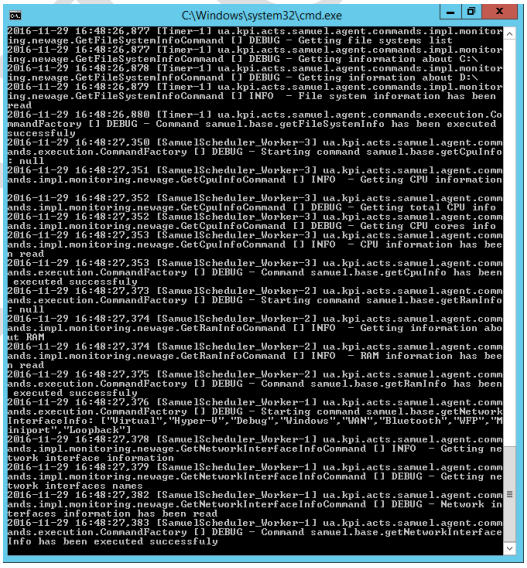


Рис. 11.61. Консольное окно агентского приложения

В этом случае можно видеть логи приложения в реальном времени, что удобно при разработке и отладке, а также активно используется администраторами при настройке агентов. Кроме того агент может быть запущен как сервис Windows или демон Unix. При этом логи будут записываться в специальный каталог с файлами журналов.

ВЫВОДЫ К ГЛАВЕ 11

1. Определены принципы построения систем управления ИТ-инфраструктурой на основе разработанной ИТ УИТИ для построения многоуровневых территориально-распределенных организационно-технических систем, процедуры принятия решений которых построены на предложенных в работе математических моделях и методах.

2. Определены технологические аспекты применения разработанной информационной технологии управления ИТ-инфраструктурой для построения процессов мониторинга, анализа, управления, оптимизации и планирования в СУИ.

3. Разработана архитектура системы управления ИТ-инфраструктурой SmartBase.ITS.Control, построенная на основе клиент-серверной и агентской технологий, с интеграцией методов активного и пассивного мониторинга и консолидацией централизованного и децентрализованного управления, что позволяет обеспечить высокую надежность, оперативность и гибкость управления.

4. Разработаны структура, определены принципы функционирования и разработаны механизмы взаимодействия программного сервера, агентского модуля и клиента СУИ, использующие предложенные методы анализа состояния, сведения метрик, моделей, методов и инструментальных средств управления ресурсами и нагрузкой компонентов СУИ, что позволяет легко подключать новые типы оборудования, применять новые технологии и легко реагировать на изменения в ИТ-инфраструктуре.

5. Предложенные модели и методы практически реализованы в программном комплексе СУИ SmartBase.ITS.Control, разработанном в Национальном техническом университете Украины «Киевский политехнический институт имени Игоря Сикорского». СУИ прошла апробацию в системах специального назначения и корпоративных ИТ-инфраструктурах. СУИ SmartBase.ITS.Control может быть использована в различных СУ корпоративными ИТ-инфраструктурами для повышения надежности и эффективности использования ресурсов ИТ-инфраструктуры, а также повышения результативности применения информационных технологий. Внедрение СУИ доказало реализуемость предложенных в работе методов и эффективность полученных решений.

АУТСОРС ИТ

СПИСОК УСЛОВНЫХ СОКРАЩЕНИЙ

АРМ	Автоматизированное рабочее место
БД	База данных
ВМ	Виртуальная машина
ВМП	Подалгоритм выполнения модели для импакт-анализа
ВТ	Вычислительная техника
ВЦКП	Вычислительный центр коллективного пользования
ГА	Генетический алгоритм
ИВС	Информационно-вычислительная сеть
ИКТ	Информационно-коммуникационные технологии
ИПКФ	Интегральный показатель качества функционирования
ИТ	Информационные технологии
ИТ УИТИ	Информационная технология управления ИТ-инфраструктурой
ИУС	Информационно-управляющая система
КОФ	Качественная оценка функционирования
ЛОУ	Логический объект мониторинга и управления
МАУОП	Мониторинг, анализ, управление, оптимизация, планирование
НС	Нейронная сеть
ОЗУ	Оперативное запоминающее устройство
ОМУ	Объект мониторинга и управления
ОТС	Операторы телекоммуникационных сервисов
ПО	Программное обеспечение
ПК	Персональный компьютер (рабочая станция)
ПИА	Подсистема импакт-анализа
ПУИП	Подсистема управления информационными потоками
ПФМ	Подалгоритм формирования модели для импакт-анализа
РОУ	Реальный объект мониторинга и управления
СМО	Система массового обслуживания
СНЛВ	Система нечеткого логического вывода
СОА	Сервис-ориентированная архитектура
СППР	Система поддержки принятия решений
СРМ	Сервисно-ресурсная модель
СУ	Система управления
СУБД	Система управления БД
СУИ	Система управления ИТ-инфраструктурой
ТКС	Телекоммуникационная сеть
ТПС	Технологическая подсистема
УИТИ	Управление ИТ-инфраструктурой
УК	Узлы коммутации
УНПР	Устройство настройки параметров регулятора

АУЭС РК

УП	Управляющая подсистема
ФГ	Функциональная группа
ФПС	Функциональная подсистема
ЦОД	Центр обработки данных
ЦФ	Целевая функция
ApM	Application Performance Management – управление производительностью приложений
API	Application Programming Interface – программный интерфейс приложения
BpM	Business Performance Management – управления производительностью выполнения бизнеса
BPM	Business Process Management – концепция управления организацией
BSM	Business Service Management – управление бизнес-сервисами
CMS	Configuration Management System – система управления конфигурациями
CRM	Customer Relationship Management – управление взаимоотношениями с клиентами
DALL	Dynamic Auto Link Library – динамически связываемые библиотеки
ERP	Enterprise Resource Planning – планирование ресурсов предприятия
eTOM	enhanced Telecom Operations Map – расширенная карта процессов телекоммуникационных компаний
IaaS	Infrastructure-as-a-Service – инфраструктура как сервис
ITIL	Information Technology Infrastructure Library – библиотека инфраструктуры информационных технологий
ITMS2	IT Management Software 2.0 – парадигма ПО управления ИТ
ITSM	IT Service Management – управление ИТ-услугами
itSMF	IT Service Management Forum – форум по ИТ сервис-менеджменту
KPI	Key Performance Indicators – ключевые показатели эффективности
KQI	Key Quality Indicators – ключевые показатели качества
OSS	Operation Support System – системы операционной деятельности
PaaS	Platform as a Service – платформа как услуга
SAN	Storage Area Network – сеть хранения данных
SaaS	Software as a Service – программное обеспечение как услуга
SLA	Service Level Agreement – соглашение об уровне обслуживания
SLM	Service-level management – управление уровнем обслуживания
TMN	Telecommunication Management Network – система управления сетями операторов электросвязи
QoS	Quality of Service – качество сервисов
QoE	Quality of Experience – удовлетворенность пользователя качеством услуг

Архив НТБ

СПИСОК ЛИТЕРАТУРЫ

1. *Айзерман М.А.* Метод потенциальных функций в теории обучения машин / М.А. Айзерман, Э.М. Браверман, Л.И. Розоноэр. – М.: «Наука», 1970. – 384 с.
2. *Алиев Р.А.* Методы и алгоритмы координации в промышленных системах управления / Р.А. Алиев, М.И. Либерзон. – М.: Радио и связь, 1987. – 208 с.
3. *Александров А.* Конкретно о CMDB / А. Александров // Открытые системы. – 2007. – №6. – С. 45–51.
4. *Алешкин А.С.* Новый подход к моделированию информационно-вычислительных сетей / А.С. Алешкин, Д.О. Жуков // Приборы и системы. Управление, контроль, диагностика. – 2007. – №9. – С. 35–40.
5. *Аноприенко А.Я.* Многоуровневое моделирование сетей на базе стека протоколов TCP/IP в среде Matlab/Simulink / А.Я. Аноприенко, С.Н. Джон, А.Ж. Нури // Науч. труды ДонНТУ. Вып. 39. Сер.: Информатика, кибернетика и вычислительная техника (ИКВТ-2002): – Донецк: ДонНТУ, 2002. – С. 271–279.
6. *Байдачный С.С.* ASP.NET 2.0: секреты создания Web-приложений / С.С. Байдачный, Д.А. Маленко. – М.: СОЛОН-ПРЕСС, 2007. – 736 с.
7. *Бесекерский В.А.* Теория систем автоматического регулирования / В.А. Бесекерский, Е.П. Попов, изд. третье, испр. – М., «Наука», 1975. – 768 с.
8. *Боданюк М.Є.* Управління системами збереження даних / М.Є. Боданюк, О.К. Карнаухов, О.І. Ролік, С.Ф. Теленик // Electronics and Communications. – 2013. – №5 (76). – С. 81–90.
9. *Бойко Ю.В.* Український академічний грід: досвід створення і перші результати експлуатації / Ю.В. Бойко, М.Г. Зинов'єв, С.Я. Свістунов, О.О. Судаков // Математичні машини і системи. – 2008. – №1. – С. 67–84.
10. *Брукс П.* Метрики для управления ИТ-услугами: пер. с англ. / П. Брукс. – М.: Альпина Бизнес Букс, 2008. – 283 с.
11. *Будко П.А.* Выбор пропускных способностей каналов связи в условиях изменяющейся нагрузки // Физика волновых процессов и радиотехнические системы. – 2000. – Т.3. – №3–4. – С. 86–72.
12. *Будко П.А.* Управление в сетях связи. Математические модели и методы оптимизации/ П.А.Будко, В.В. Федоренко. – М.: Наука, 2003. – 288 с.
13. *Будкова Л.* Методическое руководство для подготовки к профессиональным экзаменам ISO 20000 Foundation и ISO 20000 Foundation Bridge» / Л. Будкова, Р. Журавлев. – М.: Клеверикс, 2010. – 123 с.
14. *Бурков В.Н.* Теория активных систем: Состояние и перспективы / В.Н. Бурков, Д.А. Новиков. – М.: Синтег, 1999. – 128 с.
15. *Буч Г.* Объектно-ориентированный анализ и проектирование (с примерами приложений на C++) / Г. Буч. – Бином, Невский Диалект, 1998. – 560 с.

16. *Буч Г.* Язык UML. Руководство пользователя: пер. с англ. / Г. Буч, Дж. Рамбо, А. Джекобсон. – М.: ДМК, 2000. – 322 с.

17. *Васильев В.И.* Распознающие системы: [справочник] / В.И. Васильев. – К.: Наукова думка, 1983. – 422 с.

18. *Вентцель Е.С.* Теория случайных процессов и ее инженерные приложения / Е.С. Вентцель, Л.А. Овчаров. – М.: Наука. Гл. ред. физ.-мат. лит., 1991. – 384 с.

19. *Волохов В.М.* Розробка програмного забезпечення обліку суддів, штатних розписів, зведених інформаційних і статистичних форм звітностей кадрових служб судової системи Верховного Суду України – системи «Кадри Верховного суду України» / В.М. Волохов; М.Г. Рябчун – К.: ТОВ «МККУ-мережі», 2001. – 60 с.

20. *Вороновский Г.К.* Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Харьков: ОСНОВА, 1997. – 107 с.

21. *Вьюгин В.* Математические основы теории машинного обучения и прогнозирования / В. Вьюгин. – М.: МЦМНО, 2013. – 390 с.

22. *Гаврилова Т.А.* Базы знаний интеллектуальных систем / Т.А. Гаврилова, В. Ф. Хорошевский. – СПб: Питер, 2000. – 384 с.

23. *Гамма Э.* Приемы объектно-ориентированного проектирования. Паттерны проектирования: пер. с англ. / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб: Питер, 2004. – 366 с.

24. *Глибовець А.М.* Програмні агенти / А.М. Глибовець, М.М. Глибовець, С.С. Гороховський, М.О. Сидоренко. – К.: НаУКМА, 2013. – 204 с.

25. *Глибовець М.М.* Веб сервіси оброблення документів / М.М. Глибовець, А.А. Жигмановський, Р.І. Заболотний, П.О. Захоженко. – К.: [НаУКМА], 2012. – 212 с.

26. *Глибовець М.М.* Програмні засоби створення і супроводу розподіленого навчального середовища / М.М. Глибовець, І.В. Сергієнко, С.С. Гороховський, А.М. Глибовець. – К.: [НаУКМА], 2012. – 709 с.

27. *Глибовець М.М.* Розробка системи управління вищого навчального закладу / М.М. Глибовець, С.А. Іващенко, О.О. Крусь // Наук. зап. Нац. унів. «Києво-Могилянська академія». Комп'ютерні науки. – 2005. – Т. 36. – С. 6–13.

28. *Голубцов В.* Сервисно-ресурсная модель. От теории к практике / В. Голубцов, М. Федоренко // Альманах ITSM 2012. – itSMF России. – 2012.

29. *Гребешков А.Ю.* Стандарты и технологии управления сетями связи / А.Ю. Гребешков. – М.: Эко-Тренз, 2003. – 228 с.

30. *Дикин И.И.* Итеративное решение задач линейного и квадратичного программирования / И.И. Дикин // Доклады АН СССР. – 1967. – Т.174. – С. 747–748.

31. *Дубова Н.* Управление услугами для бизнеса / Н. Дубова // Открытые системы. – 2005. – № 1.

32. *Душин В.К.* Теоретические основы информационных процессов и систем: Учебник / В.К. Душин. – «Дашков и К°», 2003. – 348 с.

33. *Дымарский Я.С.* Управление сетями связи: принципы, протоколы, прикладные задачи. Серия изданий «Связь и бизнес»/ Я.С. Дымарский, Н.П. Крутякова, Г.Г. Яновский. – М.: ИТЦ «Мобильные коммуникации», 2003. – 384 с.

34. *Еремин И.И.* Теория линейной оптимизации /И.И. Еремин. – Екатеринбург, 1999. – 312 с.

35. *Ершов С.В.* Модель интеллектуальных агентов, основанная на нечеткой логике высшего типа / С.В. Ершов // Компьютерная математика, 2012. – № 1. – С. 10–16.

36. *Ершов С.В.* Принципы построения нечетких мультиагентных систем в распределенной среде / С.В. Ершов // Компьютерная математика, 2009. – № 2. – С. 54–61.

37. *Зайченко Е.Ю.* Исследование эффективности применения адаптивного генетического алгоритма для синтеза структуры компьютерной сети MPLS / Е.Ю. Зайченко, А.С. Аникиев // Вісник ЧДТУ. – 2009. – №1. – С. 3–9.

38. *Зайченко Е.Ю.* Средства повышения эффективности применения генетического алгоритма для синтеза структуры компьютерной сети MPLS / Е.Ю. Зайченко, А.С. Аникиев // Вісник ЧДТУ. – 2008. – №3. – С. 100–106.

39. *Зайченко Ю.П.* Исследование операций/ Ю.П. Зайченко. – 6 изд., перераб. и доп. – К.: Издательский дом «Слово», 2003. – 688 с.

40. *Захаров Г.П.* Математическое моделирование системы управления сетью связи / Г.П. Захаров, Н.П. Крутякова, Н.И. Горбенко // Вопросы кибернетики. Протоколы и методы коммутации в вычислительных сетях. – М.: НСК АН СССР, 1986. – С. 88–100.

41. *Згуровский М.З.* Принятие решений в сетевых системах с ограниченными ресурсами/ М.З. Згуровский, А.А. Павлов. – К.: Наукова думка, 2010. – 573 с.

42. *Згуровский М.З.* Системный анализ: проблемы, методология, приложения / М.З. Згуровский, Н.Д. Панкратова. – К.: Наукова думка, 2011. – 728 с.

43. *Згуровский М.З.* Стратегія технологічного передбачення в інноваційній діяльності / М.З. Згуровський, Н.Д. Панкратова // Науково-технічна інформація. – 2006. – 2 (28). – С. 3–10.

44. *Зителло Т.* HP Open View – настольная книга системного администратора: Network Node Manager, Customer Views, Service Information Portal, HP OpenView Operations / Т. Зителло, Д. Вильямс, П. Вебер. – Эком, Интернет-университет информационных технологий, 2008. – 743 с.

45. Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации: ГОСТ Р ИСО/МЭК 8824-1-2001. – М.: ИПК Издательство стандартов, 2001. – 104 с.

46. Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 2. Спецификация информационного объекта: ГОСТ Р ИСО/МЭК 8824-2-2001. – М.: ИПК Издательство стандартов, 2001. – 27 с.

47. Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 3. Спецификация ограничения. ГОСТ Р ИСО/МЭК 8824-3-2002. – М.: ИПК Издательство стандартов, 2002. – 9 с.

48. Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 4. Параметризация спецификации АСН.1: ГОСТ Р ИСО/МЭК 8824-4-2003. – М.: ИПК Издательство стандартов, 2003. – 15 с.

49. Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 4. Основы административного управления: ГОСТ Р ИСО/МЭК 7498-4-99. – М.: ИПК Издательство стандартов, 1999. – 12 с.

50. *Катышев С.* Об одной концепции управления распределенным ресурсами / С. Катышев // Открытые системы. – 1998. – №3.

51. *Клейнрок Л.* Вычислительные системы с очередями / Л. Клейнрок. – М.: Мир, 1979. – 595 с.

52. *Клейнрок Л.* Теория массового обслуживания/ Л. Клейнрок. – М.: Машиностроение, 1979. – 432 с.

53. *Коберн А.* Современные методы описания функциональных требований к системам/ А. Коберн. – М.: Лори, 2002. – 366 с.

54. *Коваленко В.Н.* Метод опережающего планирования для грид / В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин, Э.З. Любимский // Препринт №112, М.: ИПМ РАН. – 2005. – С. 1–33.

55. *Коршевнюк Л.А.* Решение задачи распределения инвестиций на основе нечеткого логического вывода / Л.А. Коршевнюк, П.И. Бидюк // Системні дослідження та інформаційні технології. – 2003. – № 2. – С. 34–42.

56. *Коршевнюк Л.А*. Решение задачи распределения инвестиций между альтернативными проектами / Л.А. Коршевнюк, П.И. Бидюк // Збірник наукових праць Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова. – Вип. 17. – Київ, 2002. – С. 26–33.

57. *Коршевнюк Л.О.* Система нечіткого логічного виводу із зваженою істинністю / Л.О. Коршевнюк, М.Ю. Мінін // Единое информационное пространство 2004: Сб. докладов II-й Межд. науч.-практ. конф. – Днепропетровск: ИПК ИнКомЦентра УГХТУ, 2004. – С. 114–117.

58. *Костин А.А.* Модель системы интегрированного управления телекоммуникационными сетями и услугами / А.А. Костин // Электросвязь. – 2002. – №10. – С. 22–26.

59. *Кривий С.Л.* Дискретна математика / С.Л. Кривий. – Чернівці-Київ: Букрек, 2017. – 567 с.

60. *Кривий С.Л.* Дискретна математика: вибрані питання / С.Л. Кривий. – К.: Видавничий дім «Києво-Могилянська академія», 2007. – 570 с.

61. *Кривий С.Л.* Лінійні діофантові обмеження та їх застосування / С.Л. Кривий. – Київ-Чернівці, 2015. – 224 с.

62. *Кровчик* Э. .Net. Сетевое программирование для профессионалов / Э. Кровчик, В. Кумар, Н. Лагари. – М: Лори, 2005. – 400 с.

63. *Круглов В.В.* Искусственные нейронные сети. Теория и практика. 2-е изд. / В.В. Круглов, В.В. Борисов. М.: ГЛ Телеком, 2002. – 382 с.

64. *Курейчик В.М.* Генетические алгоритмы и их применение / В.М. Курейчик. – 2-е изд., дополненное. Таганрог: Изд-во ТРТУ, 2002. – 242 с.

65. *Кученко Ю.* ЦОД как объект системной и структурной оптимизации / Ю. Кученко // Компьютерное обозрение. – 2009. – №15.

66. *Лазарев В.Г.* Динамическое управление потоками информации в сетях связи / В.Г. Лазарев, Ю.В. Лазарев. – М.: Радио и связь. – 1983.

67. *Лепетюк А.Л.* Наступает новая эпоха ИТ / А.Л. Лепетюк // Информационные технологии для ИТ-менеджмента. – 2011. – №5. – С. 22–27.

68. *Лэсдон Л.С.* Оптимизация больших систем / Л.С. Лэсдон. – М.: Наука, 1975. – 432 с.

АУТСКП

69. *Месарович М.* Теория иерархических многоуровневых систем: пер. с англ. / М. Месарович, Д. Мако, И. Такахара; пер. с англ. под ред. И.Ф. Шахнова. – М.: Мир, 1973. – 344 с.

70. *Методы* исследования свойств высокопроизводительных инфраструктур. Обзор/ Ю.В. Бойко, Н.Н. Глибовец, С.В. Ершов, С.Л. Крывый, С.Д. Погорелый, А.И. Ролик, С.Ф. Теленик, А.И. Куляс, Ю.В. Крак, М.В. Ясочка // Управляющие системы и машины. – 2015. – № 1. – С. 3–13.

71. *Методи* та новітні підходи до проектування, управління і застосування високопродуктивних ІТ-інфраструктур: монографія / Ю.В. Бойко, В.М. Волохов, М.М. Глибовець та ін. за ред. проф. А. В. Анісімова. – К.: ВПЦ «Київський університет», 2016. – 447 с.

72. Методы классической и современной теории автоматического управления: Учебник в 5-и тт.; 2-е изд., перераб. и доп. Т.5: Методы современной теории автоматического управления / Под ред. К.А. Пупкова, Н.Д. Егупова. – М.: Издательство МГТУ им. Н.Э. Баумана, 2004. – 784 с.

73. Методы робастного, нейро-нечеткого и адаптивного управления: Учебник / Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 744 с.

74. *Михалевич В.С.* Последовательные алгоритмы оптимизации и их применение/ В.С. Михалевич // Кибернетика. – 1965. – №1. – С. 45–55; №2. – С. 85–88.

75. Моделювання компонентів базового програмного забезпечення розподілених середовищ / Ю.В. Бойко, М.М. Глибовець, С.Л. Кривий, С.Д. Погорілий, О.І. Ролік, С.Ф. Теленик, С.В. Єршов // Наукові праці: Науково-методичний журнал. – Вип. 238. Т. 250. Комп'ютерні технології. – Миколаїв: Вид-во ЧДУ ім. Петра Могили, 2014. – С. 6–13.

76. *Моисеев Н.Н.* Численные методы в теории оптимальных систем/ Н.Н. Моисеев. – М.: Наука, 1971. – 424 с.

77. *Мясников А.С.* Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов / Мясников А.С. // Наука и образование: электронное научно-техническое издание. МГТУ им. Н.Э. Баумана. – 2010. – № 1. – С. 3–5.

78. Надійність техніки. Проектна оцінка надійності складних систем з урахуванням технічного і програмного забезпечення та оперативного персоналу. Основні положення: ДСТУ 3524-97. – К.: Держстандарт України, 1997. – 54 с.

79. *Острейковский В.А.* Теория надежности / В.А. Острейковский. – М.: Высшая школа, 2003. – 463 с.

80. Оценка качества предоставления мультимедийных сервисов с использованием нейросетевого классификатора / А.И. Ролик, Д.А. Галушко, В.В. Барна, А.В. Томащук, М.В. Ясочка // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: Століття +, 2015. – №63. – С. 25–30.

81. *Павлов О.А.* Інформаційні технології та алгоритмізація в управлінні / О.А. Павлов, С.Ф. Теленик. – К.: Техніка, 2002. – 344 с.

82. *Палагин А.В.* Реконфигурируемые вычислительные системы/ А.В. Палагин, В.Н. Опанасенко. – К.: Просвіта. – 2006. – 295 с.

83. *Парасюк И.Н.* Мультиагентные модели на основе нечеткой логики высшего типа для высокопроизводительной среды / И.Н. Парасюк, С.В. Ершов // Проблеми програмування, 2012. – № 2-3. – С. 260–269.

84. *Парасюк И.Н.* Нечеткие модели мультиагентных систем в распределенной среде / И.Н. Парасюк, С.В. Ершов // Проблеми програмування, 2010. – №2-3. – С. 330–339.

85. *Пасечников И.И.* Методология анализа и синтеза предельно нагруженных информационных сетей. – М.: Машиностроение, 2004. – 216 с.

86. Пат. 65906 Україна, МПК G06F 11/34. Спосіб аналізу параметрів функціонування об'єктів інформаційно-телекомунікаційної системи / О.І. Ролік, П.Ф. Можаровский. – № u 2010 12774; заявл. 28.10.10; опубл. 26.12.11, Бюл. №24. – 6 с.

87. Пат. 103475 Україна, МПК G06F 11/34, G06F 11/00. Спосіб аналізу функціонування об'єктів інформаційно-телекомунікаційної системи / О.І. Ролік, П.Ф. Можаровский. – № a 2010 12773; заявл. 28.10.10; опубл. 25.10.13, Бюл. №20. – 6 с.

88. Пат. 5872911 США, МКИ G06F 11/34. Method and system of service impact analysis in a communications network/ M. Berg; MCI Communications corp. – № 581748; Заявл. 29.12.95; Опубл. 16.02.99. – 16 с.

89. Пат. №6377907 США. МПК G06F 11/34. System and method for collation UNIX performance metrics/ A. C. Waclawski. – Опубл. 23.04.2002.

90. Пат. 6708137 США, МПК G06F 11/34. System and method for providing variance analysis for network operation/ Carley S. – Опубл. 16.03.04.

91. Пат. 7092707 США, МКИ H04Q 7/20, H04B 1/38, H04B 7/00. Service impact analysis and alert handling in telecommunications systems/ R. Lau, K. Kant, R. Khare; Telcordia Technologies, Inc., – № 10/779432; Заявл. 18.08.05; Опубл. 15.08.06. – 21 с.

92. *Погорілий С.Д.* Застосування генетичних алгоритмів в комп'ютерних системах / С.Д. Погорілий, Р.В. Білоус, І.В. Білоконь. – К.: ВПЦ «Київський університет», 2014. – 320 с.

93. *Погорілий С.Д.* Методи кластерних обчислень / С.Д. Погорілий, Ю.В. Бойко, Р.І. Левченко, В.А. Мар'яновський; за ред. С.Д. Погорілого. – ВПЦ «Київський університет», 2013. – 416 с.

94. *Погорілий С.Д.* Підходи до формалізації проектування застосувань в технології GPGPU / С.Д. Погорілий, О.А. Верещинський, Д.Ю. Вітель // Системні дослідження та інформаційні технології. – 2014. – №4. – С. 45–59.

95. *Поповский В.В.* Динамическое управление ресурсами ТКС: математические модели в пространстве состояний / В.В. Поповский, А.В. Лемешко, О.Ю. Евсеева // Науково-виробничий збірник «Наукові записки Українського науково-дослідного інституту зв'язку». – 2009. – № 1(9). – С. 3–26.

96. *Порецкий С.* Моделирование алгоритма маршрутизации транспортной АТМ сети // Электросвязь. – №10. – 2000. – С.16–19.

97. Программирование на платформе Microsoft .Net Framework: пер. с англ. / Дж. Рихтер. – М.: Русская редакция, 2002. – 512 с.

98. *Просиз Дж.* Программирование для Microsoft .Net: пер. с англ. / Дж. Просиз. – М.: Издательско-торговый дом «Русская редакция», 2003. – 704 с.

99. Протоколы информационно-вычислительных сетей: Справочник/ С.А. Аничкин, В.А. Белов, А.В. Бернштейн и др., Под ред. И.А. Мизина, А.П. Кулешова. – М.: Радио и связь, 1990. – 504 с.

100. Протоколы и методы управления в сетях передачи данных / Под ред. Ф.Ф. Куо. – М.: Радио и связь, 1985. – 480 с.

101. *Растригин Л.А.* Адаптация сложных систем / Л.А. Растригин // Изв. АН ЛатвССР. – 1978. – №5 (370). – С. 38–45.

102. *Репин В.В.* Процессный подход к управлению. Моделирование бизнес-процессов / В.В. Репин, В.Г. Елиферов. – М.: РИА «Стандарты и качество», 2008. – 408 с.

103. *Ролик А.И.* Алгоритм последовательного анализа вариантов в задаче распределения виртуальных машин в центре обработки данных / А.И. Ролик, М.Е. Боданюк // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2012. – Вип. 21 (41). – С. 61–69.

104. *Ролик А.И.* Анализ качества функционирования элементов информационно-телекоммуникационных систем / А.И. Ролик, Е.В. Глушко // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: ВЕК+, 2008. – №48. – С. 113–120.

105. *Ролик А.И.* Декомпозиционно-компенсационный подход к управлению уровнем услуг в корпоративных ИТ-инфраструктурах / А.И. Ролик // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, – 2013. – № 58. – С. 78–88.

106. *Ролик А.И.* Информационная технология управления корпоративной ИТ-инфраструктурой: дис. ... доктора техн. наук: 05.13.06 / А.И. Ролик. – К., 2013. – 456 с.

107. *Ролик А.И.* Комплексный подход к управлению уровнем ИТ-услуг в корпоративных ИТ-инфраструктурах/ А.И. Ролик // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, 2014. – №61. – С. 148–161.

108. *Ролик А.И.* Концепция управления корпоративной ИТ-инфраструктурой / А.И. Ролик // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2012. – № 56. – С. 31–55.

109. *Ролик А.И.* Метод нечеткой непараметрической оценки качества функционирования элементов информационно-телекоммуникационных систем /

А.И. Ролик, Д.А. Галушко, П.Ф Можаровский [и др.] // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2012. – Вип. 19 (39). – С. 115–125.

110. *Ролик А.И.* Метод потенциальных функций в задачах оценки уровня телекоммуникационных сервисов / А.И. Ролик, Т.И. Ланге, А.А. Покотило, Б.А. Март, М.В. Ясочка // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2012. – № 57. – С. 133–143.

111. *Ролик А.И.* Метод оценки состояния элементов информационно-телекоммуникационных систем с использованием нейронных сетей / А.И. Ролик, Ю.А. Кононенко // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2012. – № 57. – С. 8–14.

112. *Ролик А.И.* Методика оперативного управления уровнем ИТ-услуг на основе сервисно-ресурсных моделей/ А.И. Ролик // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, 2014. – № 60. – С. 119–129.

113. *Ролик А.И.* Модель подсистемы управления информационными потоками с адаптивным регулятором/ А.И. Ролик, Ю.Н. Пошак, М.А. Никоненко, А.В. Мельник // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2010. – Вип. 16 (36). – С. 94–103.

114. *Ролик А.И.* Модель управления перераспределением ресурсов информационно-телекоммуникационной системы при изменении значимости бизнес-процессов// Автоматика. Автоматизация. Электротехнические комплексы и системы. – 2007. – №2 (20). – С. 73–82.

115. *Ролик А.И.* Оценка качества предоставления сервиса IPTV на основе метода потенциальных функций / А.И. Ролик, А.А. Покотило, Р.О. Боровик, Ю.И. Бабичук // Интеллектуальный анализ информации «ИАИ-2013»: междунар. науч. конф. им. Т.А. Таран, 15–17 мая 2013 г.: сб. тр. – К.: Просвіта, 2013. – С. 204–210.

116. *Ролик А.И.* Применение глубинно-упорядоченных регионов в системах управления ИТ инфраструктурой / А.И. Ролик, П.Ф. Можаровский, Б.А. Март // Интеллектуальный анализ информации «ИАИ-2010»: X междунар. науч. конф. им. Т.А. Таран, 19–21 мая 2010 г.: сб. тр. – К.: Просвіта, 2010. – С. 214–221.

117. *Ролик А.И.* Распределение мобильных компонентов системы управления информационно-телекоммуникационной системой / А.И. Ролик, Р.Л. Соколовский // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: Век+, 2007. – №47. – С. 113–124.

118. *Ролик А.И.* Сетевая модель обеспечения качества сервиса при передаче мультимедийной информации в глобальных сетях / А.И. Ролик, О.В. Ружанская // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: Век+, 2007. – №46. – С. 187–205.

119. *Ролик А.И.* Система управления информационными потоками в корпоративной IP-сети / А.И. Ролик, В.А. Иосифов // Адаптивні системи автоматичного управління. Міжвідомчий науково-технічний збірник. – Дніпропетровськ: Системні технології, 2009. – Вип. 14 (34). – С. 73–85.

120. *Ролик А.И.* Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2010. – №52. – С. 39–52.

121. *Ролик А.И.* Тенденции и перспективы развития управления информационными технологиями / А.И. Ролик // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2012. – №55. – С. 81–109.

122. *Ролик А.И.* Управление уровнем услуг корпоративной ИТ-инфраструктуры на основе координатора / А.И. Ролик //Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, 2013. – №59. – С. 98–105.

123. *Ролик А.И.* Управление устранением неисправностей в ИТ-системах / А.И. Ролик, Ю.С. Тимофеева, Н.И. Турский // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: Век+, 2008. – №49. – С. 94–107.

124. *Ролік О.І.* Метод зведення метрик якості функціонування компонентів ІТ-інфраструктури за допомогою апарату непараметричної статистики / О.І. Ролік, П.Ф. Можаровский, В.М. Вовк, Д.С. Захаров // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2011. – №53. – С. 160–169.

125. *Ролік О.І.* Метод управління резервуванням ресурсів ЦОД для підтримки заданої якості обслуговування запитів користувачів / О.І. Ролік, П.Ф. Можаровський, О.С. Тонковид // Зб. наук. праць ВІТІ НТУУ «КПІ». – Випуск № 2. – К.: ВІТІ НТУУ «КПІ», 2010. – С. 97–104.

126. *Ролік О.І.* Моделювання управління потоками даних в корпоративних IP-мережах/ О.І. Ролік, Ю.М. Пошак, М.А. Никоненко, А.В. Мельник // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2011. – Вип. 18(38). – С. 94–103.

127. *Рохилла С.* Microsoft ADO.NET: разработка профессиональных проектов/ С. Рохилла. – СПб: «БХВ-СПб», 2003. – 768 с.

128. Руководство к своду знаний по управлению проектами. (Руководство PMBOK)/ Американский национальный стандарт ANSI/PMI 99-001-2004. 3-е изд., на рус. языке. – Project Management Institute, Inc., 2004. – 238 с.

129. *Рутковская Д.* Нейронные сети, генетические алгоритмы и нечеткие системы // Д. Рутковская, М. Пилиньский, Л. Рутковский. – 2-е изд. – М.: Горячая линия-Телеком, 2008. – 452 с.

АУТСКИ

130. *Салтыков С.А.* Экспериментальное сопоставление методов взвешенной суммы, теории полезности и теории важности критериев для решения многокритериальных задач с балльными критериями / С. А. Салтыков // Управление большими системами. Вып. 29. – М.: ИПУ РАН, 2010. – С.16–41.

131. *Самофалов К.Г.* Электронные цифровые вычислительные машины: Учебник/ К.Г. Самофалов, В.И. Корнейчук, В.П. Тарасенко. – К.: Высшая школа, 1976. – 480 с.

132. *Себеста Р.У.* Основные концепции языков программирования / Р.У. Себеста. – 5-е изд. – М.: Вильямс, 2001. – С. 672.

133. *Сигнал И.Х.* Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: Учеб. пособие / И.Х. Сигнал, А.П. Иванова. – Изд. 2-е, испр. – М.: ФИЗМАТЛИТ, 2003. – 240 с.

134. *Сипко Е.Н.* Метод последовательного анализа вариантов решения задачи составления расписания занятий/ Е.Н. Сипко // Искусственный интеллект. – 2011. – №1. – С. 243 – 250.

135. *Скуратов А.К.* Статистический мониторинг и анализ телекоммуникационных сетей / А.К. Скуратов // Вестник Московского авиационного института. – 2007. – Т. 14, – №1.

136. *Скурихин А.Н.* Генетические алгоритмы / А.Н. Скурихин //Новости искусственного интеллекта. – 1995. – №4. – С. 6–46.

137. *Советов Б.Я.* Информационные технологии: Учеб. для вузов/ Б.Я. Советов, В.В. Цехановский. – 3-е изд., стер. – М.: Высшая школа, 2006. – 263 с.

138. *Советов, Б. Я.* Теоретические основы автоматического управления: Учебник / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – М.: Высшая школа, 2006. – 463 с.

139. *Синюк В.Г.* Управляемые статистические генетические алгоритмы/ В.Г. Синюк, В.Н. Акопов // Программные продукты и системы. – 2008. – №4. – С. 1–6.

140. *Судаков О.О.* GRID-ресурси інформаційно-обчислювального центру Київського національного університету імені Тараса Шевченка / О.О. Судаков, Ю.В. Бойко // Проблеми програмування. – 2006. – №2–3. – С. 165–169.

141. *Теленик С.Ф.* Адаптивный генетический алгоритм для решения класса задач распределения ресурсов ЦОД / С.Ф. Теленик, А.И. Ролик, П.С. Савченко // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: В Век+, 2011. – №54. – С. 164–174.

142. *Теленик С.Ф.* Аналіз підходів до визначення впливу неполадок елементів мережі доступу на якість обслуговування клієнтів / С.Ф. Теленик, О.І. Ролік, М.В. Ясочка, К.О. Голоднов // Вісник Вінницького політехнічного інституту. – 2011. – №3. – С. 69–83.

143. *Теленик С.Ф.* Генетичні алгоритми вирішення задач управління ресурсами і навантаженням центрів оброблення даних / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, С.А. Андросов // Автоматика. Автоматизація. Електротехнічні комплексі та системі. – 2010. – №1(25). – С. 106–120.

144. *Теленик С.Ф.* Забезпечення процесів діяльності ресурсами з визначеним рівнем надійності в інформаційно-телекомунікаційній системі спеціального призначення / С.Ф. Теленик, О.І. Ролік, П.І. Терещенко, М.М. Букасов // Зб. наук. праць ВІТІ НТУУ «КПІ». – №3. – К.: ВІТІ НТУУ «КПІ», 2006. – С. 134–138.

145. *Теленик С.Ф.* Зведення метрик оцінювання рівня обслуговування користувачів на основі експертних оцінок / С.Ф. Теленик, О.І. Ролік, О.М. Моргаль, О.С. Квітко// Вісник Вінницького політехнічного інституту. – 2011. – №1. – С. 112–123.

146. *Теленик С.Ф.* Моделі і методи розподілу ресурсів в системах з серверною віртуалізацією / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, О.А. Косован, О.І. Кобец // Зб. наук. праць ВІТІ НТУУ «КПІ». – Випуск №3. – К.: ВІТІ НТУУ «КПІ», 2009. – С. 100–109.

147. *Теленик С.Ф.* Моделі управління віртуальними машинами при серверній віртуалізації / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, А.Ю. Лабунський // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2009. – №51. – С. 147–152.

148. Т*еленик С.Ф.* Моделі управління розподілом обмежених ресурсів в інформаційно-телекомунікаційній мережі АСУ/ С.Ф. Теленик, О.І. Ролік, М.М. Букасов // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: Єкотех, 2006. – №44. – С. 234–239.

149. *Теленик С.Ф.* Модель управління розподілом ресурсів інформаційно-телекомунікаційної системи збройних сил України / С.Ф. Теленик, О.І. Ролік, П.І. Терещенко, О.В. Літвінцов // Наук.-техн. зб. ННДЦ оборонних технологій і воєнної безпеки України. – 2006. – №5(34). – С. 117–124.

150. *Теленик С.Ф.* Метод определения влияния неполадок на качество сервиса в IP-сетях с протоколами RIP и OSPF/ С.Ф. Теленик, А.И. Ролик, М.В. Ясочка, М.Е. Бутенко // Проблеми інформатизації та управління: Зб. наук. Праць. Вип. 1(37). – К.: НАУ, 2012. – С. 118–126.

151. *Теленик С.Ф.* Методы диагностики компонентов информационно-телекоммуникационных систем/ С.Ф. Теленик, А.И. Ролик, Ю.С. Тимофеева // «Наукові вісті» Інституту менеджменту і економіки «Галицька академія». – Івано-Франківськ, 2009. – №1(15). – С. 49–58.

152. *Теленик С.Ф.* Нечітке оцінювання в задачах управління рівнем обслуговування / С.Ф. Теленик, О.І. Ролік, М.В. Ясочка, О.М. Моргаль // Наукові записки УНДІЗ. – 2011. – №2(18). – С. 29–42.

153. *Теленик С.Ф.* Определение распространения влияния неисправностей в сети доступа на качество предоставляемых сервисов / С.Ф. Теленик, А.И. Ролик, М.М. Букасов, М.В. Ясочка// Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: Век+, 2009. – №50. – С. 164–173.

154. *Теленик С.Ф.* Система управления информационно-телекоммуникационной системою корпоративной АСУ / С.Ф. Теленик, А.И. Ролик, М.М. Букасов, Р.Л. Соколовский // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. – К.: Век+, 2006. – №45. – С. 112–126.

155. *Теленик С.Ф.* Система управления информационной инфраструктурой транспортного предприятия / С.Ф. Теленик, А.И. Ролик, П.Ф. Можаровский, А.В. Волошин // Автомобільний транспорт: зб. наук. праць. Вип. 25. – Харків: ХНАДУ, 2009. – С. 242–245.

156. *Теленик С.Ф.* Технологія управління ІТ-інфраструктурою на основі ресурсного підходу/ С.Ф. Теленик, О.І. Ролік, М.М. Букасов // Вісник ЖДТУ. – 2008. – №4(47). – С. 180–189.

157. *Теленик С.Ф.* Управління доступом до обмежених ресурсів інформаційно-телекомунікаційної мережі АСУ військового призначення / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, П.І. Терещенко // Зб. наук. праць Центрального НДІ Збройних Сил України. – 2006. – №3(37). – С. 33–43.

158. *Теленик С.Ф.* Управління навантаженням і ресурсами центрів оброблення даних при віртуальному хостингу / С. Теленик, О. Ролік, М. Букасов, С. Андросов, Р. Римар // Вісн. Тернопільского держ. техн. ун-ту. – 2009. – Т. 14, №4. – С. 198–210.

159. *Теленик С.Ф.* Управління навантаженням і ресурсами центрів оброблення даних при виділених серверах / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, Р.В. Римар, К.О. Ролік // Автоматика. Автоматизація. Електротехнічні комплексі та системі. – 2009. – №2(24). – С. 122–136.

160. *Теленик С.Ф.* Управління ресурсами центрів оброблення даних / С. Теленик, О. Ролік, М. Букасов, К. Крижова // Вісник Львів. ун-ту. Серія прик. матем. інформ. – 2009. – Вип. 15. – С. 325–340.

161. *Теленик С.Ф.* Управляемый генетический алгоритм в задачах распределения виртуальных машин в ЦОД / С.Ф. Теленик, А.И. Ролик, П.С. Савченко, М.Е. Боданюк // Вісник ЧДТУ. – 2011. – №2. – С. 104–113.

162. *Теленик С.Ф.* Швидке розроблення застосувань в адаптивній технології SmartBase / С.Ф. Теленик, О.А. Амонс, В.С. Хмелюк, К.О. Крижова // Проблемы программирования. – 2006. – №2–3. Спец. вып. – С. 299–305.

163. *Теленик С.Ф.* Швидкий алгоритм визначення впливу неполадок елементів мережі доступу на якість обслуговування клієнтів / С.Ф. Теленик, О.І. Ролік, М.В. Ясочка, Д.В. Малюгін // Інформаційні технології та комп'ютерна інженерія. – 2010. – №2(18). – С. 51–62.

164. *Тржец К.* Дистанционное управление программным обеспечением с помощью агентов/ К.Тржец, Г. Ежич, М. Кушек, С. Дешич // Ericsson Nikola Tesla, REVIJA. – 2005. – vol. 18, no. 41. – P. 57–67.

165. Управління високопродуктивними ІТ-інфраструктурами / Ю.В. Бойко, М.М. Глибовець, С.В. Єршов, С.Л. Кривий, С.Д. Погорілий, О.І. Ролік, С.Ф. Теленик, М.В. Ясочка// Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, 2014. – №61. – С. 120–141.

166. *Фаулер М.* Архитектура корпоративных программных приложений: пер. с англ. / М. Фаулер. – М.: ДМК Пресс, 2003. – 544 с.

167. *Хайкин С.* Нейронные сети. Полный курс // С. Хайкин. – Вильямс, 2006. – 1104 с.

168. *Хант Э.* Искусственный интеллект / Пер. с англ. под ред. В. Л. Стефанюка. – М.: Мир, 1978. – 558 с

169. *Холланд Дж.* Генетические алгоритмы / Дж. Холланд // В мире науки. – 1992. – №9–10. – С. 32–40.

170. *Хэрри М.* 6 Sigma. Концепция идеального менеджмента/ М. Хэрри, Р. Шредер. – М.: Эксмо, 2003. – 464 с.

171. *Черняк Л.* Корпоративное управление: первые шаги// Открытые системы. – 2007. – №8. – С. 36–42.

172. *Шварц М.* Сети связи: протоколы моделирование и анализ: в 2-х частях. Ч.I, II. – М.: Наука, 1992.

173. *Яремко І.* Методика оптимізації ресурсів центрів обробки даних / І. Яремко // Комп'ютерні технології друкарства, наук. видання. – Львів, 2011. Вип. 26. – С. 173–179.

174. *Яремко І.М.* Управління розподілом ресурсів центрів обробки даних телекомунікаційної мережі / І.М. Яремко, В.В. Турупалов // Искусственный интеллект. – 2011. – №4. – С. 380–385.

175. *Abu Osman M.T.* On the direct product of fuzzy subgroups // Fuzzy Sets and Systems. – 1984. – Vol. 12. – P. 87–91.

176. *Almes G.* A One-way Delay Metric for IPPM / G. Almes, S. Kalidindi, M. Zekauskas. – RFC 2679. – September 1999.

177. *Almes G.* A One-way Packet Loss Metric for IPPM / G. Almes, S. Kalidindi, M. Zekauskas. – RFC 2680. – September, 1999.

178. *Almes G.* A Round-trip Delay Metric for IPPM / G. Almes, S. Kalidindi, M. Zekauskas RFC 2681. – September, 1999.

179. *Andreozzi S.* On the Integration of Passive and Active Network Monitoring for Grid Systems / S. Andreozzi, A. Ciuffoletti, A. Ghiselli, D. Antoniades, E.P. Markatos, M. Polychronakis, P. Trimintzios // Integrated Research in GRID Computing, Springer-Verlag, January 2007. – P. 147–161.

180. Application Management (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. – 4th impression. – 2006. – 158 p.

181. *Ardagna D.* SLA based resource allocation policies in autonomic environments/ D. Ardagna, M. Trubian, L. Zhang // Journal of Parallel and Distributed Computing. – Vol. 67, Is. 3, March, 2007. – P. 259–270.

182. *Aron M.* Cluster reserves: A mechanism for resource management in cluster-based network servers / M. Aron, P. Druschel, W. Zwaenpoel // Proc. ACM. Measurement and Modeling of Computer Systems – SIGMETRICS, Santa Clara, CA, Jun. 2000. – P. 90–101.

183. *Asgari A.* Scalable monitoring support for resource management and service assurance / A. Asgari, R. Egan, P. Trimintzios, G. Pavlou // IEEE Network. – 2004. – Vol. 18, №6. – P. 6–18.

184. *Awduche D.O.* MPLS and traffic engineering in IP networks/ D.O. Awduche // IEEE Communications Magazine. – 1999, December. – P. 42–47.

185. *Back T.* Optimal Mutation Rates in Genetic Search / T. Back // Fifth International Conference on Genetic Algorithms: University of Illinois at Urbana-Champaign, July 17–21, 1993. – P. 2–8.

186. *Babcock B.* Distributed Top-K monitoring / B. Babcock, C. Olston // Proc. of the ACM SIGMOD international conference on Management of data. – San Diego, CA, USA, 2003. – P. 28–39.

187. *Baker F.* Requirements for IP Version 4 Routers / F. Baker. – RFC 1812. – June, 1995.

188. *Bartolini C.* Business-impact analysis and simulation of critical incidents in IT service management / C. Bartolini, C. Stefanelli, M. Tortonesi // 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009). – 2009. – Vol. 11, №1. – June 2009. – P. 9–16.

189. *Bejerano Y.* Robust monitoring of link delays and faults in IP networks / Y. Bejerano, R. Rastogi // IEEE ACM Transactions on networking. – 2006. – Vol.14, №5. – P. 1092–1103.

190. *Bergnum K.* On the stability of adaptive service management / K. Bergnum, M. Burgess, T.M. Jonassen, S. Fagernes // eTransactions on Network and Service Management. – 2006. – Vol.2, №1. – P. 13–21.

191. *Bernard P.* Foundations of ITIL / P. Bernard. – Van Haren Publishing, 2012. – 460 p.

192. *Biazetti A.* Integrating business performance management with IT management through impact analysis and provisioning / A. Biazetti, G. Goldszmidt // 11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008). – 2008. – Vol. 11, №1. Apr. 2008. – P. 504–518.

193. *Bilous R.* Adaptive Genetic Algorithm for Load Balancing in Computer Networks / R. Bilous, S. Pogorilyy //Journal of Qafqaz University. Mathematic and Computer Science. – 2012. – № 34. – P. 34 – 41.

194. *Boehm S.* Aggregation of Real-Time System Monitoring Data for Analyzing Large-Scale Parallel and Distributed Computing Environments / S. Boehm, C. Engelmann, S.L. Scott // High Performance Computing and Communications (HPCC), 2010 12th IEEE Int. Conf. 1–3 Sept. 2010. – 2010. – P. 72–78.

195. *Boros S.* Policy-based network management with SNMP / S. Boros // Proceedings of the 6th Eunice Summer School, September 2000. – P. 165–172.

196. *Boser B.E.* A training algorithm for optimal margin classifiers/ B.E. Boser, I.M. Guyon, V.N. Vapnic // ACM. – 1992. – P. 1–7.

197. *Boutaba R.* CyberPlanner: A Comprehensive toolkit for network service Providers / R. Boutaba, J. Xiao, I. Aib // NOMS 2008 – 11th IEEE/IFIP Network Operations and Management Symposium. – 2008. – Vol. 11, N. 1. – P. 379–386.

198. *Bradner S.* Benchmarking Terminology for Network Interconnection / S. Bradner. – RFC1242. – July, 1991.

199. *Braden R.* Requirements for Internet Hosts – Application and Support / R. Braden. – RFC 1123. – October, 1989.

200. *Braden R.* Requirements for Internet Hosts – Communication Layers / R. Braden. – RFC 1122. – October, 1989.

201. *Bradner S.* Benchmarking Methodology for Network Interconnect Devices / S. Bradner, J. McQuaid. – RFC 2544. – March, 1999.

202. *Breitgand D.* Hierarchical reactive monitoring of multicast membership size / D. Breitgand, D. Dolev, D. Raz // IEEE INFOCOM 2003. – The Conference on Computer Communications. – 2003.

203. *Buco M.J.* Utility computing SLA management based upon business objectives/ M.J. Buco, R.N. Chang, L.Z. Luan, C. Ward, J.L. Wolf, P.S. Yu// IBM Systems Journal. – 2004. – Vol.43, №1. – P. 159–178.

204. *Building* Bridges: ITIL and eTOM, Release 1.0: TR143. – TM Forum, 2009, July. – 76 p.

205. *Bulut A.* Optimization techniques for reactive network monitoring / A. Bulut, N. Koudas, A. Meka // Department of Computer Science, University of California, Santa Barbara. Doc.: 2007-01. – 2007. – March. – 18 p.

206. Business Perspective (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. Vol. 1. – 2th impression. – 2005. – 180 p.

207. *Cai Z.* Automated Availability Management Driven by Business Policies/ Z. Cai, Y. Chen, V. Kumar, D. Milojicic, K. Shwan // 10th IFIP/IEEE Symposium on Integrated Management (IM). – 2007. – P. 264–273.

208. *Cannon D.* ITIL Service Strategy. 2011 Edition / D. Cannon. – UK: TSO, Norwich. – 2011. – 496 p.

209. *Cardellini V.* Dynamic load balancing on Web-server systems / V. Cardellini, M. Colajanni, P. Yu // IEEE Internet. Computing, 3, 1999. – P. 28–39.

210. *Case J.* Simple Network Management Protocol / J. Case, M. Fedor, M. Schoffstall, J. Davin. – SNMP Research, Performance Systems International, MIT Laboratory for Computer Science. – RFC 1157. – May 1990.

211. *Cascos I.* Data Depth: Multivariate Statistics and Geometry / I. Cascos // New Perspectives in Stochastic Geometry / S. Kendall, I. Molchanov. – Oxford.: Oxford University Press, 2009. – P. 398–425.

212. *Catanis I.* An Introduction to the cognitive concept as a cost-effective method for network performance management // Wireless Communications and Networking Conference, IEEE. – 2008. – P. 3197–3201.

213. *Chase J.* Managing Energy and Server Resources in Hosting Centers / J. Chase, D. Anderson, P. Thakar, A. Vahdat// Proc. of the Eighteenth ACM Symposium on Operating Systems Principles (SOSP), October 2001. – P. 103–116.

214. Cisco Advanced Services Network Management Systems Architectural Leading Practice. White Paper. – Cisco Systems, Inc. 2007. – 11 p.

215. COBIT 4.1. Российское издание. IT Governance Institute. – М.: Аудит и контроль информационных систем, 2008. – 230 с.

216. *Cockayne W.R.* Mobile Agents / W.R. Cockayne, M. Zyda. – N J. USA: Prentice Hall, Englewood Cliffs. 1998. – 312 p.

217. *Coombs C.* Some Views on Mathematical Models and Measurement Theory / C. Coombs, H. Raiffa, R. Thral // Psychological Review. – 1954. – Vol.61(2). – P. 132–144.

218. *Cormode G.* Algorithms for Distributed Functional Monitoring / G. Cormode, S. Muthukrishnan, K. Yi // Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms. – San Francisco, California, 2008. – P. 1076–1085.

219. *Courtois P.J.* On time and space decomposition of complex structures// Communications of the ACM – 1985. – Vol. 28 (6). – P. 590–603.

220. *Crovella M.E.* Self-similarity in world wide web traffic: Evidence and possible causes / M.E. Crovella, A. Bestavros // IEEE/ACM Trans. on Networking – 1997, Dec. – Vol. 5. – P. 835–846.

221. Defining And Measuring IT Efficiency: Maximizing Return On Storage Investments // A Forrester Consulting Thought Leadership Paper Commissioned By NetApp. – Forrester Research, Inc. 2011. – March. – 21 p.

222. Department of Defense Trusted Computer System Evaluation Criteria (TCSEC). DoD 5200.28-STD. – December 26. – 1985. – 216 p.

223. *Dilman M.* Efficient reactive monitoring / M. Dilman, D. Raz // IEEE JSAC. – 2002. – Vol. 20, №4. – P. 668–676.

224. *D'Inverno M.* Understanding Agent Systems / M. D'Inverno, M. Luck. – Berlin: Springer-Verlag, 2001. – 191 p.

225. *Donoho D. L.* Breakdown properties of multivariate location estimators / D.L. Donoho // Ph. D. Qualifying Paper. – Harvard University, 1982.

226. *Dyckerhoff R.* Computing zonoid trimmed regions of bivariate data sets. / R. Dyckerhoff // COMPSTAT Proc. in Computational Statistics. – 2000. – P. 29–30.

227. *Dyckerhoff R.* Zonoid data depth: theory and computation / R. Dyckerhoff, G. Koshevoy, K. Mosler // A. Pratt, ed., COMPSTAT Proc. in Computational Statistics. – 1996, Nov. – P. 235–240.

228. Dynamic Virtual Machine Allocation Based on Adaptive Genetic Algorithm / O. Rolik, S. Telenyk, E. Zharikov, V. Samotyy // Proc. CLOUD COMPUTING 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, February 19-23, 2017, Athens, Greece, IARIA. – 2017. – P. 108–114.

229. *Federgruen A.* The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality/ A. Federgruen, H. Groenvelt // Oper. Res. 34, 1986. – P. 908–918.

230. *Foundations* of IT Service Management, based on ITIL/ ITSMF-NL// Van Haren Publishing, Second edition, 2005. – 231 p.

231. *Frankel D.S.* Model driven architecture: applying MDA to enterprise computing / D.S. Frankel. –N. Y.: Wiley, 2003. – 328 p.

232. *Foster I.* The Grid: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. – USA, San Francisco, California: Morgan Kaufmann Publishers, Inc. 1999. – 677 p.

233. *Frey J.* Condor-G: A computation management agent for multiinstitutional Grids / J. Frey, T. Tannenbaum, J. Foster // Cluster Computing. – 2002. – №3(5). – P. 237–246.

234. *Fricke S.* Agent-based Telematic Services and Telecom Applications / S. Fricke, K. Bsufka, J. Keiser, T. Schmidt, R. Sesseler, S. Albayrak// Communications of the ACM. – 2001. – Vol. 44, №4. – P. 43–48.

235. *Flood R.* Dealing with Complexity: An Introduction to the Theory and Application of Systems Science / R. Flood, E. Carson. – Second edit. – New York, NY: Plenum Press, 1993. – 280 p.

236. *Gabès J.* Nagios 3 pour la supervision et la métrologie / J. Gabès. – Eyrolles. 2009. – 482 p.

237. *Garbani J.-P.* Competitive Analysis: Application Performance Management And Business Transaction Monitoring/ J.-P. Garbani. – Forrester Research, Inc. 2010. – Sep. 9. – 29 p.

238. *Garbani J.-P.* IT Infrastructure And Operations: The Next Five Years / J.-P. Garbani, M. Cecere. – Forrester Research, Inc. 2011. – May 3. – 20 p.

239. *Garbani J.-P.* It's Time For IT Management Software 2.0 / J.-P. Garbani, T. Mendel, E. Hubbert. – Forrester Research, Inc. 2009. – Dec. 8. – 13 p.

240. *Garbani J.-P.* Market Overview: Application Performance Management, Q4 2011 / J.-P. Garbani, E. Hubbert, D. Washburn, L. Kempton. – Forrester Research, Inc. 2011. – Dec. 7. – 17 p.

241. *Garbani J.-P.* Market Projections For 2010: IT Management Software / J.-P. Garbani, T. Mende. – Forrester Research, Inc. 2010. – Feb. 4. – 6 p.

242. *Garbani J.-P.* The IT Management Software Megavendors // J.-P. Garbani, P. O'Neill. – Forrester Research, Inc. 2009. – Aug. 12. – 14 p.

243. *Garbani J.-P.* The Key To IT Business Alignment Is In Operations/ J.-P. Garbani, S. Yates, R. Batiancila. – Forrester Research, Inc. 2007. – Apr. 2. – 14 p.

244. *Garbani J.-P.* Who's Who In IT Management Software 2.0 / J.-P. Garbani, T. Mendel// Forrester Research, Inc. 2010. – Aug. 12. – 13 p.

245. *Glibovets N.N.* A review of niching genetic algorithms for multimodal function optimization / N.N. Glibovets, N.M. Gulayeva // Cybernetics and Systems Analysis. – 2013. – Vol. 49, №6. – P. 815-820.

246. *Glibovets N.N.* Genetic algorithms used to solve a scheduling problem / N.N. Glibovets, S.A. Medvid // Kibernetika i Sistemnyj Analiz. – 2003. – №1. – P. 95-108.

247. *Gont F.* ICMP Attacks against TCP / F. Gont. – RFC 5927. – July, 2010.

248. *Grancharov V.* Non-intrusive speech quality assessment with low computational complexity / V. Grancharov, D. Y. Zhao, J. Lindblom, W. Bastiaan Kleijn // INTERSPEECH 2006 – ICSLP, Ninth Inter. Conf. on Spoken Language Processing, Pittsburgh, PA, USA, September 17–21, 2006. ISCA. – 2006.

249. Grid resource management: State of the Art and Future Trends / Edited by J. Nabrzyski, J. M. Schopf, J. Weglars. – NY: Kluwer Academic Publishers, 2004. – 574 p.

250. *Habib A.* On detecting service violations and bandwidth theft in QoS network domains/ A. Habib, S. Fahmy, S. Avasarala // Computer Communications. – 2003. – № 26. – P. 861–871.

251. *Hamscher V.* Evaluation of job-scheduling strategies for grid computing / V. Hamscher, U. Schwiegeishohn, A. Streit, R. Yahyapour // Lecture Notes in Computer Science. – 2000. – P. 191–202.

252. *Hanemann A.* Towards a framework for failure impact analysis and recovery with respect to service level agreements / A. Hanemann, D. Schmitz, M. Sailer // IFIP/IEEE International Symposium on Integrated Network Management (IM 2005). – №1. – May 2005. – 2005. – P. 1189–1192.

253. *Harte L.* Service Quality Monitoring, Analyzing, and Diagnostics for IP Television Systems and Services / L. Harte. – Althos Publishing, 2008. – 116 p.

254. *Haupt R.L.* Practical genetic algorithms / R.L. Haupt, S.E. Haupt. – John Wiley & Sons. 2004. – 272 p.

255. *He J.* Towards Robust Multi-layer Traffic Engineering: Optimization of Congestion Control and Routing/ J. He, M. Bresler, M. Chiang, J. Rexford// IEEE Journal on Selected Areas in Communications. – 2007. – Vol. 25, Issue 5. – P. 868–880.

256. *Hill P.* Combine ITIL and COBIT to Meet Business Challenges / P. Hill, K. Turbitt. – BMC Software, Inc. 2007, March 15. – 16 p.

257. *Hoppermann J.* The Banking Platform Of The Future / J. Hoppermann, P. Murphy, A. Knoll. – Forrester Research, Inc. 2009. – Dec. 8. – 18 p.

258. *Hubbert E.* TechRadar™ For I&O Professionals: IT Service Management Processes, Q1 2012 / E. Hubbert, J.P. Garbani, G. O'Donnell, S. Mann, J. Rakowski. – Forrester Research, Inc. 2012. – Feb. 7. – 44 p.

259. *Ibaraki T.* Resource Allocation Problems-Algorithmic Approaches / T. Ibaraki, N. Katoh. – The MIT Press, Cambridge, MA, 1988. – 246 p.

260. ICT Infrastructure Management (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. – 6th impression. – 2005. – 286 p.

261. Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model: ISO/IEC 7498-1:1994. Second editions, 1994. – 68 p.

262. Information technology. Service management. Part 1: Specification: ISO/IEC 20000-1:2005. – ISO/IEC, 2005. – 16 p.

263. Information technology. Service management. Part 2: Code of practice: ISO/IEC 20000-1:2005. – ISO/IEC, 2005. – 34 p.

264. IT Service Management: An Introduction// Jan Van Bon, G. Kemmerling, D. Pondman, Publisher: Van Haren Publishing, 2002.

265. ITIL v3 Glossary Russian Translation v0.92, 30 Apr 2009, (Глоссарий Терминов и Определений) itSMF / ITIL® V3 Translation Project. 2009. – 147 p.

266. Introducing the IBM process reference model for IT. IBM Corporation, 2008. – 21 p.

267. *Jan O.* NAGIOS au cœur de la supervision Open Source. De l'installation à l'optimisation / O. Jan. – France: Editions ENI, 2008. – 374 p.

268. *Jiao J.* Toward efficient monitoring / J. Jiao, S. Naqvi, D. Raz, B. Sugla // IEEE Journal on Selected Areas in Communications. – 2000. – May, 18 (5). – P. 723–732.

269. *Julstrom B.A.* What Have You Done for me Lately? Adapting Operator Probabilities in a Steady-State Genetic Algorithm / B.A. Julstrom // Proc. of the Sixth International Conference on Genetic Algorithms: University of Pittsburgh, July 15–19, 1995. – P. 81–87.

270. *Karve A.* Dynamic Placement for Clustered Web Applications / A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder // Proc. 15th International Conference on World Wide Web. – ACM New York, NY, USA, 2006. – P. 595–604.

271. *Kashyap S.* Efficient constraint monitoring using adaptive thresholds / S. Kashyap, J. Ramamirtham, R. Rastogi, P. Shukla // ICDI conference. – 2008.

272. *Kendall W.S.* New perspectives in stochastic geometry / W.S. Kendall, I. Molchanov – Oxford University Press, 2010. – 608 p.

273. *Kimbrel T.* Dynamic application placement under service and memory constraints / T. Kimbrel, M. Steinder, M. Sviridenko, A. Tantawi// Proc. 4th international conference on Experimental and Efficient Algorithms (WEA'05), Santorini Island, Greece, May 2005. – P. 391–402.

274. *Knapik M.* Developing Intelligent Agents for Distributed Systems: Exploring Architecture, Technologies, and Applications/ M. Knapik, J. Johnson. – NY. USA: McGraw-Hill, 1997. – 389 p.

275. *Koshevoy G.* Zonoid trimming for multivariate distributions / G. Koshevoy, K. Mosler // Ann. Statist. – 1997, Oct. – Vol. 25, №5. – P. 1998–2017.

276. *Kosorukoff A.* Using incremental evaluation and adaptive choice of operators in a genetic algorithm / A. Kosorukoff // Proc. of the Genetic and Evolutionary Computation Conf.: (GECCO-2002), New York, USA, July 9–13, 2002. – P. 688.

277. *Kratochvíl T.* Digital Video Image Quality / T. Kratochvíl, M. Slanina // Digital Video, F. De Rango (Ed.), InTech, 2010. – P. 487–500.

278. *Kryvyi S.L.* Design of Grid Structures on the Basis of Transition Systems with the Substantiation of the Correctness of Their Operation / S.L. Kryvyi, Y.V. Boyko, S.D. Pogorilyy, O.F. Boretskyi, M.M. Glybovets // Cybernetics and Systems Analysis. – Springer Science + Business Media, N. Y. – Jan., 2017. – V. 53. – Is. 1. – P. 105–114.

279. *Lambert L.* Topic Overview: IT Management Software / L. Lambert, S. Yates, G. Schreck, E. Hubbert, T. M., J-P. Garbani. – Forrester Research, Inc. 2007. – Nov. 13. – 11 p.

280. *Leopoldi R.* IT Service Management: The Role of Service Request Management/ R. Leopoldi // RL Information Consulting LLC. – 2007. – June, 1. – 9 p.

281. *Leung K.S.* A New Model of Simulated Evolutionary Computation – Convergence Analysis and Specifications / K.S. Leung, Q.H. Duan, Z.B. Xu, C.K. Wong//

IEEE Transactions on Evolutionary Computation, Vol. 5, N. 1. – February, 2001. – P. 3–16.

282. *Lis J.* Self-adapting Parallel Genetic Algorithms with the Dynamic Mutation Probability, Crossover Rate and Population Size / J. Lis, M. Lis // Proc. of the 1st Polish National Conference on Evolutionary Computation. Oficina Wydawnica Politechniki, Warszawskiej. – 1996. – P. 324–329.

283. *Mahdavi J.* IPPM Metrics for Measuring Connectivity / J. Mahdavi, V. Paxson. – RFC 2678. – September, 1999.

284. *Malkin G.* RIP Version 2./ G. Malkin. – RFC2453. – Nov., 1998. – 39 p.

285. Management framework for open systems interconnection (OSI) for CCITT applications: ITU-T Rec. X.700. – Geneva. – 1992. – 11 p.

286. Managing the Quality of Customer Experience, Release 1.0: Technical Report: TR148. – TM Forum. – Nov., 2009. – 121 p.

287. *Mamdani E.H.* Applications of fuzzy logic to approximate reasoning using linguistic synthesis // IEEE Transactions on Computers. – 1977. – Vol. 26, №12. – P. 1182–1191.

288. *Mandeville R.* Benchmarking Terminology for LAN Switching Devices / R. Mandeville. – RFC 2285. – February, 1998.

289. *Marshall D.* Advanced Server Virtualization: VMware and Microsoft Platforms in the Virtual Data Center / D. Marshall, W. A. Reynolds, D. McCrory. – Auerbach Publications, 2006. – 742 p.

290. *McCloghrie K.* Management Information Base for Network Management of TCP/IP-based Internets / K. McCloghrie, M. Rose. – Hughes LAN Systems and Performance Systems International. – RFC 1156. – May, 1990.

291. Measurement of the quality of service. Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference. Rec. ITU-T J.144: ITU-T Recommendation. – 2001. – 70 p.

292. Measurement of the quality of service. Full reference and reduced reference calibration methods for video transmission systems with constant misalignment of spatial and temporal domains with constant gain and offset. Rec. ITU-T J.244: ITU-T Recommendation. – 2008. – 53 p.

293. *Mendel T.* Holistic View: The IT Management Software Market / T. Mendel, C. Townsend. – Forrester Research, Inc. 2008. – Jun. 19. – 13 p.

294. *Mendel T.* Market Overview: IT Financial Management Software / T. Mendel, P. O'Neill. – Forrester Research, Inc. 2009. – May 7. – 14 p.

295. *Mendel T.* What Vendor Strategists Need To Know About Lean – The Hot Topic For CIOs Right Now / T. Mendel, A. Peters, E. Radcliffe. – Forrester Research, Inc. 2009. – Oct. 29.

296. Method for objective measurements of perceived audio qualityю Rec. ITU-R BS.1387-1: Recommendation ITU-R. – 2001. – 100 p.

297. *Meyler K.* System Center Operations Manager 2007 Unleashed / K. Meyler, C. Fuller, J. Joyner, A. Dominey. – USA: Sams Publishing 2008. – 1416 p.

298. *Michalewich Z.* Genetic Algorithms + Data Structures = Evolution Programs / Z. Michalewich. – Berlin: Springer, 1996. – 387 p.

299. *Michalewich Z.* How to Solve It: Modern Heuristics / Z. Michalewich, D. Fogel. – Berlin: Springer, 2002. – 483 p.

300. *Mordal-Manet K.* An Empirical Model for Continuous and Weighted Metric Aggregation / K. Mordal-Manet, J. Laval, S. Ducasse, N. Anquetil, F. Balmas // 15th

European Conf. on Software Maintenance and Reengineering. 1–4 March 2011. – Oldenburg. Germany, 2011. – P. 141–150.

301. *Mosler K.* Computing zonoid trimmed regions of dimension d>2 / K. Mosler, T. Lange, P. Bazovkin // Computational Statistics and Data Analysis. – 2009. – Vol. 53, Issue 7. – P. 2500–2510.

302. *Mosler K.* Multivariate dispersion, central regions and depth / K. Mosler. – New York.: Springer, 2002. – 291 p.

303. *Moy J.* OSPF specification/ J. Moy. – RFC1131. – October, 1989. – 107 p.

304. *Naur P.* Revised report on the algorithmic language ALGOL 60// P. Naur // Communications of the ACM 6. –1 Jan. 1963. – P. 1–17.

305. *NIQA* – Non-Instrisuve voice Quality Analyzer. Режим доступа: http://www.sevana.fi/non-intrusive-voice-quality-testing-software.php

306. Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference. Rec. ITU-R BT.1683: Recommendation ITU-R (Question ITU-R 44/6). – 2004. – 107 p.

307. *O'Donnell G.* The state of it service management in 2011 / G. O'Donnell, D. Washburn, S. Mann, J. Rakowski, E. Hubbert. – Forrester Research, Inc. – 2011. – Nov. 4.

308. One-way transmission time. Rec. ITU-T G.114: ITU-T Recommendation. – 2003. – 12 p.

309. *Palagin A.V.* Design and application of the PLD-based reconfigurable devices / A.V. Palagin, V.N. Opanasenko // Design of Digital Systems and Devices. – Berlin, Heidelberg: Springer International Publishing, 2011. – Vol. 79. – P. 59–91.

310. *Palagin A.V.* Reconfigurable computing technology / A.V. Palagin, V.N. Opanasenko // Cybernetics and Systems Analysis. Springer New York, 2007. – Vol. 43, № 5. – P. 675–686.

311. *Palagin A.V.* Resource and Energy Optimization Oriented Development of FPGA-Based Adaptive Logical Networks for Classification Problem / A.V. Palagin, V.N. Opanasenko, S.L. Kryvyi // Green IT Engineering: Components, Networks and Systems Implementation. – Springer, 2016. – P. 195–218.

312. *Paxson V.* Framework for IP performance metrics / V. Paxson, G. Almes, J. Mahdavi, M. Mathis. – RFC 2330. – May, 1998.

313. *Paxson V.* Wide-Area Traffic: The Failure of Poisson Modeling / V. Paxson, S. Floyd // IEEE/ACM Trans. on Networking. – 1995, June. – Vol. 3, N. 3. – P. 226–224.

314. *Pelikan M.* Combining the strengths of Bayesian optimization algorithm and adaptive evolution strategies / M. Pelikan, D.E. Goldberg, S. Tsutsui // Genetic and Evolutionary Computation Conf.: (GECCO-2002), New York, USA, July 9–13, 2002. – P. 512–519.

315. *Perry E.* Network Management for Residential Broadband Interactive Data Services / E. Perry, S. Ramanathan // IEEE Communications Magazine. – 1996. – Nov. – P. 114–121.

316. *Perrow C.* Complex organizations: a critical essay / C. Perrow. – New York: McGraw-Hill, 1986. – 307 p.

АУСКИ

317. *Pictet O.V.* Genetic algorithms with collective sharing for robust optimization in financial applications/ O.V. Pictet, M.M. Dacorogna, R.D. Dave, B. Chopardy, R. Schirru, M. Tomassini// Neural Network World, 5(4). – 1996. – P. 573–587.

318. Planning to Implement Service Management (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. – 6th impression. – 2005. – 208 p.

319. *Pogorilyy S.* Computer Network Performance Optimization using Genetic Algorithms / S. Pogorilyy, R. Bilous // 3rd Int. Conf. on Control and Optimization with Industrial Applications COIA 22-24 August 2011. – Bilkent University, Ankara, Turkey. – 2011. – P. 19–20.

320. *Postel J.* Internet Control Message Protocol / J. Postel. – RFC 792. – 5, September, 1981.

321. Principles for a telecommunications management network: ITU-T Rec. M.3010. – Geneva. 2000. – 36 p.

322. *Raman R.* Matchmaking: An extensible framework for distributed resource management / R. Raman, M. Livny, M. Solomon // Cluster Computing. – 1999. – №2(2). – P. 311–321.

323. Relative timing of sound and vision for broadcasting. Rec. ITU-R BT.1359: Recommendation ITU-R (Question ITU-R 35-4/11). – 1998. – 3 p.

324. Release Notes for QoS Policy Manager 1.1/ Cisco Systems, Inc. –1999. – 8 p.

325. *Rolik O.* Decomposition-compensation Approach to Microcloud-based IoT Infrastructure Management / O. Rolik, S. Telenyk, E. Zharikov, M. Yasochka // Proc. of 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT) 12-14 Dec. 2016. – Reston, USA – P. 603–608.

326. *Rolik O.* Decomposition-Compensation Method for IT Service Management / O. Rolik, V. Kolesnik, D. Halushko // Information Technology and Computational Physics. CITCEP 2016. Advances in Intelligent Systems and Computing, vol 462, Springer, Cham. – 2017. – P. 89–107.

327. *Rolik O.* Decomposition-compensation method of service level management in corporate IT infrastructures with the use of adaptive genetic algorithm / O. Rolik, V. Kolesnik and D. Halushko// Proc. of the International Conference Radio Electronics & Info Communications (UkrMiCo). – Kiev, Ukraine. – 2016. – P. 1–5.

328. *Rolik O.* IoT and Cloud Computing: The Architecture of Microcloud-Based IoT Infrastructure Management System / O. Rolik, S. Telenyk, E. Zharikov // Emerging Trends and Applications of the Internet of Things, Hershey, PA, USA: IGI Global, (Book Chapter). – 2017. – P. 198–234.

329. *Rolik O.* Microcloud-based architecture of management system for IoT infrastructures / O. Rolik; E. Zharikov, S. Telenyk // Proc. 3rd International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), October 4-6, 2016, Kharkiv, Ukraine. – IEEE, 2016. – P. 149–151.

330. *Rolik O.* Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // Information Technology, Computational and Experimental Physics. Kulczycki P., Kowalski P. A., Łukasik S. (eds). – AGN-UST. – 2016. – P. 13–16.

331. *Rolik O.* Service Quality Management in Microcloud-Based IoT Infrastructure / O. Rolik, S. Telenyk, E. Zharikov / Technical Transactions. Electrical Engineering. – Politechnika Krakowska. – 2016 (113). – Issue 2-E (12). – P. 231–244.

332. *Rudolph G.* Convergence analysis of canonical genetic algorithms / G. Rudolph // IEEE Trans. Neural Networks. – 1994. – №5. – P. 96–101.

333. Rule-based Algorithmic Approach for Solving Problems of Impact Analysis in Access Networks / O. Rolik, E. Zharikov, V. Kolesnik, M. Yasochka // Proc. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), May 29 – June 2, Ukraine Kyiv, Ukraine. – 2017. – P. 1161–1166.

334. *Scherbaum T.* Praxisbuch Nagios / T. Scherbaum. – O'Reilly. – 2009. – 272 p.

335. *Schulz G.* The Green and Virtual Data Center/ Schulz G. – CRC Press – Taylor & Francis Group, 2009. – 376 p.

336. Security Management (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. – 4th impression. – 2006. – 158 p.

337. *Serpanos D.N*. MMPacking: A load and storage balancing algorithm for distributed multimedia servers / D.N. Serpanos, L. Georgiadis, T. Bouloutas// IEEE Transactions on Circuits and Systems for Video Technology, 8(1): Feb. 1998. – P. 13–17.

338. Service Delivery (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. – 10th impression. – 2005. – 382 p.

339. Service Support (IT Infrastructure Library)/ Office of Government Commerce (OGC). Stationery Office. – 12th impression. – 2005. – 312 p.

340. *Setzer T.* Decision support for service transition management enforce change scheduling by performing change risk and business impact analysis / T. Setzer, K. Bhattacharya H. Ludwig // 11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008). – Vol. 11, №1, April. 2008. – 2008. – P. 200–207.

341. *Shachnai H.* On two class-constrained versions of the multiple knapsack problem / H. Shachnai, T. Tamir// Algorithmica, № 29. – 2001. – P. 442–467.

342. *Shankar S.* An Automated System for Analyzing Impact of faults in IP Telephony Networks / S. Shankar, O. Satyararayaman //10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006). – Apr. 2006. – P.1–4.

343. *Sharfman I.* A geometric approach to monitoring threshold functions over distributed data streams / I. Sharfman, A. Schuster, D. Keren // Proceedings of the 2006 ACM SIGMOD international conference on Management of data table of contents. – Chicago, IL, USA. – 2006. – P. 301–312.

344. *Shaviner G.* Highly available monitoring system architecture (HAMSA)// Institute of Computer Science, The Hebrew University of Jerusalem. – 2003. – 68 p.

345. *Shaw M.* Larger Scale Systems Require Higher-Level Abstractions / M. Shaw // SIGSOFT Engineering Notes. – 1989 – Vol.14 (3). – P. 143–146.

346. *Simon H.* The Sciences of the Artificial / H. Simon. – Cambridge, MA: The MIT Press, 1982. – 218 p.

347. SLA Management Handbook. Vol. 1. Executive Overview. Ver. 2.1. TeleManagement Forum. – 2005. – 39 p.

348. SLA Management Handbook. Vol. 2. Concepts and Principles. Release 2.5. – TeleManagement Forum. – 2005. – 218 p.

349. SLA Management Handbook. Vol. 3. Service and Technology Examples. Ver. 2.1. – TeleManagement Forum. – 2005. – 89 p.

350. Software engineering – product quality. Part 1: Quality model: ISO/IEC 9126-1. – ISO/IEC, 2001.

351. *Snevely R.* Enterprise Data Center Design and Methodology / R. Snevely. – Sun Microsystems Press, 2006. – 220 p.

352. *Spears W.M.* Adapting Crossover in Evolutionary Algorithms / W.M. Spears // 4th Annual Conf. on Evolutionary Programming: San Diego, California, March 1–3, 1995. – P. 367–384.

353. *Stahel W.A.* Breakdown of covariance estimators / W.A. Stahel // Research Report 31, Fachgruppe fuer Statistik, ETH, Zurich. – 1981, Dec.

354. *Stallings W.* SNMP, SNMPv2, SNMPv3, RMON1 and 2 / W. Stallings. – 3rd edition. AdisonWesley, 1998. – 640 p.

355. *Staten J.* Deliver Cloud Benefits Inside Your Walls / J. Staten, S. Yates, J. Rymer, F. Gillett, L.E. Nelson. – Forrester Research, Inc. 2009. – Apr. 13. – 14 p.

356. System Center Configuration Manager (SCCM) 2007 Unleashed / K. Meyler, B. Holt, G. Ramsey. – Sams Publishing, 2009. – 1224 p.

357. System Center Operations Manager 2007 Unleashed / K. Meyler, C. Fuller, J. Joyner, A. Dominey. – Sams Publishing, 2008. – 1416 p.

358. System Center Service Manager 2010 Unleashed / K. Meyler, A. Verkinderen, A. Bengtsson, P. Sundqvist, D. Pultorak. – Sams Publishing, 2011. – 768 p.

359. *Sugeno M.* An introductory survey of fuzzy control // Information Sciences. – 1985. №36 – P. 59–83.

360. *Surhone L.M.* HP Operations Manager / L.M. Surhone, M.T. Tennoe, S.F. Henssonow. – βetascript Publishing, 2011. – 88 p.

361. Technical Specification ETSI TS 102 034. Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks. V1.4.1. – ETSI. – 2009. – 229 p.

362. *Telenyk S.* An approach to software defined cloud infrastructure management / S. Telenyk, E. Zharikov, O. Rolik // Proc. of the XI Int. Scient. and Technical Conf. «Computer Science and Information Technologies Congress on Information Technology» (CSIT 2016) 6–10 September, Lviv, Ukraine. – 2016. – P. 21–26.

363. *Telenyk S.* An approach to virtual machine placement in cloud data centers / S. Telenyk, E. Zharikov, O. Rolik // Proc. of International Conference Radio Electronics & Info Communications (UkrMiCo). – Kiev, Ukraine. – 2016. – P. 1–6.

364. *Telenyk S.* Architecture and Conceptual Bases of Cloud IT Infrastructure Management / S. Telenyk, E. Zharikov, O. Rolik // Advances in Intelligent Systems and Computing. – Springer, 2017. – P. 41–62.

365. *Telenyk S.* Consolidation of Virtual Machines Using Simulated Annealing Algorithm / S. Telenyk, E. Zharikov, O. Rolik // Proc. of the XII Int. Scientific and Technical Conf. «Computer Science and Information Technologies» (CSIT 2017) 5–8 September, Lviv, Ukraine. – 2017. – P. 117–121.

366. *Telenyk S.* Consolidation of Virtual Machines Using Stochastic Local Search / S. Telenyk, E. Zharikov, O. Rolik // Advances in Intelligent Systems and Computing II, Selected Papers from the Int. Conf. on Computer Science and Information Technologies, CSIT 2017, September 5-8 Lviv, Ukraine, Springer. – 2017. – P. 523–537.

367. *Telenyk S.* Models and methods of resource management for VPS hosting / S. Telenyk, O. Rolik, M. Bukasov, D. Halushko // Technical transaction. Automatic control. – Politechnica Krakowska, 2013. – Vol. 4-AC. – P. 41–52.

368. *Telenyk S.* Resource allocation and load management considering the assessed quality of the provided service and the use of agent technologies / S.F. Telenyk, A.I. Rolik, A.A. Pokotylo // 23rd International Crimean Conference «Microwave & Telecommunication Technology» (CriMiCo'2013). 9–13 September, Sevastopol, Crimea, Ukraine. – 2013. – P. 535–536.

369. *Telenyk S.* Qualitative evaluation method of IT-infrastructure elements functioning/ S. Telenyk, Y. Dorogiy, O. Rolick, D. Halushko, M. Bukasov, A. Pysarenko // Proc. of 2014 IEEE Int. Black Sea Conf. on Communications and Networking (BlackSeaCom 2014), Chisinau, Moldova, May 27–30, 2014. – P.165–169.

370. The E-model: a computational model for use in transmission planning. Rec. ITU-T G.107: ITU-T Recommendation. – 2011. – 18 p.

371. The NIST Definition of Cloud Computing. Rec. of the National Institute of Standards and Technology Special Publication 800-145 / NIST, 2011. – 7 p.

372. *Thierens D.* Adaptive mutation rate control schemes in genetic algorithms / D. Thierens // Congress on Evolutionary Computation: CEC'02, Honolulu, Hawaii, May 12–17, 2002. – P. 980–985.

373. *Tukey J.W.* Mathematics and the Picturing of Data / J.W. Tukey // Proceedings of the International Congress of Mathematicians, Vancouver. – 1974.

374. *Urgaonkar B.* Dynamic Resource Management in Internet hosting Platforms. / Ph.D., University of Massachusetts Amherst, 2005. – 119 p.

375. *Verma D.* Policy based management of content distribution networks / D. Verma, S. Calo, K. Amiri // IEEE Network. – 2002. – Vol. 16, №2. – P. 34–39.

376. *Vilemaitis M.* HP Network Node Manager 9: Getting Started/ M. Vilemaitis. – UK: Packt Publishing Ltd. – 2011. – 562 p.

377. *Vinoski S.* Service Discovery 101/ S. Vinoski// IEEE. Internet Computing. – Vol. 7, N 1. – 2003. – P. 69–71.

378. *Wang N.* An overview of routing optimization for internet traffic engineering/ N. Wang, K. Hon Ho, G. Pavlou, M. Howarth // IEEE Communications Surveys & Tutorials. – 2008. – Vol. 10, N 1. – P. 36–56.

379. *Womack J.P.* Lean Thinking: Banish Waste and Create Wealth in Your Corporation, Revised and Updated / J.P. Womack, D.T. Jones. – Second Edition. – New York: Simon & Schuster, Inc. 2003. – 397 p.

380. *Worthen B.* IT Governance: ITIL Power / B. Worthen // CIO. – 2005. – Sept. 1. – 7 p.

381. *Wuhib F.* Decentralized computation of threshold crossing alerts / F. Wuhib, M. Dam, R. Stadler, A. Clemm // Proc. 16th IEEE/IFIP International Workshop on Distributed Systems. – Barcelona, Spain. – 2005. – Vol. 3775. – P. 220–232.

382. *Wuhib F.* Decentralized service-level monitoring using network threshold crossing alerts /F. Wuhib, R. Stadler, A. Clemm // IEEE Communications Magazine. – 2006. – Vol. 44, №10. – P. 70–76.

383. *Wolf J.L.* Disk load balancing for video-on-demand systems / J.L. Wolf, P.S. Yu, H. Shachinai // ACM. Springer Multimedia Systems Journal, 5(6), 1997. – P. 358–370.

384. *Wolf J.L.* On Balancing the Load in a Clustered Web Farm / J.L. Wolf, P.S. Yu, T.J. Watson // Proc. ACM Transactions on Internal Technology, Vol. 1, N 2, Nov. 2001. – P. 231–261.

385. *Xiao J.* Assessing network service profitability: Modeling from market science perspective/ J. Xiao and R Boutaba// IEEE/ACM Transactions on Networking, Vol. 15, N 6, December 2007. – P. 1307–1320.

386. *Xiao J.* Customer-centric network upgrade strategy: Maximizing investment benefits for enhanced service quality / J. Xiao, R. Boutaba // IEEE/IFIP Network Operations and Management Symposium, (NOMS'04), 2004. N 1, Apr. – P. 759–777.

387. *Xie G.G.* An efficient network architecture motivated by application-level QoS / G.G. Xie, S.S. Lam // High Speed Networking. – 1998. – Vol. 6 N 3. – P. 165–179.

388. *Yau D.K.Y.* Migrating sockets – End system support for networking with quality of service guarantees / D.K.Y. Yau, S.S. Lam // IEEE ACM Transactions on Networking. – 1998. – Vol. 6, №6. – P. 700–716.

389. *Zabbix* 1.8 Network Monitoring / R. Olups. – Packt Publishing, 2010. – 428 p.

390. *Zadeh L.A.* Fuzzy Sets // Information and Control. – 1965. – №8. – P. 338–353.

391. *Zhang L.* SLA based profit optimization in autonomic computing systems / L. Zhang, D. Ardagna // Proc. 2nd Int. Conf. on Service Oriented Computing (15 Nov. 2004). – ACM. – 2004. – P. 173–182.

392. *Zhen L.* Optimal control of resource allocation in e-business environments with strict quality-of-service performance guarantees / L. Zhen, M.S. Squillante, J.L. Wolf // Proc. of the 41st IEEE Conference on Decision and Control, 2002. – Vol. 4. – P. 4431– 4439.

393. *Zuo Y.* General notions of statistical depth functions / Y. Zuo, R. Serfling // Ann. Statist. – 2000. – Vol. 28, N 2. – P. 461–482.

ОГЛАВЛЕНИЕ

Введение 3

Глава 1. **Анализ проблем, тенденций и перспектив развития управления корпоративной ИТ-инфраструктурой** 8

1.1. Корпоративная ИТ-инфраструктура как объект управления 8
1.2. Анализ тенденций развития управления информационными технологиями и ИТ-инфраструктурой 38
1.3. Анализ особенностей управления ИТ-инфраструктурой 45
1.4. Постановка задач управления корпоративной ИТ-инфраструктурой 70
Выводы к главе 1 76

Глава 2. **Концепция управления корпоративной ИТ-инфраструктурой** 78

2.1. Суть предлагаемой концепции 78
2.2. Развитие парадигмы ITMS2 84
2.3. Методика управления уровнем ИТ-услуг на основе сервисно-ресурсных моделей 95
Выводы к главе 2 103

Глава 3. **Управление уровнем услуг в корпоративной ИТ-инфраструктуре** 105

3.1. Организация управления уровнем ИТ-услуг 105
3.2. Представление СУИ в виде двухуровневой системы с координатором 116
Выводы к главе 3 128

Глава 4. **Методы мониторинга элементов ИТ-инфраструктуры** ... 129

4.1. Реальные и логические объекты управления в ИТ-инфраструктуре 129
4.2. Методы мониторинга ИТ-инфраструктуры 145
Выводы к главе 4 156

Глава 5. **Методы анализа состояния и сведения метрик качества функционирования элементов ИТ-инфраструктуры** 157

5.1. Методы анализа состояния элементов ИТ-инфраструктуры 157
5.2. Принципы построения и использования метрик при управлении ИТ-инфраструктурой 160
5.3. Кодирование состояния объектов мониторинга и управления 169

АЛТГТУ

5.4. Аналитический метод оценки состояния и качества функционирования элементов ИТ-инфраструктуры ... 176
5.5. Сведение метрик оценивания уровня обслуживания пользователей на основе экспертных оценок ... 182
5.6. Нечеткое оценивание в задачах управления уровнем обслуживания ... 197
5.7. Метод оценивания состояния элементов ИТ-инфраструктуры с использованием нейронных сетей ... 213
5.8. Метод распознавания образов на основе потенциальных функций в задачах оценки уровня мультимедийных услуг ... 217
5.9. Сведение метрик качества функционирования элементов ИТ-инфраструктуры с помощью аппарата непараметрической статистики ... 224
Выводы к главе 5 ... 235

Глава 6. Методы оценки влияния неисправностей на качество предоставления ИТ-услуг ... 237

6.1. Применение глубинно-упорядоченных регионов для выявления неисправных элементов ИТ-инфраструктуры ... 237
6.2. Алгебраический подход к определению распространения влияния неисправностей в сети доступа на качество предоставляемых сервисов ... 248
6.3. Обобщенный подход к оценке влияния неисправностей в телекоммуникационной сети на уровень услуг ... 259
6.4. Импакт-анализ в IP-сетях с протоколами RIP и OSPF ... 291
Выводы к главе 6 ... 303

Глава 7. Управление ресурсами и нагрузкой корпоративной ИТ-инфраструктуры ... 304

7.1. Процессно-ресурсный подход к управлению распределением ресурсов ... 304
7.2. Модели и методы управления распределением ограниченных ресурсов корпоративной ИТ-инфраструктуры ... 316
Выводы к главе 7 ... 332

Глава 8. Управление ресурсами и нагрузкой центров обработки данных ... 333

8.1. Анализ проблем управления ресурсами и нагрузкой ЦОД при виртуальном хостинге ... 333
8.2. Управление нагрузкой и ресурсами ЦОД при виртуальном хостинге ... 351
8.3. Управление нагрузкой и ресурсами ЦОД при выделенных серверах ... 359
8.4. Модели и методы распределения ресурсов в системах с серверной виртуализацией с учетом важности приложений ... 371
8.5. Модели управления виртуальными машинами при серверной виртуализации и живой миграции виртуальных машин ... 382
8.6. Метод управления резервированием ресурсов ЦОД для поддержания заданного качества обслуживания запросов пользователей ... 389
Выводы к главе 8 ... 397

АВТОРСКИЙ

Глава 9. **Модели и методы управления информационными потоками в корпоративной ИТ-инфраструктуре 398**

9.1. Анализ проблем управления сетевым трафиком 398
9.2. Метод управления сетевым трафиком в корпоративной IP-сети 409
9.3. Исследование моделей управления потоками данных в корпоративных IP-сетях 418
9.4. Модель подсистемы управления информационными потоками с адаптивным регулятором 426
Выводы к главе 9 432

Глава 10. **Методы и алгоритмы решения задач управления распределением ресурсов и нагрузкой ИТ-инфраструктуры .. 433**

10.1. Математическая модель распределения виртуальных машин по серверам 433
10.2. Методы решения задач управления нагрузкой и распределением ограниченных ресурсов 436
10.3. Эвристические алгоритмы распределения ресурсов ИТ-инфраструктуры 438
10.4. Решение задач оптимального распределения нагрузок и ресурсов с использованием классического генетического алгоритма 440
10.5. Управляемый генетический алгоритм в задачах распределения ресурсов 444
10.6. Адаптивный генетический алгоритм для решения задач распределения ресурсов ЦОД 456
10.7. Алгоритм последовательного анализа вариантов в задаче распределения виртуальных машин 476
Выводы к главе 10 481

Глава 11. **Разработка и реализация информационной технологии и системы управления ИТ-инфраструктурой 483**

11.1. Концептуальные и методические особенности проектирования и внедрения системы управления ИТ-инфраструктурой 483
11.2. Структура и принципы функционирования ИТ управления ИТ-инфраструктурой 491
11.3. Разработка системы управления ИТ-инфраструктурой SmartBase.ITS.Control 496
11.4. Технологические особенности реализации ИТ управления ИТ-инфраструктурой и системы управления ИТ-инфраструктурой 510
11.5. Интерфейс администратора 529
Выводы к главе 11 545

СПИСОК УСЛОВНЫХ СОКРАЩЕНИЙ **546**

СПИСОК ЛИТЕРАТУРЫ **548**

АЛТСКИЙ

Наукове видання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

РОЛІК Олександр Іванович
ТЕЛЕНИК Сергій Федорович
ЯСОЧКА Максим Володимирович

Управління корпоративною IT-інфраструктурою

(Російською мовою)

Київ, Науково-виробниче підприємство
«Видавництво “Наукова думка” НАН України», 2018

Підп. до друку ??.??.2018. Формат 70×100/16. Папір офс. №1.
Гарн. Таймс. Друк. офс. Ум. друк. арк. 36,0.
Обл.-вид. арк. 53,7. Тираж 300 прим. Зам. № .

НВП «Видавництво “Наукова думка” України»
Свідоцтво про внесення до Державного реєстру
суб’єкта видавничої справи
ДК № 2440 від 15.03.2006 р.
01601, Київ 1, вул. Терещінковська, 3

АУТСКП