

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНСТИТУТ МОДЕРНІЗАЦІЇ ЗМІСТУ ОСВІТИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»**

**WINTER INFOCOM
ADVANCED
SOLUTIONS 2017**

МАТЕРІАЛИ

**V МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ
КОНФЕРЕНЦІЇ**

CONFERENCE PROCEEDINGS

5th SCIENTIFIC AND PRACTICAL CONFERENCE

КИЇВ, УКРАЇНА

1-2 ГРУДНЯ 2017 РОКУ

УДК 004

Редакційна колегія:

Бідюк П.І., д.т.н., проф., ІПСА, КПІ ім. Ігоря Сікорського, Україна, Київ

Павлов О.А., д.т.н., проф., КПІ ім. Ігоря Сікорського, Україна, Київ

Теленик С.Ф., д.т.н., проф., КПІ ім. Ігоря Сікорського, Україна, Київ

Грішин І.Ю., д.т.н., проф., Кубанський державний технологічний університет, Російська Федерація

Головний редактор:

Писаренко А.В., к.т.н., доц., КПІ ім. Ігоря Сікорського, Україна, Київ

Winter InfoCom 2017: Матеріали V Міжнародної науково-практичної конференції з інформаційних систем та технологій, м. Київ, 1-2 грудня 2017 р. – К.: Вид-во ТОВ "Інжиніринг", 2017. – 76 с. – Мови укр., рос., англ.

Конференція входить до Переліку міжнародних та всеукраїнських науково-практичних конференцій здобувачів вищої освіти та молодих учених у 2017 році (додаток до листа Міністерства освіти і науки України № 1/9-24 від 23 січня 2017 року).

Проведення конференції регламентоване наказом ректора КПІ ім. Ігоря Сікорського № 3-469 від 06 листопада 2017 р.

Усі права застережено. Передруки та переклади дозволяються лише за згодою автора та редакції. За достовірність фактів, цитат, назв та іншої інформації несуть відповідальність автори.

Редакційна колегія дотримується прийнятих міжнародною спільнотою принципів публікаційної етики, відображених, зокрема, в рекомендаціях Комітету з етики наукових публікацій (Committee on Publication Ethics, COPE), а також враховує досвід авторитетних міжнародних видавництв. Щоб уникнути недобросовісної практики в публікаційній діяльності (плагіат, виклад недостовірних відомостей та ін.), з метою забезпечення високої якості наукових публікацій, визнання громадськістю отриманих автором наукових результатів, кожен член редакційної колегії, автор, рецензент, видавець, а також установи, які беруть участь в видавничому процесі, зобов'язані дотримуватися етичних стандартів, норм і правил та вживати всіх можливих заходів для запобігання їх порушень. Дотримання правил етики наукових публікацій усіма учасниками цього процесу сприяє забезпеченню прав авторів на інтелектуальну власність, підвищенню якості видання і виключення можливості неправомірного використання авторських матеріалів в інтересах окремих осіб.

ISBN 978-966-2344-58-5

ПРОГРАМА КОНФЕРЕНЦІЇ

ПРОГРАМА / PROGRAM

Інформаційні системи та технології

Карымсакова И.Б.
Денисова Н.Ф.
Крак Ю.В. Разработка роботизированной системы для плазменного напыления имплантов

Дорошенко А.Ю.
Туманов В.В. Модуль калібрування та позиціонування системи комп'ютерного зору для цифрової нарізки матеріалів

Коноваленко А. Система для організації інтерактивних квест-ігор побудована на Bluetooth-маячках

Системи керування

Юрчук Л.Ю.
Бублінський С.М. Проектування відеокадрів людино-машинного інтерфейсу АСУ ТП

Лапханов Э.А. Оценка возможности создания дополнительной тяги для управления космическими аппаратами на основе использования постоянных магнитов

Технології програмування

Ашур И.З.
Дорошенко А.Ю. Высокопроизводительное производство матриц посредством Android NDK и JNI

Оброблення інформації у складних системах

Дмитренко О.О.
Ланде Д.В. Метод накопичувального впливу для аналізу когнітивних карт

Підготовка кадрів у галузі інформаційних технологій

Стенин А.А.
Пасько В.П.
Лемешко В.А.
Шитикова И.Г. Оптимизация учебных планов IT-специальностей

1 грудня

Інформаційні системи та технології

Стенин А.А.
Пасько В.П.
Шитикова И.Г. Аналіз і оптимізація автономних систем
 теплоснабження

Пирожков О.Ю.
Савчук О.В. Безпека даних в хмарних середовищах

Рижко Б.В.
Смолинець О.Т. Розробка архітектури системи
 автоматизованого збору, обробки та аналізу
 даних на основі технології Big Data

Сергієнко А.М.
Орлова М.М.
Молчанов О.А. Мікроконтролер для керування послідовними
 портами вводу-виводу

Розподілені та паралельні обчислення

Жабин В.
Жабина В. Ускорення обміну даними в
 мультипроцесорних системах

Loutskii H.
Volokyta A.
Rehida P. Isoefficiency control tool for distributed
 computing

Прихідько В.С.
Дорошенко А.Ю.
Яценко О.А. Розробка паралельного алгоритму кластеризації
 даних засобами автоматизації проектування

ЗМІСТ/CONTENTS

<i>Інформаційні системи та технології/Information Systems and Technologies.....</i>	9
Карымсакова И.Б., Денисова Н.Ф., Крак Ю.В.	
Разработка роботизированной системы для плазменного напыления имплантов.....	11
Дорошенко А.Ю., Туманов В.В.	
Модуль калібрування та позиціонування системи комп'ютерного зору для цифрової нарізки матеріалів.....	14
Коноваленко А.	
Система для організації інтерактивних квест-ігор побудована на Bluetooth-маячках.....	17
Сергієнко А.М., Орлова М.М., Молчанов О.А.	
Мікроконтролер для керування послідовними портами вводу-виводу.....	19
Рижко Б.В., Смолинець О.Т.	
Розробка архітектури системи автоматизованого збору, обробки та аналізу даних на основі технології Big Data.....	21
Стенин А.А., Пасько В.П., Шитикова И.Г.	
Анализ и оптимизация автономных систем теплоснабжения.....	24
Пирожков О.Ю., Савчук О.В.	
Безпека даних в хмарних середовищах.....	28
<i>Системи керування/Control Systems.....</i>	31
Лапханов Э.А.	
Оценка возможности создания дополнительной тяги для управления космическими аппаратами на основе использования постоянных магнитов.....	33
Юрчук А.Ю., Бублінський С.М.	
Проектування відеокадрів людино-машинного інтерфейсу АСУ ТП...	35
<i>Технології програмування/Programming technologies.....</i>	39
Ашур И.З., Дорошенко А.Ю.	
Высокопроизводительное производство матриц посредством Android NDK и JNI.....	41
<i>Оброблення інформації у складних системах/Information processing in complex systems.....</i>	45
Дмитренко О.О., Ланде Д.В.	
Метод накопичувального впливу для аналізу когнітивних карт.....	47

<i>Розподілені та паралельні обчислення/ Distributed and parallel computing</i>	51
Жабин В., Жабина В.	
Ускорение обмена данными в мультипроцессорных системах.....	53
Прихідько В.С., Дорошенко А.Ю., Яценко О.А.	
Розробка паралельного алгоритму кластеризації даних засобами автоматизації проектування.....	57
Loutskii H., Volokyta A., Rehida P.	
Isoefficiency control tool for distributed computing.....	59
<i>Підготовка кадрів у галузі інформаційних технологій/ Training in information technologies field</i>	61
Стенин А.А., Пасько В.П., Лемешко В.А., Шитикова И.Г.	
Оптимизация учебных планов IT-специальностей.....	63
<i>Abstracts</i>	67

**ІНФОРМАЦІЙНІ СИСТЕМИ ТА
ТЕХНОЛОГІЇ**

**INFORMATION SYSTEMS AND
TECHNOLOGIES**

Разработка роботизированной системы для плазменного напыления имплантов

Карымсакова И.Б., Денисова Н.Ф.

Восточно-Казахстанский государственный
технический университет им. Д. Серикбаева
Усть-Каменогорск, Казахстан
indviki@mail.ru, ndenisova@ektu.kz

Крак Ю.В.

Киевский национальный университет имени Тараса
Шевченко
Киев, Украина
Yuri.krak@gmail.com, Krak@univ.kiev.ua

Аннотация. В работе исследуются характеристики имплантов по физическим размерам, типу протезирования различных органов, материалах изготовления, фирмах-изготовителях и др. Рассматривается проблема нанесения на импланты биосовместимых покрытий при помощи роботизированного комплекса на базе промышленного робота. Для нанесения покрытия описывается технология плазменного напыления и соответствующее оборудование роботизированного комплекса и были отобраны импланты, которые соответствуют возможностям установки плазменного напыления биосовместимых покрытий.

Ключевые слова: имплант, плазменное напыление, роботизированные установки.

ВВЕДЕНИЕ

Усовершенствование методов построения систем создания имплантов с использованием современных производственных решений является важной и актуальной проблемой. Оборудование и комплексы для плазменного напыления имплантов, с интегрированием промышленных роботов в процессы напыления, позволяют выйти на новый уровень производства, добиться существенного улучшения свойств поверхностей имплантов путем образования на рабочих поверхностях покрытий с особыми свойствами [1-3]. Роботизированные комплексы для процессов плазменного напыления позволяют существенно увеличить точность и качество напыления, оптимизировать технологии обработки и сократить производственный цикл. Роботы для плазменного напыления и программное обеспечение позволяют наносить функциональные покрытия с точностью до нескольких микрометров, что позволяет существенно сокращать расход материалов, осуществлять манипуляции, отображать технологические параметры и управлять ими в процессе напыления. Исследование и разработка имплантов, которые используются для замены поврежденных органов человека, с последующим напылением биосовместимых покрытий с использованием роботизированных комплексов является важной научно-технической проблемой, для решения которой необходимо разрабатывать новые методы и подходы.

Целью данной работы является исследование имплантов по различным критериям для последующей разработки роботизированного комплекса для плазменного напыления имплантов и

построения системы эффективного управления таким комплексом.

КРИТЕРИИ КЛАССИФИКАЦИИ ИМПЛАНТОВ

В соответствии со скелетным и физиологическим строением человека [4] будем исследовать основные типы имплантов в зависимости от органов человека к которым эти импланты применяются.

С этой целью импланты будем классифицировать по критерию замены или вживления различных человеческих органов [4]: зубопротезный имплант; слуховой (кохлеарный) имплант; стволотомозговой имплант; имплант клапанов сердца; имплант сетчатки; имплант для конечностей.

Анализ геометро-топологических характеристик данных имплантов показывает, что они являются очень сложными поверхностями, для получения и обработки которых необходимо использовать точное оборудование, в частности, роботизированные комплексы.

Отметим, что импланты подбираются для каждого человека отдельно, поэтому физические размеры имплантов являются параметрическими функциями, что позволяет изготавливать имплант путем подгонки параметров под конкретные размеры.

В результате проведенных исследований были проанализированы основные типы имплантов по критерию их физических геометрических размеров. Полученные результаты этих исследований представлены в виде базы данных [5].

Из анализа физических размеров имплантов следует, что некоторые импланты, в частности: зубные импланты, слуховые импланты, импланты сетчатки глаз и клапанов сердца, – имеют достаточно незначительные размеры (миллиметры и даже сотые части миллиметра), поэтому для решения проблемы нанесения напыления на такие изделия нужно разрабатывать специальное оборудование и использование промышленного робота для этих целей не представляется возможным.

Кроме данной классификации, важной есть классификация по критерию материала изготовления. В этом исследовании будем анализировать импланты из титана, поскольку этот металл по своим характеристикам наиболее подходит для создания имплантов. Также важным для практического использования имплантов есть их изготовление известными фирмами. Данные по

материалу использования и о фирмах-производителях имплантов собраны в виде базы данных [5].

ТЕХНОЛОГИЯ ПЛАЗМЕННОГО НАПЫЛЕНИЯ

Производство металлических имплантов состоит из этапа изготовления самого импланта и этапа обработки его поверхности. Причем обработка поверхности является важным этапом, определяющим биологическую толерантность (терпимость) имплантов.

Имплантаты из титана и кобальтохромового состава изготавливают способом литья, фрезерования и токарной обработки, порошковой металлургии, горячей штамповки, электрохимической обработки и штамповки с последующим фрезерованием.

Технологический процесс плазменного напыления в общем случае состоит из предварительной очистки поверхности, активационной обработки и непосредственно нанесения покрытия путем перемещения изделия относительно плазмотрона или наоборот.

Наведем некоторые технические характеристики данного процесса. Скорость перемещения 2...30 мм/сек, расстояние между плазмотроном и изделием 100...150 мм, диаметр пятна напыления 10...25 мм, толщина покрытия 0,05...1,0 мм. Температура нагрева деталей при плазменном напылении не превышает 100...150°C. Угол между осью потока частиц и восстанавливаемой поверхностью должен приближаться к 90°.

Плазмообразующим газом являются аргон или воздух. Расход аргона от 15 до 20 литров в минуту. В качестве порошкового материала, формирующего покрытие, используются различные материалы и сплавы, тугоплавкие соединения, оксиды, полимеры и их композиции размером частиц до 100 мкм.

С увеличением толщины покрытия величина остаточных растягивающих напряжений возрастает, возникает опасность разрушения слоя, поэтому следует стремиться к получению покрытия минимальной толщины с учетом допуска на возможный износ и припуск на обработку после напыления. Оптимальная толщина припуска составляет 0,15 мм, а для карбидных покрытий менее 0,1 мм. Минимальная толщина покрытий после оплавления может составлять 0,25 мм. Для покрытий равномерной толщины припуск на шлифование составляет от 0,1 до 0,4 мм в расчете на радиус. Усадка при оплавлении составляет 20%. Для валов на участках запрессовки следует напылять покрытия толщиной 0,13 мм независимо от диаметра.

Покрытие должно быть сплошным, однородного цвета, без частиц нерасплавленного металла, без трещин и отслоений. Шероховатость покрытия не более 80- 100 мкм.

Технологические режимы плазменного напыления определяются видом и дисперсностью материала, током плазменной струи и его напряжением, видом и расходом плазмообразующего газа, диаметром сопла

плазменной горелки и расстоянием от сопла до напыляемой поверхности [2, 6].

Приведем некоторые технологии, которые используются при плазменном напылении. Технология PVD (Physical Vapor Deposition) - это физическое осаждение паров титана на поверхность изделия. Высокая точность толщины покрытия, исключительная твердость, средний диапазон температур (400-600°C) означает, что эти покрытия могут быть применены к широкому спектру материалов.

Процедура нанесения PVD-покрытия состоит из следующего этапов. Изделие, на которое наносится PVD-покрытие, очищается. Процесс очистки меняется в зависимости от уровня качества поверхности, материала подложки и геометрии. Изделия загружаются в вакуумную камеру на специальные приспособления, предназначенные для оптимизации нагрузки камеры и обеспечения равномерного покрытия. Вакуумная камера откачивается до 10⁻⁶ мм. рт. ст. (высокий вакуум), чтобы удалить все загрязняющие вещества в системе. В камеру напускают инертный газ азот и подают напряжение на подложку, в результате образуется тлеющий разряд. Это чистка изделия в тлеющем разряде для начальной стадии осаждения металла. Большой ток и низкое напряжение дуги подается на мишень (твердый материал, используемый для нанесения). Металл испаряется и мгновенно ионы и основные свойства металла испарения (мишени) остаются неизменными в течение всего цикла осаждения металла. От изменения объема газа, типа газа во время реактивного осаждения изменяется структура покрытия керамики, карбидов, нитридов и оксидов.

Кроме того, поскольку эти покрытия обладают высокими медико-биологическими свойствами, то нитрид титана (TiN) широко используют при изготовлении имплантов и других хирургических инструментов, в силу того, что такое покрытие обладает высокой коррозионной стойкостью к окислению.

Отметим, что в качестве покрытия широко используется карбонитрид титана, который позволяет увечить в 2-3 раза[2] сроки использования, поскольку добавление углерода увеличивает твердость на 80%.

Таким образом, системы плазменного напыления классифицированы по критериям, которые характеризуют способ, тип напыления, расстояния до объекта, вспомогательное оборудование. Для практического использования полученных результатов была создана соответствующая база данных, в которую включена также информация о фирмах-производителях.

СОЗДАНИЕ РОБОТИЗИРОВАННОГО КОМПЛЕКСА ДЛЯ ПЛАЗМЕННОГО НАПЫЛЕНИЯ

Доступное оборудование, имеющееся на базе Восточно-Казахстанского государственного технического университета имени Д. Серикбаева включает промышленного робота фирмы Kawasaki RS-010L, с контроллером E40F-A001, компрессор с

давлением до 6 атмосфер, плазматрон с источником питания ДС120П.33, установку микроплазменного напыления МПН-004 разработанную в институте электросварки имени О.Е. Патона НАН Украины, с источником питания фирмы Fronius Magic Wave 2200 Job G/F и охладитель к нему, а так же фрезерно-гравировальный станок с контроллером RZNC-0501.

Манипуляционный робот представляет собой устройство, состоящее из подвижных частей с шестью степенями свободы для перемещения, установленного на нем оборудования по заранее заданному профилю, управляется посредством программируемого контроллера. На манипуляторе смонтировано устройство для плазменной обработки поверхности и для нанесения порошковых покрытий в потоке плазмы, в котором осуществляется генерация плазменного разряда и подача плазменного потока или порошка в плазменной струе в атмосфере аргона на подложку любой формы. Архитектура робота Kawasaki построена на AS языке, программное обеспечение (приложение) написано на языке C# для создания программ движения робота.

Основные характеристики робота Kawasaki RS-010L: количество степеней подвижности - 6; точность позиционирования - 0,06 мм; максимальная линейная скорость - 13100 мм/сек; зона досягаемости - 1925 мм; грузоподъемность - 10 кг.

Для изготовления осесимметричных имплантов будет использоваться токарно-фрезерный станок типа NXT 1000, а для изготовления имплантов любой сложности - фрезерный с ультразвуковой обработкой типа VU5 или ULTRASONIC 20. В комплексе эти два станка позволяют изготовить имплант любой сложности. Дополнительный комплект инструментов позволяет производить обработку титана и его сплавов.

Установка предназначена для плазменного напыления металлических и керамических поверхностей деталей и узлов, а также для восстановления изношенных поверхностей, к которым предъявляются повышенные эксплуатационные требования в отношении износостойкости, коррозионной стойкости, состоит из инверторного источника питания с газоподготовкой, питателя порошка и плазматрона. Технические характеристики плазменной установки: мощность плазматрона 2 кВт, производительность напыления от 2 до 3 кг/ч, рабочий ток плазматрона 60А, рабочее напряжение дуги 30 В.

К биосовместимым покрытиям имплантов предъявляются следующие требования: толщина покрытия должна быть в пределах от 300 мкм до 500 мкм, величина пор – от 150 мкм до 300 мкм, прочность сцепления покрытий с основой согласно стандарту

ISO 13779-2 – не менее 15 МПа, пористость – до 50%, шероховатость – от 150 мкм до 300 мкм. Параметры процесса напыления должны быть следующими: рабочий ток – 16 А; расход плазмообразующего газа – 140 л/ч; дистанция напыления – 40 мм; скорость подачи проволоки – 3 м/мин, диаметр пятна напыления – 5-10 мм [2].

Таким образом, созданный роботизированный комплекс позволяет решить проблему плазменного напыления имплантов.

ЗАКЛЮЧЕНИЕ

В силу использования промышленного робота с установкой плазменного напыления МПН-004, необходимо чтобы импланты имели достаточные физические размеры. Поэтому для проведения экспериментальных исследований были отобраны следующие импланты: имплант тазобедренного сустава; импланты коленного сустава; эндопротез плечевого сустава; имплант локтевого сустава; имплант голеностопного сустава; шеечный диафизарный имплант; имплант лучезапястного сустава; межфаланговый сустав пальцев кисти; пястно-фаланговый сустав пальцев кисти.

Дальнейшие исследования будут направлены на построения системы управления процессом плазменного напыления, построения траекторий движения промышленного робота и дополнительного оборудования для качественного напыления биосовместимых покрытий на предложенные импланты.

ЛИТЕРАТУРА

1. <http://irobs.ru/robotyi-dlya-napyleniya1> (дата обращения 25.10.2017).
2. Alontseva D. Modeling of Processes Taking Place during Powder Coating Treatment by an Electron Beam or a Plasma Jet / D. Alontseva, A. Krasavin, T. Kolesnikova, A. Russakova // *Acta Physica Polonica.* – 2014. – vol. 125, is. 6. – P. 1275-1279.
3. Достанко А.П. Плазменная металлизация в вакууме. / А.П. Достанко, С.В. Грушецкий, Л.И. Киселевский, М.И. Пикуль, В.Я. Ширипов. – М.: Наука и техника, 1983. – 201с.
4. Коробков А. В. Атлас по нормальной физиологии / А.В. Коробков, С.В. Чеснокова. – М.: Высшая школа, 1987. – 351 с.
5. Karymsakova I.B. Criteria for implants classification for coating implants using plasma spraying by robotic complex / I.B. Karymsakova, Iu.V. Krak, N.F. Denisova // *Eurasian Journal of Mathematical and Computer Applications.* – 2017. – vol. 5, is. 3. – P.44-52.
6. Пузряков А.Ф. Теоретические основы технологии плазменного напыления. / А.Ф. Пузряков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 360 с

Модуль калібрування та позиціонування системи комп'ютерного зору для цифрової нарізки матеріалів

Дорошенко А.Ю.

КПІ ім. Ігоря Сікорського
Київ, Україна
a-y-doroshenko@ukr.net

Туманов В.В.

КПІ ім. Ігоря Сікорського
Київ, Україна
tumanowlad@gmail.com

Анотація. Описано принцип роботи модуля калібрування та позиціонування для системи комп'ютерного зору, що застосовується в сфері цифрової нарізки матеріалів. Досліджено природу та вплив оптичних спотворень на фотознімки різальної поверхні і методи їх усунення. Також розроблено систему позиціонування, що гарантує високу точність визначення фізичного положення об'єктів за їх позицією на фотознімку.

Ключові слова: система комп'ютерного зору, цифрова нарізка, OpenCV, калібрування камери.

ВСТУП

На сучасному етапі розвитку технологій цифрової нарізки матеріалів, найбільш перспективним є застосування систем комп'ютерного зору для досягнення максимальної швидкодії та точності.

Попередні методи в більшості базувались на максимальному використанні локальної камери, розташованої безпосередньо біля різального механізму дуже близько до поверхні матеріалу для нарізки. Переміщаючи її вздовж контуру об'єкта що необхідно вирізати, можна було отримати дуже точну траєкторію подальшого ходу різця для акуратної нарізки. Але в даному випадку точність купується високими витратами часу та подвійним зношуванням механізмів станка, оскільки виконується подвійна робота: спочатку різець з камерою проходить по різальній поверхні щоб визначити положення об'єкта, а вже потім відбувається власне нарізка.

Новим підходом, який позбавлений недоліків попереднього є використання фотознімку різальної поверхні разом з об'єктами нарізки на ній, для отримання всіх необхідних даних для нарізки. Ідентифікацію та позиціонування об'єктів нарізки виконує спеціально розроблена для таких цілей система комп'ютерного зору. Проте в такого підходу також є одне слабе місце – точність. При аналізі зображення всієї різальної поверхні найголовнішою задачею є усунення оптичних спотворень та створення системи позиціонування, яка б дозволила визначити положення об'єктів на різальній поверхні з точністю ± 1 мм.

Метою даної статті є опис методу отримання точної системи позиціонування для цифрової нарізки що засновується на використанні фотознімку різальної поверхні.

ПОСТАНОВКА ЗАДАЧІ

Головна задача розроблюваного модулю калібрування та позиціонування в контексті всієї системи комп'ютерного зору для цифрової нарізки полягає в тому, щоб для будь-якої точки (пікселю) фотознімка робочої поверхні станка можна було однозначно встановити її координати в міліметрах відносно точки відліку локальної камери різця, а також відстань між будь-якими двома точками на фото, з точністю до 1 мм.

Для забезпечення технічної бази описуваної системи комп'ютерного зору необхідне встановлення додаткової глобальної фотокамери над різальною поверхнею станка для цифрової нарізки. З метою отримання оптимального для аналізу зображення, камеру слід розташувати над центром різальної поверхні, так щоб оптична вісь була перпендикулярна до неї. Ви-соту треба обрати достатню для отримання зображення всієї робочої поверхні станка з мінімумом зайвого на фото.

Реалізація основних алгоритмів обробки та аналізу растрових зображень вже є у відкритій бібліотеці OpenCV (англ. Open Source Computer Vision Library), яку може вільно використовувати академічних та комерційних цілях. Алгоритми цільової програми повністю базуються на використанні даної бібліотеки.

КАЛІБРУВАННЯ КАМЕРИ

Перед початком аналізу зображення різальної поверхні слід переконатись що воно повністю відображає реальність. Отримані за допомогою фотозйомки зображення часто «страждають» від різних оптичних спотворень (дисторсії) [2], вплив яких варіюється в залежності від якості оптики об'єктива камери. На противагу, існують доволі ефективні методи боротьби з оптичними спотвореннями на фотографіях.

В першу чергу необхідно виконати калібрування камери. Калібруванням камери називають розрахунок її внутрішніх і зовнішніх параметрів за отриманими з її допомогою фото та відео. В результаті отримується необхідна для усунення дисторсії інформація про камеру.

В розрахунках коефіцієнтів дисторсії бібліотека OpenCV враховує як тангенціальні та і радіальні складові [3]. Для радіальної складової використовується наступна формула:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6),$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6).$$

Таким чином, піксельна точка на спотвореному зображенні маючи координати (x, y) , отримує координати $(x_{corrected}, y_{corrected})$ на виправленому зображенні. Присутність радіальної дисторсії виявляється в ефектах «бочки» та «риб'ячого ока» на фотографіях.

Тангенціальна дисторсія виникає через те що лінзи об'єктива камери не можуть бути ідеально паралельні площині яка фотографується. Її можна усунути за наступною формулою:

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)],$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy].$$

В підсумку можна виділити п'ять параметрів спотворення, які в OpenCV представлені у вигляді однорядкової матриці з п'ятьма стовпцями.

$$Distortion_{coefficients} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3).$$

Для перетворення координат точок реального світу в координати пікселів на фотознімку, застосовується матриця камери:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

Присутність складової w у складі матриці піксельних координат пояснюється необхідністю дотримання відповідності розмірностей матриць координатних систем при множенні, і не носить реального змісту. Невідомими параметрами є f_x і f_y – фокусні відстані камери, та (c_x, c_y) , які представляють оптичні центри, виражені в піксельних координатах. Якщо для обох осей використовується спільна фокусна відстань зі співвідношенням a (зазвичай 1), тоді $f_y = f_x * a$ і рівняння вище буде мати єдину фокусну відстань f .

Процес знаходження матриць камери і коефіцієнтів дисторсії називається калібруванням. Розрахунки їх параметрів виконуються за допомогою геометричних рівнянь. Застосовувані рівняння залежать від типу об'єктів що використовуються для калібрування. В даний момент OpenCV підтримує три типи об'єктів для калібрування:

- класична чорно-біла шахова дошка;
- симетричний круговий патерн;
- асиметричний круговий патерн.

Необхідно зробити знімок цих патернів і знайти їх на ньому за допомогою засобів OpenCV. Кожному знайденому патерну відповідає нове рівняння. Для рішення цього рівняння треба зробити деяку кількість знімків і проаналізувати їх, щоби скласти систему з рівнянь. Наприклад, теоретично, шаховий патерн передбачає мінімум два знімки. Але цього можна уникнути, зробивши один знімок кількох патернів одразу.

Для наших цілей краще за все підходить шаховий

патерн, оскільки він, по суті, потребує лише «сітку» з точок. Справа в тому, що шахова дошка є лише обгорткою для полегшення розпізнавання. Насправді OpenCV цікавлять лише позиції вузлів шахового патерну.

Таким чином, необхідно лише надрукувати і сфотографувати калібрувальну сітку, яка складається зі зручних для розпізнавання об'єктів, розташованих у вузлах уявної решітки з комітками у вигляді квадратів з попередньо заданою стороною. Найбільш зручними для розпізнавання об'єктами будуть так звані реєстраційні марки – чорні круги діаметром 8–10 мм, які мають радіальну симетрію з огляду на їх форму і можуть бути легко розпізнані як камерою на різаку станка, так і засобами OpenCV. Сторона квадрата особливого значення для розпізнавання не має, але мусить бути достатньо великою щоб, вузли сітки не зливались між собою на фотографії, і в той же час достатньо малою щоб забезпечити велику точність позиціонування. Останнє твердження витікає з того що калібрувальна сітка в нашому випадку служить для двох цілей – надає патерн для калібрування камери і утворює основу системи позиціонування об'єктів на різальній поверхні.

Розроблювана система передбачає нерухоме закріплення камери над різальною поверхнею, з чого слідує, що один і той самий піксель на різних фотографіях камери буде представляти одну й ту саму точку в просторі (на різальній поверхні). Залишається лише задати точку відліку і встановити опорні точки на різальній поверхні з відомими координатами, відносно яких будуть розраховуватись координати всіх інших.

ПОЗИЦІОНУВАННЯ ОБ'ЄКТІВ

Позиціонування в межах самої калібрувальної сітки є найпростішою задачею, оскільки крок сітки (сторона квадрата комірки) відомий заздалегідь. Зробивши знімок калібрувальної сітки (калібрувальний знімок), достатньо тільки розрахувати відношення відстані в міліметрах до відстані в пікселях між вузлами сітки і користуватись його значенням для знаходження відстаней в межах сітки.

Для позиціонування відносно точки відліку різак необхідно разове застосування локальної камери, котра повинна знайти кожен вузол калібрувальної сітки і надати його координати, знайдені з високою точністю, системі комп'ютерного зору, яка «прив'яже» реальні координати вузлів в міліметрах відносно точки відліку, до їх координат на фотографіях в пікселях.

Подальша схема позиціонування нічим не відрізняється від поданої вище. Проте точність буде вищою, оскільки без даних від локальної камери ми зможемо знати лише заданий при виготовленні крок між вузлами калібрувальної сітки, який в процесі її виготовлення (при друкуванні) може бути спотворений в силу різних технічних факторів.

Після отримання даних від обох камер, інформації про розміри та крок калібрувальної сітки та команди запуску калібрування, програма починає пошук реєстр. марок на калібрувальній фотографії.

Результати пошуку у вигляді списку координат пікселей центрів марок на фотографії проходять процедуру пошуку закономірностей розташування, на основі яких будуються об'єкти калібрувальних сіток. Основу цих об'єктів складають три двомірні масиви точок, розміри яких співпадають з розмірами сіток, а комірки призначені для збереження координат вузлів сітки, тобто центрів марок. Перший масив зберігає координати вузлів сітки в піксельних координатах фотографії, а другий – отримані від камери різача відповідні координати реального світу в міліметрах. Третій призначений для зберігання неспотворених координат вузлів на «вирівняній» фотографії.

Після побудови всіх об'єктів сіток відповідно до кількості реальних сіток на фотографії, та заповнення їх перших масивів в порядку реального розташування вузлів, відбувається пошук відповідності розташування комірок першого масиву та точок, отриманих від камери різача. Таким чином встановлюється однозначна відповідність реальних координат вузлів калібрувальних сіток в міліметрах до їх координат на фотографії в пікселях та заповнюються другі масиви об'єктів сіток. На основі отриманих даних визначається також позиція точки відліку (початку координат) на фотографії.

Далі слідує власне калібрування камери. Для отримання калібрувальних матриць OpenCV пропонує метод `calibrateCamera()` [4]. Він приймає координати точок патернів (калібрувальних сіток) в просторі, їх координати на фотографії (перші два масиви об'єктів сіток), а також пусті матриці для власних коефіцієнтів та коефіцієнтів спотворення камери, які заповнює розрахованими значеннями. Отримані матриці зберігаються і далі використовуються для розрахунку «правильних» координат вузлів калібрувальної сітки за допомогою функції OpenCV `undistortPoints()`. Перераховані неспотворені координати записуються в третій масив об'єктів сіток. В подальшому всі точки, знайдені на фотографіях, будуть проходити таку обробку для підвищення точності позиціонування.

На цьому процес калібрування закінчується. В результаті, програма комп'ютерного зору отримує всю необхідну інформацію для точної ідентифікації та позиціонування об'єктів на наступних фотографіях різальної поверхні.

ВИСНОВКИ

Розроблена та описана концепція модуля калібрування та позиціонування системи комп'ютерного зору для точної ідентифікації та позиціонування об'єктів цифрової нарізки за фотознімком робочої поверхні станка для цифрової нарізки. Описано та обґрунтовано принцип роботи алгоритмів які відповідають за калібрування камери та утворення високоточної системи позиціонування на основі калібрувальної сітки з шаховим патерном, та даних про патерн від локальної камери різця.

ЛІТЕРАТУРА

1. Шапиро Л. Комп'ютерное зрение / Л. Шапиро, Дж. Стокман. – Москва: БИНОМ. Лаборатория знаний, 2013. – 752 с.

2. Иофис Е. А. Фотокинетика / Е. А. Иофис. –

Москва: «Советская энциклопедия», 1981. – 447 с.

3. Szeliski R. Computer Vision: Algorithms and Applications / Richard Szeliski. – New York: Springer, 2011. – 957 с. – (Texts in Computer Science).

4. Camera calibration With OpenCV [Електронний ресурс] – Режим доступу до ресурсу: https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html (дата звернення 26.10.2017).

Система для організації інтерактивних квест-ігор побудована на Bluetooth-маячках

Андрій Коноваленко
КПІ ім. Ігоря Сікорського
Київ, Україна
konovalenko.andrii@gmail.com

Анотація. У роботі розглянуто систему інтерактивних квест-ігор, що побудована з використанням Bluetooth-маячків та мобільного додатку. Мобільний додаток дозволяє відвідувачу брати участь у квест-грі шукаючи маячки та відповідаючи на питання квесту. Система представлена у вигляді веб-сервісу, який приймає та обробляє дані, гібридного мобільного додатку для відвідувача (користувача) та Bluetooth-маячків. Мобільний додаток розроблено з використанням Angular 4, TypeScript, HTML5 та Ionic Framework 3.

Ключові слова: інтерактивний квест, гра-квест, розваги, Bluetooth, гібридний мобільний додаток.

ВСТУП

Смартфони та Інтернет вже давно стали невід'ємною частиною життя людини. Сучасне суспільство значну частину свого часу проводить у віртуальній реальності. Це дозволяє нам отримувати цікаву інформацію, спілкуватись та розважатись граючи у ігри. Дивлячись на це з іншого боку ми розуміємо, що Інтернет – це один із важливіших каналів таргетованої реклами для будь-якої сучасної компанії. Технологія Bluetooth-маячків набирає популярності завдяки можливості їх широкого застосування: Bluetooth-маркетинг, навігація в приміщенні, «Physical Web» [1] та ін.

Проведення інтерактивних квестів та акцій потребує довготривалої підготовки та контролю учасників, але являє собою гарний спосіб залучення нових клієнтів, бо ви маєте можливість безпосередньої взаємодії з ними та даруєте їм позитивні емоції.

Квест – аматорське спортивно-інтелектуальне змагання, основою якого є послідовне виконання задалегідь підготовлених завдань командами або окремими гравцями.

Під час гри команди вирішують логічні завдання, здійснюють пошук на місцевості, будують оптимальні маршрути переміщення, шукають оригінальні рішення і підказки. Після завершення чергового завдання команди переходять до виконання наступного. Перемагає команда, що виконала всі завдання швидше за інших.

Створений мобільний додаток дозволяє організувати проведення інтерактивних квестів на місцевості та автоматизувати процес взаємодії користувача з елементами квесту, що представлені у реальному світі, – Bluetooth-маячками, надання користувачу прізв'язок та знижок відбувається автоматично відповідно до отриманих результатів.

АРХІТЕКТУРА СИСТЕМИ

Система інтерактивних квест-ігор (рис. 1) є програмно-апаратним комплексом, що складається з трьох частин: серверна частина, мобільний додаток та Bluetooth-маячки.

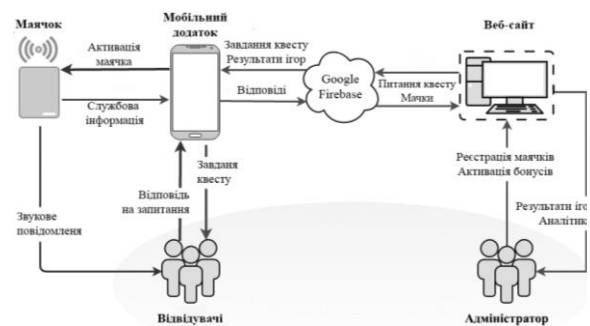


Рис. 1. Схема роботи системи

Bluetooth-маячок розроблено на основі SoC NRF51822, який виробляє Nordic Semiconductor. Даний чіп підтримує протоколи Bluetooth LE 4.0 та Bluetooth LE 5.0 [2], що дозволило створити автономні та енергоефективні Bluetooth-маячки, розрахункова тривалість роботи яких становить від 12 до 16 місяців, а використання зовнішньої антени дозволило збільшити радіус дії з 15 до 100 метрів (пряма видимість), що є дуже важливим показником для реалізації інтерактивних квест-ігор.

Мобільний додаток побудовано з використанням сучасної платформи Ionic Framework 3, яка дозволяє розробляти мобільні додатки з використанням технологій веб стеку [3]. Завдяки використанню Ionic Framework 3 додаток виглядає нативно на будь-якій мобільній ОС.

Другою перевагою використання Ionic Framework 3 є можливість роботи з апаратними функціями пристрою та системи, такими як аудіо, камера, Bluetooth та ін. Можливість роботи з Bluetooth, абстрагуючись від системи та пристрою, дозволила значно прискорити розробку та, що є важливим, зменшити об'єм коду майже у 2 рази, порівняно з розробкою окремих додатків для Android та iOS. Архітектура мобільного додатку (рис 2.) поділяється на три модулі: модуль авторизації, модуль роботи з квестами та модуль для взаємодії з Bluetooth-маячками. Завдяки поділу системи на такі модулі [4] ми маємо можливість розвивати функціонал роботи з

квестами та додавати нові ігрові функції не зволікаючи на процедуру взаємодії з Bluetooth-маячками.

Серверна частина була побудована з використанням сервісів Google Firebase. Cloud Firestore [5] – це сервіс, який дозволяє зберігати дані у нереляційній базі даних, яка синхронізується з клієнтами у режимі реального часу, його використання дозволило організувати швидко та якісне зберігання даних, налаштувати правила доступу до даних без довготривалого написання серверного коду, забезпечити можливість зберігання даних у нереляційній базі даних Cloud Firestore і швидко реалізувати основну бізнес-логіку. Можливість роботи з базою даних без підключення до інтернету дозволила реалізувати участь у грі навіть тоді, коли відсутній доступ до Інтернету, а при його наявності автоматично синхронізувати зміни з сервером.

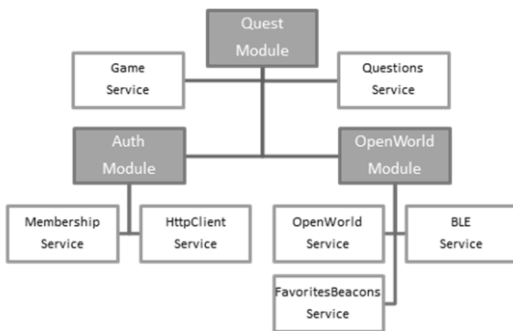


Рис. 2. Архітектура мобільного додатку

Використання сервісу Firebase Authentication [6] дозволило реалізувати можливість реєстрації та авторизації користувачів через соціальні мережі або з використанням email.

Під час квесту маячок виступає у ролі об'єкту, який необхідно знайти (наблизитись на певну відстань) та відповісти на запитання, яке закріплено за цим маячком. Користувач приймає участь у квесті за допомогою мобільно додатку, який дозволяє зареєструватись у квесті, знаходити маячки за допомогою віртуального сенсора, який візуалізує потужність сигналу від маячка, відповідати на питання квесту та використовувати отримані знижки та бонуси, інтерфейс додатку зображено на рис. 3.



Рис. 3. Початок гри та отримана знижка

Серверна частина реалізує основну бізнес-логіку, а web-інтерфейс дозволяє адміністратору додавати нові маячки та запитання у систему, переглядати результати ігор.

Мобільний додаток є не тільки засобом для участі у грі, а й інструментом для менеджера, який дозволяє: створювати нову команду, додавати маячки та питання, активувати знижки та бонуси. Уся взаємодія у системі між клієнтом та менеджером реалізована з використанням QR-кодів, що дозволяє зручно та швидко виконувати додавання користувачів у команду та активацію бонусів.

ВИСНОВКИ

Результатом роботи стало створення системи, що дозволяє автоматизувати проведення командних інтерактивних квест-ігор. У якості інструменти гри відвідувач використовує свій особистий смартфон, а роль об'єктів квесту виконують спеціально розроблені Bluetooth-маячки.

Система з 7 маячків була встановлена та протестована у канатному-парку у м. Київ. Протягом тестування було визначено, що подальше використання цієї системи дозволить зацікавити відвідувачів парку та збільшити кількість клієнтів.

Аналіз поведінки користувачів дозволив визначити, що у наступній версії системи Bluetooth-маячки будуть обладнані світлодіодною індикацією, сенсорами дотику та зчитувачем RFID-міток – це дозволить додати нові типи завдань у квест та ще більше зацікавити аудиторію.

ЛІТЕРАТУРА

1. <https://google.github.io/physical-web/> (дата звернення 10.10.2017)
2. <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832> (дата звернення 10.10.2017)
3. <http://ionicframework.com/> (дата звернення 10.10.2017)
4. https://uk.wikipedia.org/wiki/Модульне_програмування (дата звернення 10.10.2017)
5. <https://firebase.google.com/products/firestore/> (дата звернення 10.10.2017)
6. <https://firebase.google.com/products/auth/> (дата звернення 10.10.2017)

Мікроконтролер для керування послідовними портами вводу-виводу

Сергієнко Анатолій
Михайлович
КПІ ім. Ігоря Сікорського
Київ, Україна
aser@comsys.kpi.ua

Орлова Марія Миколаївна
КПІ ім. Ігоря Сікорського
Київ, Україна
orlova@ua.fm

Молчанов Олексій
Андрійович
КПІ ім. Ігоря Сікорського
Київ, Україна
oleksii.molchanov@gmail.com

Анотація. Запропонована архітектура мікроконтролера для керування послідовними портами вводу-виводу на основі стекового процесора SM8, яка призначена для реалізації у програмованих логічних інтегральних схемах. Мікроконтролер з цією архітектурою має невеликі апаратні витрати, скорочений об'єм програмного забезпечення а також систему команд, яка адаптована до задачі, що вирішується.

Ключові слова: стековий процесор, Forth, ПЛІС, VHDL.

Вступ

Багато вбудованих комп'ютерних систем мають зв'язок через такі поширені послідовні інтерфейси, як I2C, SPI, Ethernet та інші. Якщо така система реалізована навколо програмованої логічної інтегральної схеми (ПЛІС), то поширеніше рішення — це вбудоване ядро мікроконтролера, яке керує цими інтерфейсами. Для цього є невеликий вибір з універсальних RISC-процесорів, таких як Xilinx Picoblaze, Microblaze, Altera Nios [1, 2], або клони поширеного мікроконтролера, такого як i8051 [3].

Але у багатьох випадках обмін даними виконується за нескладним протоколом і з порівняно невеликою швидкістю, як наприклад, через інтерфейс I2C. При цьому можливості мікроконтролерів використовуються нерационально.

При реалізації у ПЛІС мікроконтролерів важливо мати мінімізований об'єм матзабезпечення, бо об'єм блоків пам'яті ПЛІС, є суттєво обмеженим. При цьому бажано мати такий мікроконтролер, для якого зручно складати програми пересилки даних через інтерфейси. У доповіді пропонується архітектура такого мікроконтролера.

СТРУКТУРА МІКРОКОНТРОЛЕРА

Для організації обмінів між послідовними портами вводу-виводу зі швидкістю до 10 – 100 Мбод, мікроконтролер має задовольняти наступні вимоги:

- необхідні апаратно реалізовані приймачі-передавачі послідовних бітових потоків у відповідності зі стандартом інтерфейсу;

- достатньо 8-розрядного мікропроцесора зі швидкодією близько 100 млн. команд за секунду для а формування і читання пакетів, які складаються з байтових даних;

- необхідна взаємна синхронізація приймачів-передавачів та мікропроцесора через обмін сигналами запуску-готовності;

- при невеликій кількості портів система переривання може бути реалізована програмно;

- бажано мати зручні засоби для швидкої розробки вбудованого матзабезпечення мікроконтролера, які адаптовані до вирішення даної задачі.

Як правило, послідовні вводи-виводи приєднані до одного мікропроцесорного ядра. Такий порт сприймає специфічну послідовність керуючих команд. Наприклад, порт інтерфейсу I2C має команди: START – запуск пересилок, PAUSE – пауза в пересилках, A_SEND – передачі байта адреси, D_SEND – передачі байта даних, STOP – зупинка пересилок та інші.

Перелічені вимоги задовольняє мікропроцесорне ядро SM8, яке має мінімізовані апаратні витрати та зручності у програмуванні нескладних алгоритмів [4].

МІКРОПРОЦЕСОРНЕ ЯДРО SM8

Восьмирозрядне мікропроцесорне ядро SM8 виконане за двохстековою архітектурою. Воно має стек адрес повернення глибиною 33, стек даних глибиною 34, пам'ять програм об'ємом до 7936 байт та пам'ять даних об'ємом до 256 байт. До нього можна підключити до 32 периферійних регістрів, в тому числі регістрів даних і керування портів вводу-виводу.

Система команд ядра має 23 базові команди. Усі команди, крім команд виклику підпрограми call, умовного переходу if та вводу константи lit, є однобайтовими. Команди call, if та lit виконуються за два такти, а решта команд – за один. Команди виконуються у конвеєрному режимі. Крім того, є можливість додати до 128 команд користувача, які виконуються як команда call з виконанням відповідної підпрограми, але мають довжину один байт.

Завдяки тому, що при програмуванні ядра SM8 часто використовуються виклики підпрограм, одержуються відносно короткі програми. За рахунок цього, а також через часте використання команд if та lit, середня тривалість виконання однієї команди в даній архітектурі складає півтора такти.

Команди користувача можуть зберігатись як в пам'яті програм, так і в пам'яті даних. Завдяки цьому, можна приймати з порту і записувати в пам'ять динамічний сценарій обробки даних або виконувати граматичний розбір рядка. Наприклад, це може бути заголовок пакету, що пересилається або заданий алгоритм поведінки з послідовним інтерфейсом.

АСЕМБЛЕР ЯДРА SM8

Для програмування мікропроцесорного ядра SM8 було розроблено асемблер на мові Java. Нижче представлено приклад програми на мові асемблера SM8, яка виконує однократну пересилку байту даних у порт I2C.

```

DEFINE nap 9 \ розмір пам'яті = 2^9
DEFINE WAITRDY 82h \ команда - чекати готовності порту
DEFINE DELAYN 83h \ команда - затримка на N циклів
EQU START 2
EQU A_SEND 4
EQU D_SEND 5
EQU STOP 12
EQU PAUSE 15
ORG 256 \ початок сегменту програми
: WR2I2C (r1 - I2C addr., r2 - inner addr., r3 - byte,
         r8 - I2C data, r9 - I2C control)
    lit START out r9
    inr r1 out r8 lit A_SEND out r9 WAITRDY
    inr r2 out r8 lit D_SEND out r9 WAITRDY
    inr r3 out r8 lit D_SEND out r9 WAITRDY
    lit STOP out r9
    lit PAUSE out r9
    lit 100 DELAYN xor if END
;
: DELAYN (N -- - N cycles)
    dec dup ifn DELAYN
;
: WAITRDY (do while rdy=1)
    inr r10 \0-th bit = rdy
    lit 1 and if WAITRDY
;
: END

```

Мова асемблера ядра SM8 запозичує синтаксис у мови Forth. Тому тут коментарі вкладені в круглі дужки чи слідує за оберненою похилою. Мітка слідує за двокрапкою, оператори та літерали відокремлюються пробілами. Крапкою з комою позначається оператор повернення з підпрограми. Команди `inr`, `out` виконують читання-запис у регістри даних.

Прагомою `DEFINE` задається код операції команди користувача. Як результати асемблювання, генеруються VHDL-файли описів пам'яті програм даних, а також блоку перекодування кодів операцій команд користувача у адреси відповідних підпрограм.

Як результат, мова, якою програмується мікроконтролер, за своїми користувацькими властивостями займає проміжне становище між мовою асемблера звичайного мікроконтролера та мовою високого рівня. Завдяки цьому, значно прискорюється написання та відлагодження програм. Для виконання останнього, VHDL-модель ядра SM8 оснащена дізасемблером.

ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ

Мікроконтролер був використаний для керування системою генерування радіосигналів у FM-діапазоні. У склад системи входять універсальний послідовний приймач-передавач (UART), три порти I2C та декілька керуючих регістрів. Одержані параметри мікроконтролера та його складових при їх конфігуруванні у ПЛІС Xilinx Spartan-6 представлені у таблиці. Для порівняння, у таблиці показані також параметри 8-розрядних мікроконтролерів, які часто використовуються для вирішення аналогічних задач

Аналіз даних у таблиці показує, що запропонований мікроконтролер має на 40% більші

апаратні витрати, ніж мікроконтролер Picoblaze. Але, враховуючи те, що в Picoblaze одна команда виконується за два такти, то мікроконтролер на базі SM8 має удвічі більшу швидкодію. Крім того, новий мікроконтролер має приблизно удвічі коротші програми і його зручніше програмувати.

ВИСНОВКИ

Запропонований мікроконтролер для керування послідовними портами вводу-виводу має невеликі апаратні витрати при високій продуктивності та скороченому об'ємі програм. Мікроконтролер описано мовою VHDL і він може бути реалізований у ПЛІС будь-якої серії. Завдяки тому, що програмування виконується Forth-подібною мовою і є можливість додавати команди користувача, а розроблений асемблер компілює програми безпосередньо у VHDL-опис, процес розробки застосунку є зручним і нетривалим.

Таблиця 1

Параметри мікроконтролера та його складових

Мікроконтролер та його складові	Апаратні витрати у ЛТ	Апаратні витрати у тригерах	Максимальна тактова частота, МГц
I2C core	105	66	205
UART	32	48	280
Ядро SM8	156	50	150
Мікроконтролер	324	204	152
FS8051 разом з I2C [3]	1413	110	80
KCPSM6 (Picoblaze) разом з I2C і UART [2]	231	—	105

ПЕРЕЛІК ПОСИЛАНЬ

1. Processor Design. System-on-Chip Computing for ASICs and FPGAs / J. Nurmi., Ed. — Springer. — 2007. — 525 p.
2. Chapman, K. PicoBlaze for Spartan-6, Virtex-6, and 7-Series (KCPSM6). — Xilinx, Inc. — 2012. — 118 p.
3. Сергієнко А.М., Лепеха В.Л. Деякі особливості проектування мікроконтролерів для СНК // Вісник НТУУ «КПІ», сер. Інформатика, управління та обчислювальна техніка. — Т. 50. — 2009. — С. 70-73.
4. Сергієнко А. М., Молчанов О. М. Мікроконтролер з системою команд, що розширюється // Праці міжн. конф. Інформаційні системи та технології. Summer InfoCom Advanced Solutions 2017. — С. 58–60.

Розробка архітектури системи автоматизованого збору, обробки та аналізу даних на основі технології Big Data

Рижко Борис Володимирович
КПІ ім. Ігоря Сікорського
Київ, Україна
r.borya@gmail.com

Смолинець Остап Тарасович
КПІ ім. Ігоря Сікорського
Київ, Україна
o.smolynets@ukr.net

Анотація. Для зменшення витрат на побудову високопродуктивних кластерних систем (ВПКС) пропонується використовувати відкрите програмне забезпечення, що базується на технології Big Data. Описаний метод побудови архітектури систем автоматизованого збору, обробки та аналізу даних (САЗ).

Ключові слова: обробка даних, великі дані.

ВСТУП

На сьогоднішній день актуальною стала проблема збору, обробки і аналізу інформації. Прикладом є аналіз продаж і витрат у розподілених системах, управління трафіком через розподілені автоматизовані системи, обробка великої кількості відеоінформації, тощо.

Для рішення цих завдань активно використовують високопродуктивні кластерні системи [1]. Вони побудовані за архітектурою кластерних обчислювальних систем з розподіленою пам'яттю.

Одним із завдань, які потрібно вирішити при проектуванні ВПКС, є формалізація та моделювання програмного забезпечення і роботи паралельних систем. Ці задачі вирішувалися з використанням теорії мереж Петрі [2,3].

Проблемою при розробці ВПКС стає дуже висока вартість запуску і експлуатації подібних систем. Для зменшення витрат пропонується низка відкритих програмних комплексів для реалізації ВПКС.

Завдання даної системи дещо відрізняються від типових ВПКС. Окрім традиційної для ВПКС функції аналізу даних, до САЗ включаються також функції збору та обробки великих наборів даних (наприклад, погодних даних, геолокаційних даних, даних зібраних з «розумних» пристроїв та ін.)

Загальна шкала великих наборів даних постійно змінюється, залежить від часу і може істотно відрізнятися в різних організаціях експлуатантх, що спонукає до розробки такого класу систем.

Таким чином основною метою є розробка архітектури САЗ з використанням відкритих програмних систем.

Для реалізації поставленої мети потрібно:

- сформувати структуру САЗ на основі ВПКС, програмною базою якої має бути відкритий програмний комплекс;

- провести вибір та обґрунтування елементів структури САЗ;

- побудувати програмну і апаратну архітектуру САЗ на основі ВПКС.



Рис. 1. Структурна схема САЗ

ВИМОГИ ДО САЗ

САЗ, побудовані з використанням технологій «Big Data» можуть бути використані в будь-яких сферах бізнесу.

Такі системи повинні мати необхідний інструментарій для обробки як структурованих, так і неструктурованих даних дуже великих обсягів.

В заданих умовах необхідно забезпечити можливості для швидкого збору дуже великої кількості інформації, швидкої обробки й аналізу цього масиву даних.

Таким чином, до САЗ можна встановити наступні вимоги:

- масштабованість;
- можливість збирати структуровані та неструктуровані дані;
- швидка обробка великих обсягів даних;
- можливість зберігати дані у кластерній файлової системі;
- відмовостійкість;
- представлення результатів аналізу у вигляді, зручному для подальшого відображення.

Продуктивність інфраструктури розподілених обчислень (ІРО) цілком залежить від технічного та програмного забезпечення.

СТРУКТУРА СИСТЕМИ

Виходячи з зазначених вимог до САЗ, було визначено наступні структурні елементи:

інфраструктура розподілених обчислень (ІРО), підсистема збору та обробки інформації, система керування базами даних (СКБД).

Структурна схема САЗ наведена на рис. 1.

Визначені елементи дозволяють розділити функції всієї САЗ, що значно полегшує налагодження системи і закладає принципи модульності. Така структура також визначає взаємозамінність кожної з підсистем у випадку необхідності.

В свою чергу, кожна з підсистем повинна також мати модульний принцип побудови, що є одним з критеріїв їх вибору.

КОМПОНЕНТИ СИСТЕМИ

Для вибору окремих компонентів було встановлено ряд вимог, що повинні задовольнятися, для реалізації поставленої мети. Таким чином, ІРО повинна відповідати наступним вимогам:

- горизонтальна масштабованість;
- наявність кластерної файлової системи (ФС);
- обробка великих даних;
- наявність інтерфейсів доступу до ФС та обчислювальних модулів;
- відмовостійкість.

З огляду на поставлені критерії, було проаналізовано існуючі рішення і обрано ІРО Hadoop.

Проект складається з чотирьох модулів [4]:

- Hadoop Common (набір інфраструктурних програмних бібліотек і утиліт, використовуваних для інших модулів і споріднених проектів);
- HDFS (розподілена файлова система);
- YARN (система для планування завдань і управління кластером);

– Hadoop MapReduce (платформа програмування і виконання розподілених MapReduce-обчислень).

При реалізації обчислень в парадигмі MapReduce, вони приймають набір вхідних пар ключ-значення і створюють набір вихідних пар ключ-значення. Користувач бібліотеки MapReduce виражає розрахунок як дві функції: Map та Reduce.

Map, написаний користувачем, приймає пару вхідних даних і видає набір проміжних пар ключ-значення. Бібліотека MapReduce об'єднує всі проміжні значення, пов'язані з тим самим проміжним ключем, і передає їх в функцію Reduce.

Функція Reduce, також написана користувачем, приймає проміжний ключ і набір значень для цього ключа. Вона об'єднує разом ці значення, щоб сформуванати, якщо можливо, менший набір значень. Зазвичай при виклику Reduce вихідне значення або не повертається, або повертається лише одне. Проміжні значення подаються на функцію Reduce користувача за допомогою ітератора. Це дозволяє обробляти списки значень, які занадто великі для розміщення в пам'яті.

Вибір підсистеми збору та обробки інформації базується на наступних критеріях:

- модульний дизайн;
- масштабованість;
- сумісність з базою даних;
- сумісність з ІРО;
- гнучкі налаштування;

- розподіленість виконання задач;
- багатопоточність;
- відкрите програмне забезпечення.

В якості підсистеми збору та обробки інформації було обрано відкрите програмне забезпечення Apache Flume.

Flume є розподіленим та надійним сервісом збору, обробки та переміщення великих обсягів даних. Він має просту та гнучку архітектуру, що базується на потоках даних [5].

Критерії вибору СКБД наступні:

- масштабованість;
- сумісність з ІРО;
- сумісність з підсистемою збору та обробки інформації;
- можливість використання інтерфейсу доступу до СКБД зовнішніми модулями;
- швидкий доступ до великих обсягів даних;
- надійність;
- відкрите програмне забезпечення.

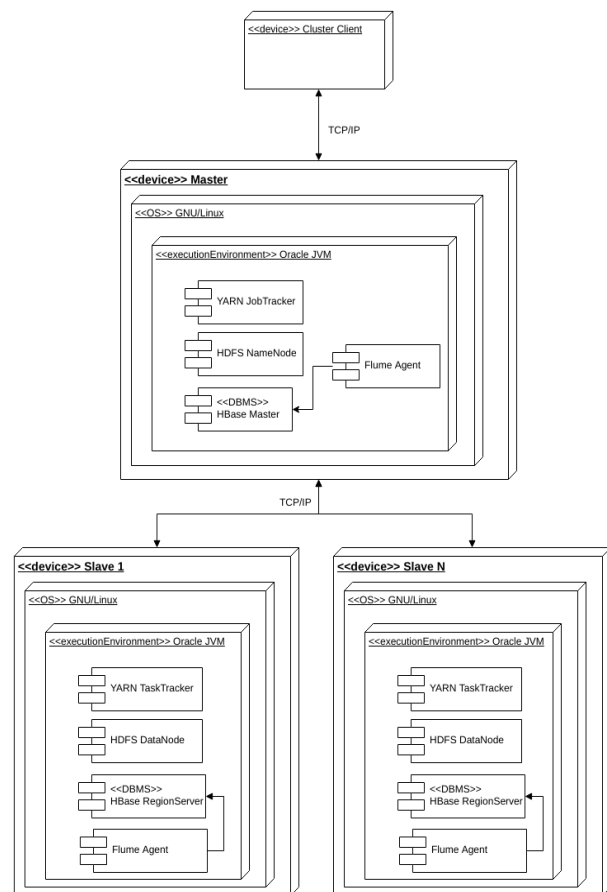


Рис. 2. Архітектура САЗ

Заданим вимогам відповідає обрана СКБД Apache HBase. Це стовпчико-орієнтована NoSQL СКБД. NoSQL означає, що СКБД відрізняється від традиційного представлення даних у реляційних СКБД. Дані у HBase зберігаються у вигляді структурованих та напівструктурованих пар ключ-значення. Дана СКБД надає наступні можливості [6]:

- запис-зчитування у режимі реального часу;
- інтерфейс управління СКБД;

- конфігурація розподіленої СКБД між вузлами;
- додаткові інтерфейси для доступу зі сторонніх програм.

АРХІТЕКТУРА СА3

Розглянемо архітектуру СА3, що побудована на основі структурної схеми, в нотації UML [7] (рис. 2).

В розробленій архітектурі були враховані усі вимоги, що пред'являються структурною схемою та окремим програмним забезпеченням. Отримана архітектура системи відповідає типу master-slave.

Master-вузол керує операціями простору імен файлової системи (такими як відкриття, закриття та перейменування файлів і каталогів), регулює доступ клієнтів до файлів, є точкою взаємодії між користувачами та системою MapReduce; виконує призначення регіону, контролює операції DDL (створення, видалення таблиць) бази даних HBase.

Відповідає за:

- додаткові інтерфейси для доступу зі сторонніх програм;
- координацію серверів регіону;
- призначення регіонів при запуску;
- перепризначення областей для відновлення або балансування навантаження;
- моніторинг усіх примірників RegionServer у кластері (прослуховує сповіщення від Zookeeper).

Крім того, на master-вузлі розміщено приймач даних від усіх агентів (slave-вузлів), що збирають та обробляють інформацію з різних джерел.

Slave-вузли несуть відповідальність за обслуговування запитів на читання та запис клієнтів файлової системи разом із виконанням створення, видалення та реплікації блоків за інструкцією master-вузла; виконують обчислювальні задачі MapReduce, що надаються master-вузлом; виконують задачі збору, первинної обробки інформації, запису до бази даних.

ВИСНОВКИ

В результаті аналізу різних IPO було виявлено, що на сьогоднішній день, за співвідношенням вартість-ефективність для розробки автоматизованої системи збору, обробки та аналізу інформації доцільно використовувати кластери, побудовані з використанням Hadoop, для загального користування.

Розроблена архітектура і СА3 в цілому забезпечує: масштабованість, модульність та гнучкість у налаштуванні всіх параметрів.

Модульність архітектури дозволяє підключати додаткові модулі до складу робочих програмних систем або окремо від них з використанням програмних інтерфейсів цих систем.

Перевагами системи можна назвати:

– поєднання ресурсів: об'єднання наявного простору для зберігання даних є чіткою перевагою, але процесор та об'єднання пам'яті також надзвичайно важливі; обробка великих наборів даних вимагає великої кількості всіх трьох цих ресурсів;

– висока доступність: кластери можуть забезпечувати різний рівень стійкості до відмов та наявності гарантій, щоб запобігти апаратному або програмному збою від доступу до даних та обробки; це стає все більш важливим, оскільки ми продовжуємо підкреслювати важливість аналітики в реальному часі;

– легка масштабованість: кластери полегшують масштабування горизонтально, додавши до групи додаткові машини. Це означає, що система може реагувати на зміни вимог до ресурсів без розширення фізичних ресурсів на машині.

За співвідношенням вартості до ефективності розроблена система має переваги перед іншими аналогічними комерційними рішеннями.

ЛІТЕРАТУРА

1. November 2015 | TOP500 Supercomputer Sites [Електронний ресурс] – Режим доступу до ресурсу: <http://www.top500.org/lists/2015/11/> (дата звернення 30.05.2017). — Назва з екрана.

2. Катін П. Ю., Досенко Є. В., Іванів Р. Б. Моделювання програм з паралельними обчисленнями за допомогою мереж Петрі / П. Ю. Катін, Є. В. Досенко, Р. Б. Іванів // Економіка і управління. – 2013. – №4(60). – С.124-132

3. Катін П. Ю. Моделювання процесів синхронізації багато поточних програм з використанням мереж Петрі / П. Ю. Катін // Автоматика-2014: Матеріали 21-ї Міжнародної конференції з автоматичного управління, м. Київ, 23-27 вересня 2014 р. – К.: Вид-во НТУУ “КПІ” ВІПІ ВПК “Політехніка”, 2014.–С. 320-321.

4. White T. Hadoop: The Definitive Guide / O'Reilly Media, 2010. – 528 с.

5. Welcome to Apache Flume — Apache Flume [Електронний ресурс] – Режим доступу до ресурсу: <https://flume.apache.org/> (дата звернення 30.05.2017). — Назва з екрана.

6. Apache HBase Reference Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://hbase.apache.org/book.html> (дата звернення 31.05.2017). — Назва з екрана.

7. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя. – М., СПб.: ДМК Пресс, Питер, 2004. – 432 с.

Рецензент: к.т.н., доц. каф. АУТС, КПІ ім. Ігоря Сікорського, П.Ю. Катін

Анализ и оптимизация автономных систем теплоснабжения

Стенин А.А.
КПИ им. Игоря Сикорского
Киев, Украина
alexander.stenin@yandex.ua

Пасько В.П.
КПИ им. Игоря Сикорского
Киев, Украина
vppasko@gmail.com

Шитикова И.Г.
Институт телекоммуникаций и
глобального информационного
пространства НАН Украины
Киев, Украина
irinashitikova54@gmail.com

Аннотация. Предложены методика расчета базовых значений конструктивных параметров многоконтурных теплообменных аппаратов с баком-аккумулятором на стадии проектирования, численная процедура идентификации параметров теплообменных аппаратов в режиме эксплуатации. Показаны преимущества использования для имитационного моделирования SolidWorks, позволяющего воспроизвести любую конфигурацию теплообменных аппаратов, дать количественные оценки и наглядную визуализацию протекания теплообменных процессов.

Ключевые слова: теплообменный аппарат, параметрическая идентификация, динамическая модель, имитационное моделирование.

ВВЕДЕНИЕ

В настоящее время наиболее приемлемыми для практической эксплуатации в жилищно-коммунальном хозяйстве являются многоконтурные теплообменники с последовательным и параллельным расположением змеевиков (рис.1). Но при их практической реализации возникает вопрос, какая именно из предложенных конструкций теплообменников змеевикового типа является более эффективной. Рассматривается запатентованная одним из авторов новая конструкция МТАБ [1] на основе теплообменника (ТО) змеевикового типа с аккумулирующим баком (рис.2).

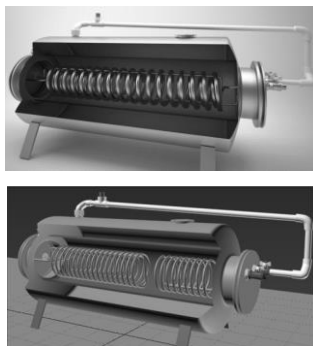


Рис. 1. Теплообменники с параллельным и последовательным расположением змеевиков

Здесь ПАУ – подогревающе-аккумулирующая установка; ИТП – индивидуальный тепловой пункт; G_i – затраты нагреваемых теплоносителей; C_i – теплоёмкости нагреваемых теплоносителей; t_i^I – температура на входе; t_i^{cp} – средние температуры

теплоносителей; $t_1^II, t_2^II, t_3^II, t_4^II$ – температура на выходе контура; $t_{окр}$ – температура окружающей среды; F_i – диаметры контуров; k_i – затраты тепла; $k_{общ}$ – общие затраты тепла. A_i – площадь поперечного сечения; i – соответствующий контур МТАБ.

РАСЧЕТ БАЗОВЫХ КОНСТРУКТИВНЫХ ПАРАМЕТРОВ МТАБ В СТАЦИОНАРНОМ РЕЖИМЕ

Определяем тепловую мощность МТАБ для контуров системы отопления Q2 и горячего водоснабжения Q3 по формулам:

$$Q_2 = G_2 c_2 (t_2^{II} - t_1^{II}), \quad (1)$$

$$Q_3 = G_3 c_3 (t_3^{II} - t_3^I), \quad (2)$$

где C_2 и C_3 – теплоемкость теплоносителей при средних температурах теплоносителей t_2^{cp} та t_3^{cp} :

$$t_2^{cp} = 0,5(t_2^{II} + t_2^I) \text{ и } t_3^{cp} = 0,5(t_3^{II} + t_3^I). \quad (3)$$

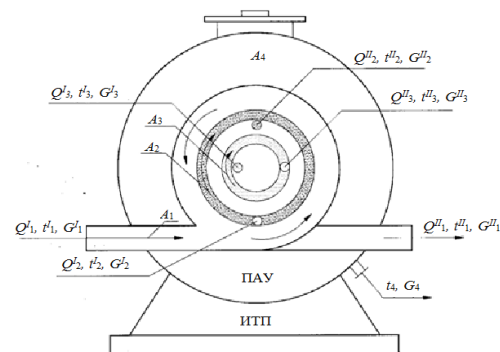


Рис. 2. Теплообменники с параллельным и последовательным расположением змеевиков

С учетом тепловых потерь через наружную поверхность бака-аккумулятора определяем расход первичного теплоносителя по формуле:

$$G_1 = \frac{Q_2 + Q_3}{C_1 (t_1^{II} - t_1^I)} \eta_{ПОТ}. \quad (4)$$

Если первичным теплоносителем является пар, то формула (4) принимает вид:

$$G_{II} = \frac{Q_2 + Q_3}{(i_{II}^{II} - i^I)} \eta_{ПОТ}, \quad (5)$$

где ΔC_1 – теплоемкость первичного теплоносителя при средней температуре

$$t_1^{cp} = 0,5(t_1^{II} - t_1^I), \quad (6)$$

где i_{II}^{II} – энтальпия насыщенного пара при давлении РНАС; i^I – энтальпия воды на линии насыщения при давлении РНАС; $\eta_{ПОТ}$ – коэффициент потерь теплоты через внешнюю поверхность БА.

Далее расписываем уравнение теплового баланса [2]:

– теплообмен между первичным теплоносителем и теплоносителем отопительного контура:

$$Q_{1-2} = K_1 F_1 (t_1^{cp} - t_2^{cp}); \quad (7)$$

– теплообмен между первичным теплоносителем и теплоносителем контура системы горячего водоснабжения:

$$Q_{1-3} = K_2 F_3 (t_1^{cp} - t_3^{cp}); \quad (8)$$

– теплообмен между первичным теплоносителем и водой в БА через корпус теплообменника:

$$Q_{1-4} = K_3 F_1 (t_1^{cp} - t_4^{cp}). \quad (9)$$

При этом средние температуры первичного греющего теплоносителя и теплоносителей, нагреваемых в змеевиках отопительного контура и системы горячего водоснабжения и воды в БА, соответственно равны:

$$t_1^{cp} = 0,5(t_1^I + t_1^{II}), \quad (10)$$

$$t_2^{cp} = 0,5(t_2^I + t_2^{II}), \quad (11)$$

$$t_3^{cp} = 0,5(t_3^I + t_3^{II}), \quad (12)$$

$$t_4^{cp} = 0,5(t_4^I + t_4^{II}). \quad (13)$$

Количество теплоты, принимаемое теплоносителем контура системы отопления:

$$Q_2 = G_2 c_2 (t_2^I + t_2^{II}). \quad (14)$$

Количество теплоты, получаемой теплоносителем контура горячего водоснабжения:

$$Q_3 = G_3 c_3 (t_3^I + t_3^{II}). \quad (15)$$

Количество теплоты, идущей на подогрев воды, находящейся в БА, при снижении водоотведения на горячее водоснабжение через корпус теплообменника:

$$Q_4 = G_4 c_4 (t_4^{cp} + t_4^{cp}). \quad (16)$$

Таким образом, расчет неизвестных параметров основан на совместном решении системы уравнений теплового баланса и уравнения теплопередачи (7) – (9) и (14) – (16). Эта система уравнений является системой уравнений с двенадцатью неизвестными величинами:

$$Q_{1-2}, Q_{1-3}, Q_{1-4}, F_1, F_2, F_3, G_2, t_2^I, t_2^{II}, G_4, Q_2, Q_4.$$

В приведенных уравнениях величину t_2^I можно принять равной t_2 в связи с особенностями

конструкции МТАБ. Отсюда, величина $(t_2^{II} - t_2^I)$ должна быть близкой к нулю. Отклонения ее от нуля может служить мерой обоснованности допустимых упрощений. Если $(t_2^{II} - t_2^I) < 0$, то необходимо увеличить теплообменную поверхность змеевика отопительного контура МТАБ, а если $(t_2^{II} - t_2^I) > 0$, тогда теплообменную поверхность можно не менять, а уменьшить расход первичного теплоносителя G_1 . Задавая часть известных конструктивных параметров возможно определить другие по заданным стационарным температурным режимам. Полученные базовые значения конструктивных параметров должны удовлетворять уравнению суммарного теплового баланса:

$$G_1 \cdot c_1 (t_1^I - t_1^{II}) = G_2 \cdot c_2 (t_2^{II} - t_2^I) + G_3 \cdot c_3 (t_3^{II} - t_3^I) + G_4 \cdot c_4 (t_4^{II} - t_4^I) + G_4 \cdot c_4 (t_4^{II} - t_{окр}); \quad (17)$$

$$k_{общ} \sum F \cdot \Delta t_{cp}^{общ} = k_1 \cdot F_2 (t_1^{cp} - t_2^{cp}) + k_2 \cdot F_3 (t_1^{cp} - t_3^{cp}) + k_3 \cdot F_1^{TO} (t_1^{cp} - t_4^{cp}) + k_4 \cdot F_4^{TA} (t_4^{cp} - t_{окр}).$$

Однако, поскольку система (17) является недоопределенной, то всегда будет присутствовать погрешность в уравнениях теплового баланса (17). Таким образом, для собранного на основе приведенных выше расчетов МТАБ при его дальнейшей эксплуатации и исследования в нестационарных режимах работы, необходимо уточнение значений базовых конструктивных параметров. Для этого ниже предложена численная процедура параметрической идентификации стационарных моделей теплообменников змеевикового типа, основанная на сочетании методов Ньютона-Рафсона и последовательных приближений. Решение находится из условий двух уравнений теплового баланса (17). Процедура идентификации конструктивных параметров носит итеративный характер и может быть представлена как алгоритм получения значений на текущем шаге по значению предыдущего шага:

$$\bar{Y}_k = f(\bar{Y}_{k-1}). \quad (18)$$

Коррекция идентифицированных параметров возможна лишь на основе использования информации об их влиянии на работу теплообменника. Мерой выполнения этого условия являются среднеквадратические невязки \bar{Q}_k между расчетным теплоэнергетическим показателем и полученным на реальном теплообменнике в режиме эксплуатации. Эти невязки зависят от конструктивных параметров \bar{Y} , переменных состояния теплообменника \bar{X} и выбранных постоянных конструктивных параметров \bar{Z} теплообменника. Приравнивая невязки нулю, получим систему уравнений:

$$\bar{Q}(\bar{Y}, \bar{X}, \bar{Z}) = 0. \quad (19)$$

Для решения системы используем методом Ньютона-Рафсона [8], когда на каждой итерации

определяется аддитивная погрешность $\Delta \bar{Y}_k$. Элементы матрицы Якоби в методе Ньютона-Рафсона [3] вычисляются численно последовательным варьированием каждого из конструктивных параметров \bar{Y}_k и определением соответствующих невязок \bar{Q}_k . Параметры считаются идентифицированными при выполнении условия:

$$I = \frac{\sum_{k=1}^n Q_k}{n} \leq \varepsilon, \quad (20)$$

где n – количество итераций, ε – заданная допустимая ошибка.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ТЕПЛО- И МАССООБМЕННЫХ ПРОЦЕССОВ В МТАБ

Для определения оптимальных параметров тепло- и массообменных процессов в МТАБ необходимо, кроме исследования стационарных режимов работы, иметь динамическую модель для нестационарных режимов работы МТАБ. Распределенные в пространстве и времени процессы весьма сложны и описываются уравнениями в частных производных с граничными условиями, которые в данной и других областях требуют применения специальных методов [4,5]. Для исследования нестационарных режимов работы МТАБ (рис.3) в работе авторов [5] получена упрощенная однокоординатная динамическая модель уравнения Навье-Стокса в виде системы уравнений (21) – (23).

$$\begin{aligned} \frac{d\theta_{11}}{d\tau} &= - \left(\frac{\varpi_1}{\Delta x_1 / 2} + \frac{\pi \cdot D_{1n} \cdot \alpha_{11}}{A_1 \cdot \rho_1 \cdot c_{p1}} \right) \cdot \theta_{11} + \frac{\varpi_1}{\Delta x_1 / 2} \cdot \theta_1 + \frac{\pi \cdot D_{1n} \cdot \alpha_{11}}{A_1 \cdot \rho_1 \cdot c_{p1}} \cdot \theta_{cm.11}; \\ \frac{d\theta_{cm.11}}{d\tau} &= - \frac{\alpha_{11} + \alpha_{12}}{x_{cm.1} \cdot \rho_{cm.1} \cdot c_{cm.1}} \cdot \theta_{cm.11} + \frac{\alpha_{11}}{x_{cm.1} \cdot \rho_{cm.1} \cdot c_{cm.1}} \cdot \theta_{cm.11} + \\ &\frac{\alpha_{12}}{x_{cm.1} \cdot \rho_{cm.1} \cdot c_{cm.1}} \cdot \theta_{11}; \\ \frac{d\theta_{21}}{d\tau} &= - \left(\frac{\varpi_2}{\Delta x_1 / 2} + \frac{\pi \cdot D_{2e} \cdot \alpha_{21}}{A_2 \cdot \rho_2 \cdot c_{p2}} \right) \cdot \theta_{21} - \frac{\varpi_2}{\Delta x_1 / 2} \cdot \theta_2 + \frac{\pi \cdot D_{2e} \cdot \alpha_{21}}{A_2 \cdot \rho_2 \cdot c_{p2}} \cdot \theta_{cm.11}. \end{aligned} \quad (21)$$

Для контура системы горячего водоснабжения при $i = 1$ имеем:

$$\begin{aligned} \frac{d\theta_{11}}{d\tau} &= - \left(\frac{\varpi_1}{\Delta x_1 / 2} + \frac{\pi \cdot D_{1e} \cdot \alpha_{21}}{A_1 \cdot \rho_1 \cdot c_{p1}} \right) \cdot \theta_{11} + \frac{\varpi_1}{\Delta x_1 / 2} \cdot \theta_1 + \\ &+ \frac{\pi \cdot D_{1e} \cdot \alpha_{21}}{A_1 \cdot \rho_1 \cdot c_{p1}} \cdot \theta_{cm.21}; \\ \frac{d\theta_{cm.21}}{d\tau} &= - \frac{\alpha_{21} + \alpha_{22}}{x_{cm.2} \cdot \rho_{cm.2} \cdot c_{cm.2}} \cdot \theta_{cm.21} + \frac{\alpha_{21}}{x_{cm.2} \cdot \rho_{cm.2} \cdot c_{cm.2}} \cdot \theta_{11} + \\ &+ \frac{\alpha_{22}}{x_{cm.2} \cdot \rho_{cm.2} \cdot c_{cm.2}} \cdot \theta_{31}; \\ \frac{d\theta_{31}}{d\tau} &= - \left(\frac{\varpi_3}{\Delta x_1 / 2} + \frac{\pi \cdot D_{3n} \cdot \alpha_{31}}{A_3 \cdot \rho_3 \cdot c_{p3}} \right) \cdot \theta_{31} - \frac{\varpi_3}{\Delta x_1 / 2} \cdot \theta_3 + \\ &+ \frac{\pi \cdot D_{3n} \cdot \alpha_{31}}{A_3 \cdot \rho_3 \cdot c_{p3}} \cdot \theta_{cm.21}. \end{aligned} \quad (22)$$

Для теплоносителей третьего и четвертого контуров при $i=1$ имеем:

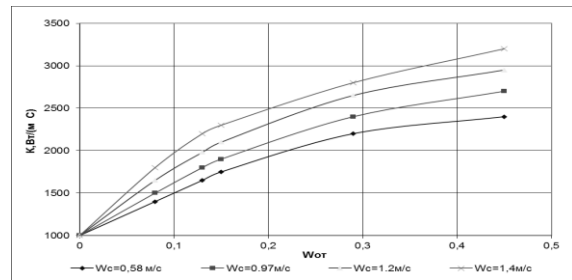


Рис. 4. Зависимости коэффициентов теплопередачи в теплообменнике змеевикового типа при изменении расхода и скорости теплоносителя отопительного контура

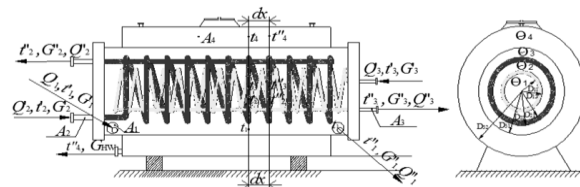


Рис. 3. Боковое и лицевое сечения МТАБ змеевикового типа

$$\begin{aligned} \frac{d\theta_{31}}{d\tau} &= - \left(\frac{\varpi_3}{\Delta x_1 / 2} + \frac{\pi \cdot D_{3e} \cdot \alpha_{31}}{A_3 \cdot \rho_3 \cdot c_{p3}} \right) \cdot \theta_{31} + \\ &+ \frac{\varpi_3}{\Delta x_1 / 2} \cdot \theta_3 + \frac{\pi \cdot D_{3e} \cdot \alpha_{31}}{A_3 \cdot \rho_3 \cdot c_{p3}} \cdot \theta_{cm.31}; \\ \frac{d\theta_{cm.31}}{d\tau} &= - \frac{\alpha_{31} + \alpha_{32}}{x_{cm.3} \cdot \rho_{cm.3} \cdot c_{cm.3}} \cdot \theta_{cm.31} + \\ &+ \frac{\alpha_{31}}{x_{cm.3} \cdot \rho_{cm.3} \cdot c_{cm.3}} \cdot \theta_{31} + \frac{\alpha_{32}}{x_{cm.3} \cdot \rho_{cm.3} \cdot c_{cm.3}} \cdot \theta_{41}; \\ \frac{d\theta_{41}}{d\tau} &= - \left(\frac{\varpi_4}{\Delta x_i} + \frac{\pi \cdot D_{3n} \cdot \alpha_{41}}{A_4 \cdot \rho_4 \cdot c_{p4}} \right) \cdot \theta_{41} - \\ &- \frac{\varpi_4}{\Delta x_1 / 2} \cdot \theta_4 + \frac{\pi \cdot D_{3n} \cdot \alpha_{41}}{A_4 \cdot \rho_4 \cdot c_{p4}} \cdot \theta_{cm.31}. \end{aligned} \quad (23)$$

Для противотока теплоносителей Q_1 и Q_2 ; Q_1 и Q_3 ; Q_3 и Q_4 – конечным есть n -й шаг.

Имитационное моделирование на основе данной динамической модели МТАБ позволяет определить оптимальные значения таких параметров, как скорость движения теплоносителей, время установления заданной температуры в различных контурах МТАБ и др. Для этого необходимо решать систему (21) – (23) известными численными методами при вариации начальных и конечных значений температур, скорости перекачки воды насосом, расхода теплоносителей и др. В частности, на рис.4 приведены зависимости коэффициентов теплопередачи в теплообменнике змеевикового типа при изменении расхода и скорости теплоносителя отопительного контура.

Кроме возможности численного решения дифференциальных уравнений в частных производных с фиксированием значений исследуемых

параметров, система SolidWorks дозволяє графічно зобразити процеси змішування і передвиження теплообмінних мас води в кожному контурі, так і в БА. Так, на рис.5 графічно представлено фрагмент процесу руху теплоносія в контурі системи гарячого

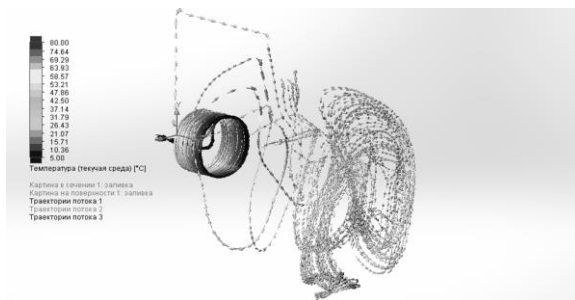


Рис. 5. Моделирование движения теплоносителя в контуре системы горячего водоснабжения

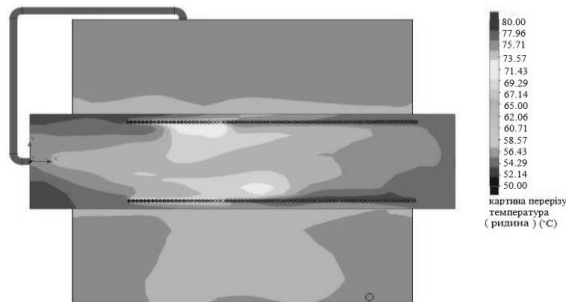


Рис. 6. Фрагмент моделирования процесса теплообмена в МТАБ

водоснабження МТАБ, а на рис.6 фрагмент моделювання процесу теплообміну в МТАБ.

Слід відзначити, що наявність у проєктувальників SolidWorks суттєво спрощає задачу імітаційного моделювання, дозволяючи

моделювати потоки рідин з урахуванням геометрії МТАБ, властивостей оточуючого середовища та умов протікання теплообмінних процесів.

Висновки

Наведено практичні рекомендації щодо вибору оптимальних конструктивних параметрів МТАБ в процесі проєктування АСТС та їх уточнення в процесі експлуатації, визначення на основі імітаційного моделювання оптимальних температурних режимів в контурах опалення та подачі гарячої води.

ЛІТЕРАТУРА

1. Патент № 81831, Україна, МПК F24D 11/00/ Багатоконтурний теплообмінний апарат змішувального типу для незалежних систем опалення та гарячого водопостачання / А.О. Олексюк, І.Г. Шитикова.- Донбаська нац.акад.буд. та арх.; заявл. 04.02.2013; опубл. 10.07.2013, Бюл. №13.
2. Міхєєв М. А. Основи теплопередачі/ М.А.Міхєєв, І.М. Міхєєва . – Навчальний посібник. – М: Вища школа, 1977.–336 с..
3. Фельдман Л.П., Петренко А.І., Дмитрієва О.А. Чисельні методи в інформатиці. К.: Видавнична група ВНУ, 2006. – 480 с.
4. Трофимчук А.Н. Методи суперпозиції, власних функцій і ортогональних многочленів в граничних задачах теорії еластичності та акустики / Гомилко А.М., Савицький О.А., Трофимчук А.Н. – Київ: Наукова Думка, 2016. – 435 с.
5. Олексюк А. А. Динамічні характеристики підогрівально-акумуляторних установок / А. А. Олексюк, А. М. Трофимчук, І. Г. Шитикова // Обчислювальна і прикладна математика. – 2016. – №1(121). – С. 78-88.

Безпека даних в хмарних середовищах

Пирожков Олексій Юрійович
КПІ ім. Ігоря Сікорського
Київ, Україна
a.y.pirozhkov@gmail.com

Савчук Олена Володимирівна
КПІ ім. Ігоря Сікорського
Київ, Україна
savchuk_11@ukr.net

Анотація. Робота присвячена концепції забезпечення безпеки та конфіденційності даних в хмарних обчисленнях, використовуючи підхід орієнтації на безпеку даних. Приводиться спосіб контролю доступу до ресурсів хмарного середовища.

Ключові слова: безпека хмарних обчислень, безпека даних, орієнтація на безпеку даних.

ВСТУП

У хмарних обчисленнях дані користувачів, в основному, зберігаються у віртуальних сховищах хмарних інфраструктур. В таких публічних моделях як “програма як послуга” (SaaS) та “робоче місце як сервіс” (DaaS) користувачі володіють лише даними, які знаходяться на зберіганні [1, 2]. Все обладнання та програмне забезпечення, залучене до зберігання та обробки інформації, знаходиться у власності сервіс провайдерів. В інших моделях, таких як публічні IaaS і PaaS моделі, користувач має доступ до обробки даних та до програмного забезпечення, при цьому доступу до апаратного забезпечення немає [3].

Відповідно, з точки зору користувача хмарного сервісу, найбільш цінним активом в хмарному середовищі є його дані, особливо ті дані, що містять інформацію делікатного характеру і вимагають особливого ставлення, а саме: дані урядового характеру, охорони здоров'я та фінансового спектру. Переваги, що надаються за використання хмарних обчислень, дають користувачам, що раніше розміщували делікатні дані на своїх ПК, більш привабливі перспективи до аутсорсингу своїх даних в хмару. Як і будь-які інші послуги в мережі Інтернет, хмарні сервіси також зазнають атак на системи безпеки. Компрометація доступності хмарних послуг, як правило, призводить до короткострокових ефектів і пошкодження можуть бути відновлені. Компрометація конфіденційності та приватності даних споживачів хмарних послуг, може привести в свою чергу до довгострокових ефектів і будь-які втрати можуть бути достатньо важким для відновлення.

Внутрішні ризики можуть бути від користувачів-зловмисників, які використовують той самий хмарний сервіс, і пом'якшення ризиків, в такому випадку, залежить повністю від хмарного провайдера і виходить з-під контролю власника даних. З точки зору власників даних, хмарне середовище невидиме і власник даних не впевнений в тому, як його/її дані захищені від ризиків, пов'язаних з безпекою [4]. Отже, власники даних стурбовані безпекою та приватністю їх даних і мають бажання зберегти свої дані в безпеці, навіть від сервіс провайдера хмарних послуг. Крім того, вони надають перевагу власноруч управляти політикою безпеки своїх даних в безпечному режимі, як ніби ці дані зберігалися на їх ПК.

КОНЦЕПЦІЯ ОРІЄНТОВАНОГО НА БЕЗПЕКУ ДАНИХ ПІДХОДУ

В даному пункті визначена концепція орієнтованого на безпеку даних відходу (care data). Ця концепція бере за основу огляд попередніх дослідницьких робіт по концепціям орієнтації на безпеку даних (ОБД) стосовно моделі хмари [5, 6]. В даній концепції безпека даних забезпечена перш, ніж покинути довірений домен власника даних. А параметри безпеки приєднані до даних як частина їх метаданих. В ідеальній ситуації, лише облікові дані, які використовуються для відміни або доступу за допомогою захисту, залишаються поза даними і власником даних та авторизованих користувачів відповідно до їх прав доступу. Будь-які інші необхідні параметри безпеки мають бути додані до даних.

Наступний список виділяє критерії, яким має відповідати забезпечене рішення безпеки, на основі ОБД підходу:

1. Кожен набір даних – це автономний контейнер, що здатен самостійно описувати себе та захищати. Таким чином, вимоги безпеки для кожного набору даних та властивостей забезпечуються всередині нього та не залежать від об'єктів за межами набору даних, за винятком деяких основних процесів обробки даних.
2. Захист даних не залежить від сервіс-провайдера або довіреної третьої особи.
3. Лише власник даних несе відповідальність за створення і управління цими вимогами безпеки та характеристики для кожного набору даних з моменту його створення до кінця життєвого циклу набору даних.
4. Всі операції, пов'язані з доступом до захищених даних, встановлюються уповноваженими користувачами, забезпечуються дотриманням політики безпеки даних та виконуються без компрометації приватності або порушення конфіденційності користувача.

DATA-ОРІЄНТОВАНЕ РІШЕННЯ

Data-орієнтоване рішення призначене для досягнення наступного переліку бажаних вимог до застосунків в середовищі хмарних обчислень [7]:

- дані зашифровані та доступні лише для авторизованих користувачів;
- дані доступні для пошуку без загрози для їх конфіденційності;
- дані відповідають вимогам самозахисту та необхідним параметрам безпеки;
- параметри контролю доступу приховані від провайдерів хмарних серверів та інших користувачів;
- провайдер серверу не знає кількості або особистості користувачів, що авторизовані та мають право доступу до даних;

– несанкціоновані суб'єкти, в тому числі постачальники послуг, не можуть отримати доступ до даних або інформацію про дії, що проводяться на даних авторизованими користувачами;

– дані містять всю необхідну інформацію для перевірки їх цілісності та достовірності для авторизованих користувачів;

– взаємодія між власниками даних та авторизованими користувачами має бути мінімальною, особливо щодо ключових цілей управління.

Ці вимоги визначені набором модулів, кожен з яких точно реалізує принаймні одну із зазначених вище вимог. Всі параметри, необхідні для досягнення функцій безпеки, прикріплюються до файлу даних і в результаті формують орієнтований на безпеку даних файл з відповідною назвою.

КОНТРОЛЬ ДОСТУПУ

У даній роботі запропоноване рішення використовує китайську теорему о залишках [8] і реалізує криптосистему з публічним ключем для безпечного обміну даними, що зберігаються в середовищі хмарних обчислень серед авторизованих користувачів. Ці дані, які іноді згадуються як ресурси, або файли, захищені симетричним методом шифрування із секретним ключем K_s . У даній роботі, ресурсом може бути набір даних або файл, який містить дані будь-якого типу, в тому числі текст, аудіо, зображення або відео. Для поширення секретного ключу для авторизованих користувачів, його зашифровано з використанням методу публічного шифрування відкритого ключа користувача. Шифрування також включає в себе інше значення C_r , яка буде використовуватися в якості відповіді на запит сервера, коли користувач робить запит щодо доступу до ресурсу. Лише авторизовані користувачі, які мають відповідний приватний ключ, можуть розшифрувати шифротекст C_r та K_r для отримання C_r , щоб отримати доступ до ресурсу r . Параметри C_r та K_s зчеплені для формування $C_r || K_s$ та розглядаються як єдине значення [9]. Це значення шифрується за допомогою публічного ключа $E_{K_{pubi}}$ для кожного авторизованого користувача u_i . В результаті отримуємо шифротекст ($E_{K_{pubi}}(C_r || K_s)$) для цього користувача u_i , де $i=1, 2, 3, \dots, k$, та k - кількість авторизованих користувачів, що мають доступ до ресурсу r .

Секретний ключ K_s , а також значення C_r є унікальними для кожного файлу, навіть якщо він використовується одним користувачем для різних файлів. Якщо значення для одного файлу було скомпрометовано, інші файли залишаються в безпеці. Секретне значення C_r використовується авторизованим користувачем, щоб показати серверу, що він або вона має право отримати доступ до ресурсу r (тобто файлу). Власник даних надійно прикріплює секретне значення C_r до файлу, а також надсилає всередині нього шифротекст X_r до сервера. Лише авторизовані користувачі можуть розрахувати секретне значення C_r з використанням спільного значення X_r . Таким чином, лише вони можуть знати і відкрити C_r серверу і довести своє право доступу до цього конкретного файлу. Таким чином, якщо один C_r

для конкретного файлу скомпрометований, ніякі інші файли не будуть під загрозою. Крім того, ця функція корисна, якщо власник даних хоче змінити деталі щодо авторизованих користувачів для файлу. Наприклад, щоб додати нового авторизованого користувача, власнику даних лише необхідно змінити X_r для цього файлу. Оскільки параметр C_r може залишатися таким же самим, немає необхідності для повторного надсилання нового C_r до серверу. Це, в свою чергу, дозволить знизити ймовірність компрометації цього значення в процесі підтримки динамічного механізму оновлення списку авторизованих користувачів. Проте, при забороні користувачу доступу до файлу, до якого він чи вона вже мають доступ, значення C_r має бути змінене з міркувань безпеки.

Запропоноване рішення поєднує в собі контроль доступу та спільного використання ключа в одному механізмі з використанням китайської теореми о залишках. Хмарний сервер зберігає зашифровані дані з відповідним секретним C_r і секретним значенням X_r , що розраховується власником даних. Провайдер може прочитати секретне значення X_r , але не може дізнатися K_s , який було використано для шифрування даних. Лише авторизовані користувачі можуть виявити K_s від X_r . Провайдер також може зчитувати секретне значення C_r для кожного файлу. Проте, число або ідентичність користувачів, які можуть отримати доступ до файлу, навіть якщо вони вже отримали доступ до файлу, залишаються прихованими від провайдера.

ВИСНОВКИ

У даній роботі виконаний орієнтований на безпеку даних підхід для хмарних обчислень.

Запропоноване рішення контролю доступу дозволило вирішити поставлені задачі, а саме:

- шифрування даних, надання доступу лише для авторизованих користувачів;
- відповідність даних вимогам безпеки;
- прихованість параметрів контролю доступу та безпосередньо даних від провайдерів хмарних серверів та інших користувачів;
- забезпечення необхідної інформації для перевірки цілісності та достовірності даних для авторизованих користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Armbrust, M., Fox, A. & Griffith, R. Above the Clouds: A Berkeley View of Cloud Computing.// Electrical Engineering and Computer Sciences University of California at Berkeley, 2009, pp. 1-8.
2. Chou, D. C., & Chou, A. Y. . Software as a Service (SaaS) as an Outsourcing Mode // An Economic Analysis, 2007, pp. 386-391.
3. Buyya, R., Yeo, C., Venugopal, S., Broberg, J. & Brandic, I. Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 2009, 25, pp. 599- 616.
4. Mukhin, V.&Volokyta, A. Integrated safety mechanisms based on security risks minimization for the

distributed computer systems // I.J. Computer Network and Information Security, 2013, vol. 2, pp. 21-28.

5. I. Iankoulova, M. Daneva: Cloud Computing Security Requirements: a Systematic Review // Sixth International Conference on Research Challenges in Information Science, 2012, 7, pp., 2012.

6. B. Grobauer, T. Walloschek, and E. Stocker, Understanding Cloud Computing Vulnerabilities// Security & Privacy, IEEE, 2011, vol. 9, pp. 50-57.

7. T. Dillon, W. Chen, and E. Chang. Cloud Computing: Issues and Challenges, in Advanced Information Networking and Applications (AINA)// 24th IEEE International Conference, 2010, pp. 27-33.

8. Kaya, Kamer & Aydin Selçuk, Ali. Secret Sharing Extensions based on the Chinese Remainder Theorem // IACR Cryptology ePrint Archive. 2010. 96.

9. C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," Journal of Computer Security, 2011, vol. 19, pp. 367-397.

СИСТЕМИ КЕРУВАННЯ

CONTROL SYSTEMS

Оценка возможности создания дополнительной тяги для управления космическими аппаратами на основе использования постоянных магнитов

Лапханов Э.А.

Институт технической механики

Национальная академия наук Украины и Государственное космическое агентство Украины

Днепр, Украина

ernando@i.ua

Аннотация. Статья посвящена оценке возможности создания дополнительной тяги для управления космическими аппаратами с использованием устройств на постоянных магнитах. Основное внимание уделяется расчетам силы тяги при взаимодействии магнитосферы, создаваемой постоянными магнитами с потоком заряженных частиц ионосферы и солнечного ветра.

Ключевые слова: космический аппарат, постоянные магниты, тяга, магнитное поле Земли, поток заряженных частиц.

ВВЕДЕНИЕ

Разработка космических двигательных установок является одним из важных направлений в развитии аэрокосмической техники. Одной из главных задач при создании новых двигательных систем является минимизация потребления топлива и бортовой энергии с сохранением или увеличением основных тяговых характеристик двигателя.

В данной работе исследована возможность использования устройства с постоянными магнитами (ПМ) для создания дополнительной тяги с целью управления орбитальным движением космических аппаратов (КА). За основу исследований была принята концепция «магнитного паруса» Японского университета Киото[1]. При анализе материалов описания проведения опытов и расчётов учеными Японского Аэрокосмического Исследовательского Агентства Икохом Фунаки и Хироши Ямакавой по применению электромагнитного паруса в межпланетном пространстве, было предложено обоснование использования ПМ с целью значительного уменьшения затрат бортовой энергии[2]. Так, было вычислено, что при использовании электромагнитного паруса, для получения тяги в 1 Н необходимо запитать токопроводящее кольцо из сверхпроводника радиусом 1 км. В свою очередь мощность источника энергии для осуществления проекта электромагнитного паруса составила порядка МВт электроэнергии, что, на сегодняшний день, делает данную задачу практически не выполнимой.

ПРИНЦИП ДЕЙСТВИЯ УСТРОЙСТВА

Принцип действия устройства с использованием ПМ заключается в креплении специально экранированных со створами контейнеров на КА и помещению в них ПМ. Открывая и закрывая створы экранированных контейнеров можно в нужный момент времени генерировать искусственную магнитосферу и создавать тягу. Контейнеры предлагается делать из специальных многослойных экранированных материалов. Управлять створами экранированных контейнеров предлагается с помощью шаговых двигателей (ШД). Импульсные ШД значительно меньше потребляют электроэнергии, нежели электромагниты, что позволяет уменьшить расход бортовой энергии.

Как известно, концепция «магнитного паруса» базируется на создании дополнительной тяги, при взаимодействии потока заряженных частиц солнечного ветра с искусственно создаваемой магнитосферой вокруг КА. Следует учесть, что в концепции, предложенной японцами, радиус искусственно генерируемой магнитосферы с помощью электромагнитного кольца составляет около 72 км. При проведении расчётов по использованию ПМ для отдельных средних и малых КА, радиус генерируемой магнитосферы в межпланетном пространстве составил всего на всего десятки м, а полученная тяга около нескольких десятков нН.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Исходя из результатов данных расчётов, применение ПМ в межпланетном пространстве для КА, класса «средние» и меньше, является нецелесообразным, поскольку не обеспечивает необходимой для управления КА тяги. Для получения необходимой площади магнитосферы, при использовании ПМ в межпланетном пространстве необходимо конструировать систему из множества ПМ, расположенных друг от друга на расстоянии радиусов генерируемых ими магнитосфер. Тогда, с учетом принципа суперпозиции полей можно получить магнитосферу для создания необходимой тяги, при взаимодействии с набегающим потоком

частиц солнечного ветра. Такая конструкция нуждается в сборке сложных конструкций и ферм для крепления ПМ на орбите или в открытом космосе, что также вызывает большие затруднения при практической реализации.

Однако, в рамках концепции «лунной деревни», такая система может быть использована для управления окололунной многоэлементной станцией, поскольку не требует значительных затрат бортовой энергии[3].

Но, в основном, предлагается использовать устройство с ПМ на низких околоземных орбитах в слоях ионосферы, высотой 600 – 1000 км, где концентрация заряженных частиц на несколько порядков выше чем у потока солнечного ветра. Как известно, заряженные частицы ионосферы не имеют главного вектора движения, в отличии от динамического напора солнечного ветра. Поэтому, при рассмотрении движения КА в слоях ионосферы заряженные частицы будут выступать, как набегающий поток, противоположно направленный главному вектору движения КА. В таком случае набегающий поток заряженных частиц будет создавать тормозящий эффект. Также, следует учесть, что при движении КА в окрестности магнитного поля Земли (МПЗ), согласно дипольной природы ПМ, на ПМ будет действовать сила, направленная в сторону увеличения градиента напряженности внешнего поля, которым выступает МПЗ[4]. Таким образом, на околоземных орбитах использование ПМ оказывает двойной эффект: дополнительная тяга при взаимодействии с набегающим потоком заряженных частиц и тяга при взаимодействии с МПЗ. С помощью расчётов было установлено, что значение тяги для «средних» и «малых» КА, при массе ПМ в несколько кг и остаточной магнитной индукции в 1,4 Тл, на низких околоземных орбитах, составило от 16 мН на высоте 600 км, до 2 мН на высоте 1000 км. Из полученных результатов расчета можно сделать вывод, что значение тяги, получаемой с помощью ПМ сопоставимо с тягой, которую дают микро реактивные двигатели (МРД), и ее достаточно для совершения

управления траекторией полета КА. Также, можно использовать устройство с ПМ и как систему увода с орбиты отработанных КА без затрат бортовой энергии, что также является преимуществом применения данного устройства.

Выводы

Исходя из полученных результатов, использование устройств с ПМ для управление орбитальным движением КА на высотах от 600 до 1000 км является целесообразным и не нуждается в больших затратах бортовой энергии. Также, после окончания термина эксплуатации КА может быть уведен с орбиты в случае действия постоянных тормозящих сил, которые получаем путем полного открытия экранированного контейнера и взаимодействия ПМ с набегающим потоком заряженных частиц ионосферы и МПЗ.

ЛИТЕРАТУРА

1. Yasumasa Ashida. Study on propulsive characteristics of magnetic sail and magneto plasma sail by plasma particle simulations: PhD dissertation. – Kyoto, 2014 – p. 7-15. URL: repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/185206/2/dkogk03813.pdf (дата обращения 05.11.2017).
2. Ikkoh Funaki. Solar Wind Sails, Exploring the Solar Wind/ Ikkoh Funaki, Hiroshi Yamakawa//InTech. – 2012. - №65, URL: <http://www.intechopen.com/books/exploring-the-solar-wind/solar-wind-sails> (дата обращения 03.11.2017).
3. Ссылка на интернет ресурс: <https://hi-news.ru/space/kak-novaya-lunnaya-stanciya-pomozhet-nam-dobratsya-do-marsa-i-dalshe.html> (дата обращения 04.11.2017).
4. Umair Ahsun. Dynamics and Control of Electromagnetic Satellite Formation: PhD dissertation. – Massachusetts, 2007 – p. 20 – 40.

Проектування відеокадрів людино-машинного інтерфейсу АСУ ТП

Юрчук Леонід Юрійович
КПІ ім. Ігоря Сікорського
Київ, Україна
leonidyu0@gmail.com

Бублінський Станіслав Михайлович
КПІ ім. Ігоря Сікорського
Київ, Україна
bublinsky.s.m@gmail.com

Анотація. Розглядаються питання розробки відеокадрів мнемосхем людино-машинного інтерфейсу систем управління технологічними процесами відповідно до вимог та рекомендацій міжнародного стандарту ANSI/ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems

Ключові слова: людино-машинний інтерфейс, відеокадр, дисплей, людський фактор, ергономіка

Аналіз причин багатьох технологічних аварій у ряді випадків визначає недостатню якість людино-машинного інтерфейсу. Одним з важливіших його елементів є візуальні засоби відображення інформації які раніше були приладовими панелями, а сьогодні мнемосхемами на екранах моніторів чи інших електронних засобів. Міжнародна спільнота фахівців з автоматизації приділяє багато уваги ефективності роботи операторів систем управління, випускаючи різні матеріали [1]. У 2015 році був випущений черговий реліз стандарту ANSI/ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems [2], який в деякій частині проголошується як обов'язковий.

На жаль в Україні відсутні аналогічні стандарти і в умовах, коли ми йдемо в напрямку міжнародного співробітництва необхідно знати відповідну нормативну базу.

У роботі [3] розглянута загальна структура стандарту, термінологія, етапи життєвого циклу. Ця робота присвячена розгляду питань урахування людського фактору при проектуванні відеокадрів мнемосхем відповідно до вимог розділу 5 стандарту – «Людський фактор - інжиніринг і ергономіка (Human Factors Engineering & Ergonomics)»

Як декларують розробники стандарту мета розділу 5 полягає у формуванні набору принципів, які дозволяють розробнику ЛМІ зрозуміти основні засади ключових проектних рішень.

Ефективний ЛМІ для забезпечення управління технологічними процесами може бути спроектовано тільки на основі врахування людського фактору (ЛФ - Human Factors Engineering - HFE) - когнітивних і сенсорних можливостей і обмежень користувачів ЛМІ. Конструкція ЛМІ має звести до мінімуму вплив другорядних завдань (наприклад, навігацію по

відеокадрам (мнемосхемам, дисплеям) в ЛМІ), які можуть відволікати оператора ЛМІ від виконання своїх основних завдань.

При проектуванні ЛМІ повинні враховуватися загальні принципи:

- спосіб функціонування ЛМІ повинен бути інтуїтивно зрозумілим користувачеві;
 - бути ефективним інструментом для безпечного і ефективного управління процесом;
 - допомагати операторам визначати пріоритети реагування на основні або декілька одночасних системних збоїв;
 - НМІ повинен бути спроектований для підтримки завдань, пов'язаних з усіма зазвичай очікуваними режимами роботи, в тому числі в аномальних умовах (наприклад, обробка сигналів тривоги);
 - допомагати в ранньому виявленні, діагностиці та правильній реакції на нештатні ситуації;
 - повинен надавати інформацію або засоби управління, що відповідають завданню;
 - інформація повинна бути представлена в формах або форматах, які відповідають цілям користувача (наприклад., якщо оператор користується представленням інформації у метрах, сантиметрах, футах або дюймах, то НМІ повинен відображати інформацію в цьому ж форматі, а не у відсотках або частинах).
 - допоміжна інформація повинна бути легко доступна для користувача (наприклад, процедури, які використовуються для запуску частини обладнання або періодичного процесу, процедури реагування на сигнали тривоги або керівництв користувача ЛМІ та ін.).
 - відмова відеокадру (дисплея) або елементів на ньому відразу повинні бути очевидні для оператора;
 - потреби оператора засновані на вимогах завдань повинні мати пріоритет над інтересами всіх інших користувачів.
 - термінологія, яка використовується в відеокадрах (дисплеях) НМІ повинна бути сумісна з загальними описами користувача.
- Оскільки існують різні користувачі ЛМІ, що мають різні особисті та функціональні потреби, то кожному з їхніх потреб слід враховувати при проектуванні.

Врахування ЛФ повинно бути закладено ще на рівні розробки концепції (філософії), при формуванні функціональних вимог, керівництва по стилю, проектній документації.

Для ефективного управління процесом оператор повинен усвідомлювати ситуацію тобто:

- а) бути в курсі того, що відбувається в цьому процесі взагалі;
- б) розуміння стану процесу в даний час;
- в) розуміння ймовірного стану процесу в майбутньому.

Коли процес працює, як і очікувалося, відеокадр повинен мати мінімум сенсорних стимулів. Коли процес відхиляється від очікувань, то ЛМІ повинен забезпечувати візуальні та/або звукові сигнали що відповідають важливості ситуації.

Недостатня інформованість ситуації була визначена в якості одного з головних чинників нещасних випадків, приписуваних людським помилкам. Закріплення на частині інформації, виключення іншої може обмежити розуміння ситуації. Фіксування фокуса уваги оператора на одному завданні може привести до виключення іншої потенційно більш важливої інформації (часто втрачається в обсязі представлених даних), призводить до зниження усвідомлення ситуації.

Проект ЛМІ повинен включати в себе принципи ергономіки, які засновані на сенсорних обмеженнях користувача (наприклад, зорових, слухових) і розуміння очікуваних функціональних вимог користувачів. Важливо враховувати пороги і верхні межі сенсорних систем користувача, а також можливі загальні недоліки сенсорної системи (наприклад, колірна сліпота, втрата слуху, вікове знецінення зору). Необхідно враховувати робочі умови. Наприклад, при проектуванні ЛМІ, який буде працювати в рамках декількох умов освітлення, система і відеокадри повинні бути розроблені на прийнятному рівні продуктивності для всіх очікуваних умов. Яскравість екранів повинна відповідати умовам зовнішнього середовища з метою запобігання стомлення очей операторів, уникаючи використання надмірного контрасту.

Приблизно 10 % чоловіків і 1 % жінок мають проблеми з сприйняттям кольорів. Якщо можливе використання операторів з вадами зору, то необхідно при проектуванні це враховувати. Найбільш поширені типи колірної сліпоти включають червоний-зелений, зелено-жовтий і біло-блакитний. Контрастність та яскравість можуть використовуватися в контексті робочого середовища для компенсації колірної сліпоти.

Недоліки, пов'язані з віком, включають:

- труднощі з сприйняттям інформації на ближньому полі;
- зниження здатності швидко переорієнтувати між ближнім і дальнім об'єктами;

- хроматичні спотворення, що можуть привести до зміни сприйняття кольорів.

Відповідно до філософії сигналізації на об'єкті і рекомендацій стандарту ANSI/ISA-18.2-2009, кольори, що використовуються для представлення сигналізації повинні бути зарезервовані і не повинні використовуватися для будь-яких інших цілей, з тим щоб посилити їх пізнавальний сенс і швидкість реакції оператора на сигнали тривоги.

Не слід покладатися на колір як єдиного на показника важливості. Кольорове кодування повинне бути додатковим для інших засобів представлення інформації, включаючи, але не обмежуючись ними, у формі, тексті, яскравості, розмірі і текстурі. Гамма кольорів, що використовуються у відеокадрі повинна відображати важливість інформації, що представляється. Найбільш помітні кольори повинні використовуватися для інформації, яка є найбільш важливою.

Кольорові градієнти не повинні використовуватися для статичних або не динамічних елементів на відеокадрі, але може бути використані, щоб виділити динамічний елемент.

Колір слід використовувати консервативно і послідовно для позначення інформації по всьому НМІ.

Фон відеокадру повинен бути ненасиченим або мати нейтральний колір (наприклад, світло-сірий), з тим щоб обмежити хроматичні спотворення і забезпечити ефективно сприйняття інформації, яка відображається. Слід уникати використання кольорів фону, які можуть привести до надмірної контрастності (наприклад, чорний).

Обраний колір фону повинен забезпечити прийнятний і достатній контраст в очікуваних умовах освітленості. Корисно перевірити кольорове зображення, роблячи його в відтінки сірого, щоб гарантувати, що всі комбінації елементів мають достатній контраст, щоб бути виявлені користувачем з дефіцитом кольору.

Щільність інформації на відеокадрі (дисплеї) повинна бути заснована на функції або меті відеокадру, з урахуванням меж сприйняття людини. На відеокадрах, що стосуються завдань, які повинні бути швидко оброблені оператором, слід уникати непотрібної інформації. Елементи індикації повинні бути розміщені послідовно і вирівнюватися в просторі, щоб уникнути плутанини.

Із-за меж сприйняття лише обмежена кількість кольорів може бути ефективно використана на відеокадрах. Тому можуть бути використані візуальні динамічні методи кодування, такі як рух, блимання, умовна видимість щоб привернути увагу оператора до конкретної інформації.

Надмірний або постійний рух, блимання може бути відволікаючим. Основні принципи управління анімацією:

- моргання повинно бути зарезервовано для елементів таких як символи або межі для невизнаних оператором сигналів тривоги;
- текст і цифри самі по собі не мають рухатися або блимати, тому що змінна поява і зникнення тексту або цифр зробить читання їх важким;
- жодна частина відеокадру не повинна мигати або блимати, якщо ця дія не потрібно оператору;
- засоби зупинки миготіння або блимання повинні бути доступні для оператора.

Використання візуальної динаміки повинно бути зарезервовано для виділення аномальних ситуацій або ситуацій, що вимагають уваги оператора (наприклад, сигналу тривоги).

Слухові попередження повинні забезпечити надлишкову інформацію в поєднанні з візуальними попередженнями.

Слухові сигнали повинні бути такими, щоб вони не злякали слухачів, значно перебільшуючи загальний рівень шуму, або заважали обміну даними між користувачами. Місце джерела сигналу, його інтенсивність, тривалість повинні бути сумісними з параметрами навколишнього середовища в області сигналу. Значення кожного слухового сигналу повинні бути чіткими й однозначними.

Повинен бути передбачений метод відключення

звукової індикації, коли вона буде визначена оператором.

Когнітивні процеси, які трансформують, зменшують, зберігають, відновлюють і використовують сенсорні данні введення мають важливе значення для людської діяльності. Продуктивність користувача і когнітивні процеси, що лежать в її основі, залежать від рівня навантаження, усвідомлення ситуації і складності завдань, можуть бути оптимізовані за рахунок конструкції ЛМІ.

ЛІТЕРАТУРА

1. Hollifield, B. The High Performance HMI Handbook./ Hollifield, B. Oliver, D., Habibi, E., & Nimmo, I. // – PAS,2008 – 218 p.

2. Стандарт ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://www.isa.org/store/ansi/isa-10101-2015,-human-machine-interfaces-for-process-automation-systems/41921133>

3. Юрчук Л.Ю. Міжнародні нормативні засади людино-машинного інтерфейсу систем управління технологічними процесами. - Summer InfoCom 2017: Матеріали IV Міжнародної науково-практичної конференції, м. Київ, 1-2 червня 2017 р – К.: Вид-во “Інжиніринг”, 2017. –с. 66-68

**ТЕХНОЛОГІЇ
ПРОГРАМУВАННЯ**

**PROGRAMMING
TECHNOLOGIES**

Высокопроизводительное производство матриц посредством Android NDK и JNI

Ашур И.З.

КПИ им. Игоря Сикорского
Киев, Украина
ilyaachour@gmail.com

Дорошенко А.Ю.

КПИ им. Игоря Сикорского
Киев, Украина
a-y-doroshenko@ukr.net

Аннотация. Работа представляет описание высокопроизводительного подхода для производства матриц посредством Android NDK и JNI. Была произведена оценка производительности подхода и его альтернатив, рассмотрены все используемые техники и особенности.

Ключевые слова: Android NDK, JNI, аффинные преобразования, OpenGL ES, Java

ВВЕДЕНИЕ

Эффективное производство матриц – одна из актуальных проблем разработки приложений для мобильных платформ, использующих высокопроизводительную двумерную компьютерную графику.

Такие приложения разрабатываются для игровой индустрии, программного обеспечения визуализации, моделирования в части рендеринга визуальных представлений элементов моделей, визуального интерактивного взаимодействия.

Одним из аспектов применения результатов данных вычислений является производство матриц вида, модели и проекции.

Матричное представление используется, в частности, для записи аффинных преобразований в компьютерной графике.

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Матрица вида описывает представление аффинного преобразования, например, сжатие, растяжение, поворот, параллельный перенос, отражение и другие частные и общие случаи трансформаций.

Пространство вида представляет результат преобразования мировых координат в координаты видимого пространства. Видимое пространство в данном случае определяется совокупностью преобразований сдвигов и вращений сцены. Эти комбинированные преобразования представляются матрицей вида.

Современные системы рендеринга двумерной графики также оперируют понятием пространства отсечения. Пространство отсечения определяет видимые области сцены и, соответственно, вершины.

Общий процесс применений данных преобразований приведен на рис. 1 [1].

Детали применения данных трансформаций и математическая их составляющая не является предметом обсуждения данной статьи и упомянуты с целью предоставления общей информации о предметной области решаемой задачи.

Проводимое исследование нацелено на разработку, создание и анализ производительности подхода

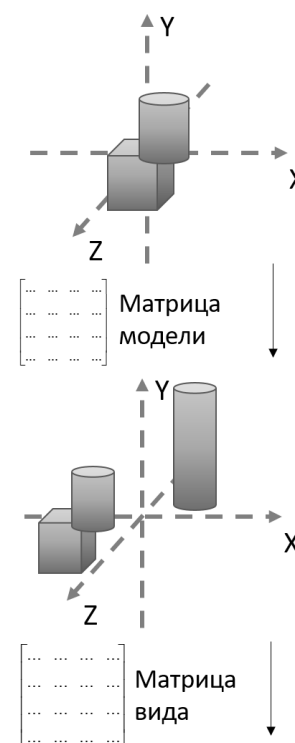
произведения матриц посредством Android NDK и JNI.

АНАЛИЗ И РЕАЛИЗАЦИЯ РЕШЕНИЙ

Отталкиваясь от введенных ограничений предметной области в виде матриц аффинных преобразований, в дальнейшем будем оперировать матрицами размером 4×4.

Классическая, наивная реализация производства матриц предусматривает использование средств JDK с применением двух вложенных циклов.

Для оценки производительности подхода и его наивных аналогов были созданы специальные программы-бенчмарки с учетом всех «best practices». Существует множество оптимизаций, которые JVM или аппаратное обеспечение могут применить к конкретному компоненту во время его изолированного тестирования. Непродуманные тесты производительности могут дать оптимистичные результаты, отнюдь не совпадающие с реальными. Именно тестирование конкретного компонента в составе рабочего, практического приложения с применением таких техник, как JVM warm up («разогрев» виртуальной машины) может дать достоверные результаты [2].



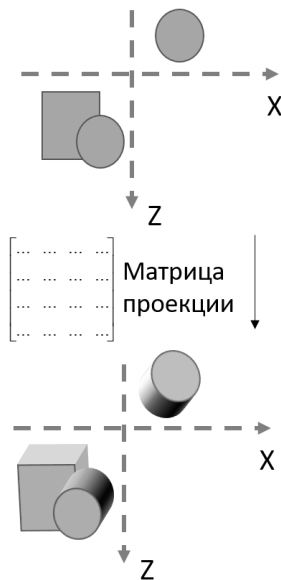


Рис. 1. Общая схема применения матриц

Далее будут приведены результаты тестов производительности для итераций разработанного подхода и его наивных альтернатив с различающимся стеком используемых техник и особенностей, их сравнение. Под метрикой производительности подразумевается время, затраченное процессором на выполнение определенного количества матричных произведений.

Результаты тестирования затрат процессорного времени на произведение 1 миллиона матриц 4×4 в представлении одномерного массива на 16 элементов типа с плавающей запятой с использованием классической, наивной реализации средствами JDK на устройстве Google Pixel XL в 10 итераций представлены в табл. 1 и рис. 2.

Таблица 1

Итерация, №	Затраты процессорного времени, мс
1	3121
2	3141
3	3132
4	3155
5	3137
6	3127
7	3133
8	3115
9	3111
10	3116
Усредненный результат	3129

Результаты тестов для разного количества матриц, разного количества итераций, разных устройств и разной размерности матриц опускаются по причине практически прямой зависимости между приведенными метриками и производительностью вычислений.

Представленные результаты получены на устройстве с процессором Qualcomm MSM8996 Snapdragon 821 Quad-core (2x2.15 GHz Kryo & 2x1.6 GHz Kryo) [3].

Следует отметить, что тесты производительности производились на сборках приложения с использованием агрессивной оптимизации и обфускации исходного кода посредством инструмента ProGuard с семью проходами оптимизации, применяющих, в частности, арифметические

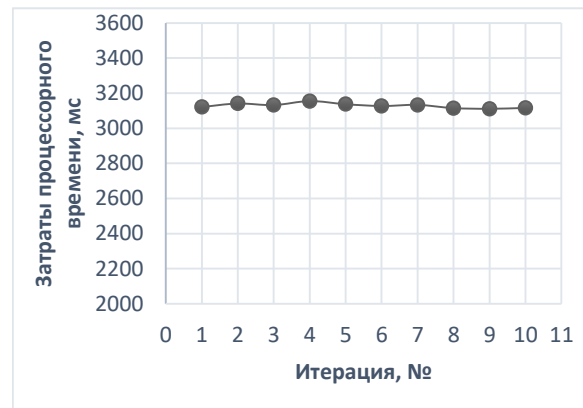


Рис. 2. График зависимости затрат процессорного времени от итерации тестирования для наивной реализации

оптимизации, финализацию классов, их членов, вертикальное и горизонтальное склеивание классов, приватизации полей, методов, применение статических методов, оптимизация количества параметров методов, встраивание константных параметров, встраивание констант возврата методов, встраивание вызовов коротких методов, встраивание уникальных методов, набор методик «reephole optimization» для арифметических операций, для ветвления кода, для строк, оптимизации приведения типов, чистка избыточных инструкций, исключений, блоков кода [4]. Тема оптимизаций производительности Java-кода на этапе компиляции проекта на текущий момент достаточно изучена и глубоко разработана, документирована известными авторитетными разработчиками и не нуждается в более подробном рассмотрении. Данные оптимизации применялись для получения «боевых» результатов тестирования производительности.

В составе Android SDK предоставляется набор классов-утилит для работы с матрицами. Их реализация, подобно далее предложенному подходу, также имплементирована на уровне NDK [5]. В целом, она полностью повторяет рассмотренный выше подход за исключением наличия целого ряда дополнительных защит и проверок, инструкций для высвобождения памяти.

Результаты тестов производительности встроенного решения по отработанному алгоритму тестирования на идентичном наборе исходных данных и при идентичном окружении аппаратной и программной среды представлены на табл. 2 и рис. 3:

Таблица 2

Итерация, №	Затраты процессорного времени, мс
1	6075
2	5997
3	5987
4	6008
5	6037

6	6023
7	6005
8	5994
9	5973
10	6053
Усредненный результат	6015

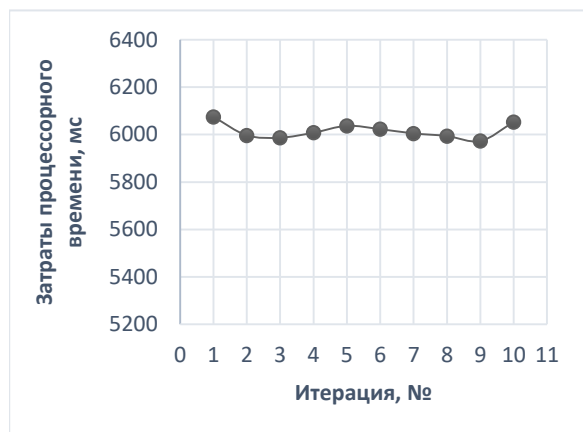


Рис. 3. График зависимости затрат процессорного времени от итерации тестирования для встроенной реализации

Исходя из приведенных результатов можно сделать вывод, что в данном случае нативная реализация данной задачи, поставляемая в составе пакета разработки Android, показывает результаты в среднем в 1,9 раз медленнее, чем наивная реализация того же алгоритма в пространстве JDK.

Возможно выделить две основные причины такой низкой производительности данной реализации. Первая – накладные затраты вызова нативных функций посредством JNI, т.н. «overhead». Вторая – наличие целого ряда дополнительных проверок входных данных и избыточного копирования данных во временные буферы для проведения вычислений.

Исходя из проведенных тестов, нацеленных на оценку затрат вызова JNI функций посредством Google Caliper [6] (рис. 4), можно сделать вывод, что на целевом аппаратном обеспечении вызов JNI функции может занимать перспективно время эквивалентное выполнению 5-30 java операций над примитивными типами.

```
Scenario{benchmark=Jni} 10.26 ns; σ=0.02 ns @ 10 trials
Scenario{benchmark=IntOperation} 0.48 ns; σ=0.02 ns @ 10 trials
Scenario{benchmark=LongOperation} 0.87 ns; σ=0.02 ns @ 10 trials

benchmark    ns linear runtime
Jni          10.265 =====
IntOperation 0.481 ==
LongOperation 0.873 ==
```

Рис. 4. Результаты оценки накладных затрат на вызов JNI функций

Таким образом, возможно предложить решение данной задачи убрав максимальное количество избыточных проверок и встроенных защит нативной имплементации, что позволительно на таком низком уровне выполнения, поскольку предусматривает применение в составе боевых, отлаженных и стабильных системах, где в принципе не допустимо поступление не валидных входных параметров для

вычислений, поскольку ответственность за такие проверки и препроцессинг в таких системах несут верхние слои абстракции и реализации системы.

Помимо этого, возможно оптимизировать представленный наивный алгоритм произведения матриц посредством развертывания циклов, введения ряда константных индексов, пакетных буферов матриц, указателей, операций, применения агрессивных уровней оптимизации компилятора нативного решения.

В результате применения предложенных изменений удалось достичь следующих показателей производительности произведения (табл. 3, рис. 4):

Таблица 3

Итерация, №	Затраты процессорного времени, мс
1	2488
2	2382
3	2326
4	2339
5	2292
6	2319
7	2352
8	2317
9	2410
10	2455
Усредненный результат	2368

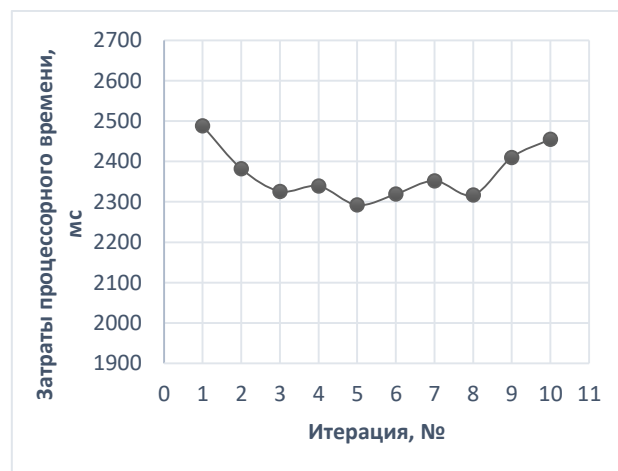


Рис. 4. График зависимости затрат процессорного времени от итерации тестирования для предложенного решения

ВЫВОДЫ

Таким образом, предложенное проприетарное решение показывает результаты в среднем в 1,3 раза лучшие в сравнении с наивным Java-решением и в 2,5 раза лучшие в сравнении с нативным решением из состава SDK.

В перспективе предлагается улучшить предложенное решение посредством реализации полноценного пакетного произведения буферов матриц, максимально сократив таким образом накладные затраты на взаимодействие с нативными функциями.

ЛИТЕРАТУРА

1. Article - World, View and Projection Transformation Matrices [Электронный ресурс] – Режим доступа до ресурсу:

http://www.codinglabs.net/article_world_view_project_ion_matrix.aspx.

2. Don't Get Caught In the Cold, Warm-up Your JVM [Электронный ресурс] / [D. Lion, A. Chiu, H. Sun та ін.]. – 2016. – Режим доступу до ресурсу: <http://www.eecg.toronto.edu/~yuan/papers/osdi16-hottub.pdf>.

3. Tech Specs [Электронный ресурс] – Режим доступу до ресурсу: https://store.google.com/us/product/pixel_phone_specs?hl=en-US.

4. Optimizations [Электронный ресурс] – Режим доступу до ресурсу: <https://www.guardsquare.com/en/proguard/manual/optimizations>.

5. `xref: /frameworks/base/core/jni/android/opengl/util.cpp` [Электронный ресурс] – Режим доступу до ресурсу: http://androidxref.com/8.0.0_r4/xref/frameworks/base/core/jni/android/opengl/util.cpp.

6. ProjectHome [Электронный ресурс] – Режим доступу до ресурсу: <https://github.com/google/caliper/wiki/ProjectHome>

**ОБРОБЛЕННЯ ІНФОРМАЦІЇ У
СКЛАДНИХ СИСТЕМАХ**

**INFORMATION PROCESSING IN
COMPLEX SYSTEMS**

Метод накопичувального впливу для аналізу когнітивних карт

Дмитренко Олег Олександрович

КПІ ім. Ігоря Сікорського
Київ, Україна
dmytrenko.o@gmail.com

Ланде Дмитро Володимирович

КПІ ім. Ігоря Сікорського
Київ, Україна
dwlande@gmail.com

Анотація. В роботі представлений новий метод аналізу когнітивних карт, так званий, метод накопичувального впливу. Показано, що він позбавлений недоліків, які містить, наприклад, один з методів, що широко використовується в наш час, – імпульсний метод, а саме: використовуючи запропонований метод, завжди можна визначити результуюче значення, не залежно від того, чи є стійким імпульсний процес. На відміну від імпульсного методу, метод накопичувального впливу не порушує масштабної інваріантності. В роботі також наведені приклади аналізу когнітивних карт, на яких продемонстровані переваги запропонованого методу.

Ключові слова: когнітивна карта, орієнтований граф, зважений орграф, матриця суміжності, матриця впливу, імпульсний метод, метод накопичувального впливу.

ВСТУП

Когнітивне моделювання є одним із нових напрямків сучасної теорії підтримки та прийняття рішень [1].

Когнітивна карта – це орієнтований граф, ребрам (а в деяких випадках і вузлам) якого можуть приписуватись певні вагові значення. Когнітивна карта, як і будь-який граф, описується матрицею суміжності W , елементами якої є w_{ij} – вагові значення ребер, що з'єднують відповідні вершини u_1, u_2, \dots, u_n . Вершинам когнітивної карти відповідають певні поняття (концепти), а ребрам (зв'язкам) – казуальні (причинно-наслідкові) зв'язки між відповідними концептами.

Введення ваг, що характеризують силу впливу, виявилось основним напрямком розвитку когнітивного підходу до аналізу ситуацій. Для аналізу добре структурованих ситуацій з кількісними параметрами використовуються вагові значення, а значення впливу за різними шляхами між двома вершинами підсумовуються.

Для аналізу нечітких когнітивних карт (НКК) найбільш відомим є підхід, запропонований Б. Коско [2], який також ввів поняття НКК (англ. FCM – fuzzy cognitive maps). Залежно від конкретного типу задач, що вирішуються, розглядаються різні модифікації НКК. Основою методів аналізу НКК є операції нечіткої математики [3].

ІМПУЛЬСНИЙ МЕТОД

Одним з методів кількісного аналізу когнітивних карт, що широко використовується в наш час, є імпульсний метод [4, 5]. В цьому методі вважається,

що кожна вершина u_i приймає значення $v_i(t)$, в дискретні моменти часу $t = 0, 1, 2, \dots$. Вага ребра має додатне значення $w_{ij} > 0$, якщо збільшення вагового значення концепта-причини призводить до збільшення вагового значення концепта-наслідка, і $w_{ij} < 0$ – в протилежному випадку. Якщо зв'язок між вершинами i та j відсутній, то $w_{ij} = 0$.

Значення вершин в зваженому орграфі змінюються з часом відповідно до формули:

$$v_i(t+1) = v_i(t) + \sum_{j=1}^n w_{ji} p_j(t) \quad (1)$$

де n – кількість вершин у графі.

Зміна імпульсу $p_j(t)$ задається різницею

$$p_j(t) = v_i(t) - v_i(t-1) \text{ при } t = 1, 2, \dots$$

Також необхідно задати початкові умови при $t = 0$: $p_j(0)$ та $v_i(0)$.

НЕДОЛКИ ІМПУЛЬСНОГО МЕТОДУ

Основним і найвагомим недоліком імпульсного методу є розбіжність результатів у випадку нестійкого імпульсного орграфу. Якщо всі ненульові власні значення зваженого орграфу, якому відповідає матриця суміжності W , різні і за абсолютною величиною не перевищують одиницю, то орграф – імпульсно-стійкий для всіх простих імпульсних процесів. Інакше, орграф – імпульсно-нестійкий для деякого простого імпульсного процесу [4]. Тобто знайдеться така вершина, якій передається початковий імпульс, що в деякій (можливо, іншій) вершині імпульс стане нескінченно великим. Це означає, що в імпульсному методі значення $v_i(t)$ при $t \rightarrow \infty$ визначити буде неможливо. Тому існують методи стабілізації нестійких орграфів [6].

Також у роботі [7] проведений критичний аналіз основних методів дослідження когнітивних карт і наведено ряд недоліків та протиріч, які виникають при застосуванні імпульсного методу:

- 1) розбіжність $v_i(t)$ при $t \rightarrow \infty$, якщо ряд в (1) розбіжний;
- 2) результат розрахунку – $v_i(t)$ залежить, відповідно до (1), від початкових значень $p_j(0)$;
- 3) початкове значення $v_i(0)$ не впливає на залежність від t (входить у вираз для $v_i(t)$ як доданок);
- 4) порушення масштабної інваріантності при

збільшення елементів матриці W в η разів. В деяких випадках призводить до розбіжності ряду в (1).

МЕТОД НАКОПИЧУВАЛЬНОГО ВПЛИВУ

В даній роботі запропоновано новий метод дослідження когнітивних карт – метод накопичувального впливу. Ідея цього методу полягає у тому, що розглядаються попарно всі вершини зваженого орграфу і визначається значення z_{ij} впливу вершини u_i на u_j (де $i, j = 1, 2, \dots, n$). В результаті, буде отримана матриця впливу – Z , що скрадатиметься з елементів z_{ij} .

Для того, щоб визначити z_{ij} – загальний вплив вершини u_i на u_j , здійснюється наступне:

1) будуються всі можливі прості шляхи з вершини u_i в u_j .

2) на кожному шляху розраховується вплив u_i на u_j , враховуючи знаки вагових значень, що приписані ребрам: вершині u_i надається одиничний імпульс, який розповсюджується по ланцюгу від u_i до u_j відповідно до правил а) – д):

$$\text{а) } u_i \xrightarrow{+} u_k \xrightarrow{-} u_j$$

Якщо вершина u_i впливає позитивно на вершину u_k , а u_k впливає на u_j негативно, то вершина u_i підсилює негативний вплив вершини u_k на u_j . Отже, в результаті, вершина u_i негативно впливає на u_j .

$$\text{б) } u_i \xrightarrow{-} u_k \xrightarrow{-} u_j$$

Якщо вершина u_i послаблює негативний вплив вершини u_k на u_j , то в даному випадку вершина u_i позитивно впливає на u_j .

$$\text{в) } u_i \xrightarrow{+} u_k \xrightarrow{+} u_j$$

Тут вершина u_i позитивно впливає на u_j , підсилюючи позитивний вплив вершини u_k на u_j .

$$\text{г) } u_i \xrightarrow{-} u_k \xrightarrow{+} u_j$$

В даному випадку відбувається послаблення позитивного впливу вершини u_k на u_j за рахунок негативного впливу u_i на u_k . Як наслідок, вершина u_i здійснює негативний вплив на u_j .

Щоб визначити частковий вплив на кінцеву вершину u_j який накопичується від початкової вершини u_i на k -му простому шляху, необхідно розрахувати загальний вплив z_{ij}^k на вершину u_j , накопичений від всіх вершин q_t^k , що входять у k -й шлях (враховуючи правила а)–д); потім – відняти від z_{ij}^k вплив на вершину u_j , який накопичується від всіх вершин q_t^k на цьому ж шляху без урахування початкової вершини $q_0 = u_i$, тобто – \tilde{z}_{ij}^k .

z_{ij}^k та \tilde{z}_{ij}^k розраховуються ітеративно відповідно до формул:

$$z_{ij}^k(t+1) = \left(1 + \text{sign}(z_{ij}^k(t)) * \alpha \left(\frac{z_{ij}^k(t)}{\mu} \right) \right) * w(q_t^k, q_{t+1}^k) \quad (2)$$

$$\tilde{z}_{ij}^k(r+1) = \left(1 + \text{sign}(\tilde{z}_{ij}^k(r)) * \alpha \left(\frac{\tilde{z}_{ij}^k(r)}{\mu} \right) \right) * w(q_r^k, q_{r+1}^k) \quad (3)$$

де q_t^k – послідовність вершин у k -му шляху ($q_0 = u_i, q_{m-1} = u_j$);

$t = 0, 1, \dots, m-2$, а $r = 1, \dots, m-2$ (m – кількість вершин, що входять до k -го шляху).

Необхідні початкові умови: $z_{ij}^k(0) = 0$, $\tilde{z}_{ij}^k(1) = 0$.

$$\mu = \left| \max |w_{ij}| \right|,$$

де $i = 0, 1, \dots, n$, $j = 0, 1, \dots, n$ (n – розмірність когнітивної карти).

Загальний вплив z_{ij} на вершину u_j , який накопичується від вершини u_i , – це сума по всіх простих шляхах (що сполучають u_i та u_j) всіх часткових впливів, розрахованих як різниця між (2) та (3)

$$z_{ij} = \sum_{k=1}^s (z_{ij}^k - \tilde{z}_{ij}^k)$$

де s – кількість простих шляхів із вершини u_i в u_j .

Якщо u_j недосяжна із вершини u_i , то $z_{ij} = 0$.

Запропонований метод має експоненційну обчислювальну складність $O(e^{2n})$, де n – кількість вершин когнітивної карти.

Тож для того, щоб зменшити кількість ітераційних кроків (а отже, і зменшити основне навантаження на обчислювальну складність методу), пропонується зробити нульовий крок, ідея якого:

а) побудувати матрицю досяжності A для початкової матриці W ;

б) кожен елемент w_{ij} початкової матриці W домножити на відповідний елемент a_{ij} матриці A .

ПЕРЕВАГИ ТА ПРИКЛАДИ РОБОТИ МЕТОДУ НАКОПИЧУВАЛЬНОГО ВПЛИВУ

Однією із переваг запропонованого методу, на відміну від імпульсного, є те, що для будь-якої скінченної кількості вершин n та при будь-яких значеннях w_{ij} зваженого орграфу, метод накопичувального впливу має обмежений за числовим значенням результат розрахунку – z_{ij} : $-2 * \mu * s < z_{ij} < 2 * \mu * s$ для $\forall s > 0$ (де s – кількість простих шляхів із вершини u_i в u_j).

Наприклад, у роботі [8] розглядається когнітивна карта взаємовпливу моделі санітарного стану (рис. 1).

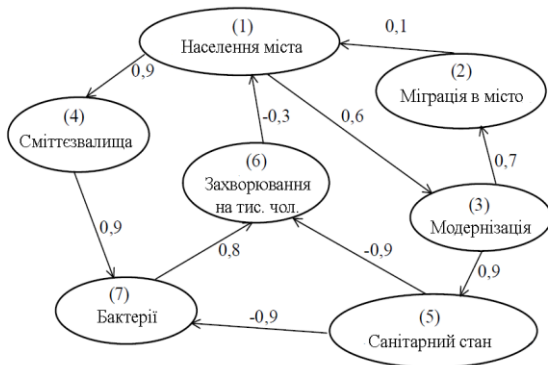


Рис. 1 Когнітивна карта взаємовпливу моделі санітарного

Для знакового ографу, представленого на рис. 1, матриця суміжності має вигляд

$$W = \begin{pmatrix} 0 & 0 & 0,6 & 0,9 & 0 & 0 & 0 \\ 0,1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,7 & 0 & 0 & 0,9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,9 \\ 0 & 0 & 0 & 0 & 0 & -0,9 & -0,9 \\ -0,3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,8 & 0 \end{pmatrix} \quad (4)$$

Ограф, що відповідає (4) – імпульсно-стійкий для всіх простих імпульсних процесів. Його ненульові власні значення за абсолютною величиною рівні: 0,686; 0,686; 0,634; 0,625; 0,625; 9,22e-17.

Сумарний абсолютний вплив кожної із вершин та їх ранжування для матриці (4), відповідно до імпульсного методу та методу накопичувального впливу, представлений у табл. 1.

Таблиця 1

Вершина (№)	Імпульсний метод	Вершина (№)	Метод накопичувального впливу
5	5,44	3	3,816
3	4,06	5	3,599
1	3,38	1	3,371
4	2,21	4	1,683
7	1,79	7	1,65
6	1,27	6	1,598
2	0,4	2	0,493

Ранжування вершин (табл. 1), внаслідок застосування методу накопичувального впливу, близьке за змістом до ранжування, яке дає імпульсний метод. Невідповідність ранжування викликана специфікою нового запропонованого методу.

Внаслідок збільшення кожного елемента матриці суміжності W у η разів, ограф, що відповідає новій W_η стане імпульсно-нестійким для деякого простого імпульсного процесу. Тому застосування імпульсного методу є неможливим. Відповідно до методу накопичувального впливу зберігається не лише масштабна інваріантність, а й пропорційне збільшення загального впливу для кожної із вершин в η разів

(табл. 2). А отже, зберігається і ранжування вершин, в порівнянні з ранжуванням для матриці W .

Таблиця 2

Вершина (№)	Загальний вплив W	Загальний вплив для W_2
5	3,816	7,632
3	3,599	7,198
1	3,371	6,742
4	1,683	3,366
7	1,65	3,299
6	1,598	3,197
2	0,493	0,986

ВИСНОВКИ

Отже, запропонований в цій роботі метод накопичувального впливу дозволяє подолати недоліки та протиріччя імпульсного методу. Наприклад, на відміну від імпульсного методу, для будь-якої скінченної кількості вершин та при будь-яких значеннях зваженого ографу, результат z_{ij} внаслідок застосування методу накопичувального впливу, існує й обмежений за числовим значенням. Початкові значення $z_{ij}^k(0)$ та $\tilde{z}_{ij}^k(0)$ у новому методі впливають на залежність $z_{ij}^k(t)$ та $\tilde{z}_{ij}^k(t)$ від t (тому вважається, що $z_{ij}^k(0) = 0$ та $\tilde{z}_{ij}^k(1) = 0$). $z_{ij}^k(t)$ та $\tilde{z}_{ij}^k(t)$ не залежить, відповідно до (2) та (3), від початкових значень імпульсу. Також, внаслідок збільшення елементів матриці W в η разів, не змінюється ранжування вершин за ступенем їх впливу, а загальна величина впливу кожної із вершин збільшується пропорційно в η разів.

Запропонований метод має і свій недолік, пов'язаний з необхідністю пошуку всіх можливих простих шляхів між кожною парою точок когнітивної карти. Алгоритм пошуку має експоненційну алгоритмічну складність – $O(e^{2n})$. Проте використання розріджених матриць дає змогу застосовувати метод накопичувального впливу для аналізу когнітивних карт великого розміру.

ЛІТЕРАТУРА

1. Axelrod R. The Structure of Decision: Cognitive Maps of Political Elites / Axelrod. – Princeton: Princeton University Press, 1976. – 404 p.
2. Kosko B. Fuzzy Cognitive Maps / Kosko. // International Journal of Man-Machine Studies. – 1986. – №24. – P. 65–75.
3. Carlsson C. Fuzzy reasoning in decision making and optimization / C. Carlsson, R. Fuller. – Physica, 2012.
4. Roberts F S. Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems / Fred Roberts. – New Jersey: Rutgers University, Prentice-Hall Inc., 1976.
5. Згуровский М. З. Принципы и методы управления импульсными процессами в когнитивных картах сложных систем. Часть 1 / М. З. Згуровский, В. Д. Романенко, Ю. Л. Милявский. // Проблемы управления и информатики. – 2016. – №2. – С. 21–29.
6. Zgurovsky M. Z. Advances in Dynamical Systems and Control. Springer International Publishing / M. Z.

Zgurovsky, V. D. Romanenko, Y. L. Milyavsky. // Springer International Publishing. – 2016. – P. 363–374.

7. Снарский А. А. Критический анализ основных методов анализа когнитивных карт / А. А. Снарский, Д. В. Ландэ. // Реєстрація, зберігання і обробка даних: зб. наукових праць за матеріалами щорічної підсумкової наукової кон-ференції 16-17 травня 2016

року Інституту проблем реєстрації інформації НАН України. – 2016. – С. 79–80.

8. Таран Т. А. Искусственный интеллект. Теория и приложения / Т. А. Таран, Д. А. Зубов. – Луганск: Восточноукраинский национальный ун-т им. Владимира Даля, 2006. – 242 с.

**РОЗПОДІЛЕНІ ТА
ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ**

**DISTRIBUTED AND PARALLEL
COMPUTING**

Ускорение обмена данными в мультипроцессорных системах

Валерий Жабин
КПИ им. Игоря Сикорского
Киев, Украина
vizhabin@comsys.kpi.ua

Валентина Жабина
КПИ им. Игоря Сикорского
Киев, Украина
val.zhabina2@gmail.com

Аннотация. Исследуются методы межпроцессорного обмена данными в системах с общей шиной и различными способами доступа к памяти. Рассматривается возможность синхронизации процедур обмена по шине данных и шине управления. Предлагается метод ускорения обмена данными в централизованных системах, основанный на непосредственном доступе каждого процессора к части адресного пространства локальной памяти других процессоров.

Ключевые слова: мультипроцессорные системы, организация памяти, обмен данными.

ВВЕДЕНИЕ

Непрерывное расширение области применения систем управления в реальном времени приводит к необходимости решения все более сложных задач. В ряде случаев это требует использования параллельных вычислительных систем [1, 2].

Для одновременной обработки параллельных ветвей алгоритмов должна быть обеспечена возможность одновременного выполнения в разных процессорах различных команд над различными данными. С точки зрения функциональной классификации Флинна [3] системы, которые позволяют выполнять одновременно разные команды в разных процессорах, относятся к классу MIMD (Multiple Instruction, Multiple Data – множественный поток команд, множественный поток данных). В системах автоматического управления преимущественно применяются мультипроцессорные вычислительные системы (МВС) с общей шиной. Преимуществом общей шины является низкая стоимость, более простая адаптация программного обеспечения, разработанного для однопроцессорных систем, простота резервирования процессоров [2, 4].

Применение технологии RSoC (Reconfigurable System on Chip – реконфигурируемая система на кристалле) позволяет с использованием ПЛИС создавать компактные МВС, которые легко адаптируются к реальным условиям эксплуатации [5, 6].

Выбор архитектуры МВС должен осуществляться с учетом не только минимизации времени параллельной обработки данных в процессорах, но и минимизации затрат на обмен данными между процессорами.

Следовательно, важной задачей ускорения параллельной обработки информации является создание методов и средств ускорения обмена данными между процессорами.

ОРГАНИЗАЦИЯ ОБМЕНА ДАННЫМИ

Наиболее простыми с точки зрения технической реализации являются симметричные мультипроцессоры (symmetric multiprocessor) или SMP-системы, содержащие одинаковые процессоры (процессорные ядра) [1, 2, 7]. Все процессоры имеют одно общее адресное пространство, через которое осуществляется доступ к общему системному ресурсу (к памяти и внешним устройствам).

Общая память используется не только для хранения системной информации и обмена данными между процессорами, но и для хранения программ процессоров. При выполнении своих программ процессоры обращаются к памяти за командами, что приводит к большим затратам времени на ожидание доступа и разрешение конфликтов. Наличие в каждом процессорном модуле кэш-памяти уменьшает число конфликтов при обращении к общей памяти. Однако в этом случае возникают трудности с обеспечением когерентности данных, что усложняет взаимодействие процессоров.

Указанные недостатки приводят к тому, что в таких системах используется небольшое число процессоров (обычно, не более 32), что ограничивает их производительность.

С целью ускорения обмена данными между компонентами системы применяют более сложную организацию коммуникационной среды системы, причем, с различными процедурами доступа к различным частям памяти.

Уменьшение числа конфликтных ситуаций может быть достигнуто за счет использования в каждом процессорном модуле собственной локальной памяти, в которой хранится программа данного процессора [2]. Процессор взаимодействует со своей памятью через локальную шину, а с общей памятью – через общую (системную) шину.

Такой подход не требует обращения процессоров к общей шине за командами, но на время обмена данными между процессорами это практически не сказывается, так как обмен (как и в системах с однородным доступом к памяти) осуществляется через общую память (минимум два обращения к общей шине).

Для ускорения обмена данными между процессорами в системах может быть предусмотрен непосредственный доступ каждого процессора к определенной области адресного пространства

локальной памяти других процессоров, называемой коммуникационной памятью [2].

Интенсивность обращений процессоров к системной магистрали при обмене данными можно оценить с помощью коэффициента эффективности использования системной магистрали для обмена данными [2], который определяется по формуле

$$K_{CM} = \frac{N_S + N_D}{M}, \quad (1)$$

где N_S – число обращений к системной магистрали для инициализации и синхронизации процедур обмена (непроизводительные затраты времени); N_D – число обращений к системной магистрали для передачи непосредственно данных; M – число передаваемых слов.

В централизованных системах функции управления процессом обработки данных возлагаются на управляющий (master) процессор, который организует работу подчиненных (slave) процессоров.

В работе [2] исследованы процедуры обмена данными между процессорами для случая синхронизации обмена как по шине данных, так и по шине управления. В первом случае синхронизация обмена осуществляется с помощью примитивов низкого уровня (флажки, семафоры), а во втором – через сигналы прерываний. Согласно формуле (1) получены следующие зависимости для коэффициента эффективности соответственно при синхронизации обмена по шине данных и по шине управления:

$$K_{CM} = 2 + 4/M, \quad (2)$$

$$K_{CM} = 2 + 2/M. \quad (3)$$

При синхронизации по шине управления подчиненные процессоры могут при готовности к обмену сообщать об этом управляющему процессору сигналом прерывания, что сокращает число обращений к магистрали.

Как следует из выражений (2) и (3), при увеличении размерности массивов данных среднее число обращений к системной магистрали стремится к двум.

МЕТОД УСКОРЕНИЯ ОБМЕНА ДАННЫМИ

В централизованных системах вычислительный процесс может быть организован таким образом, чтобы управляющий процессор взаимодействовал с подчиненными путем передачи им команд по шине данных. Команды могут передаваться в специальный регистр команд, который включен в адресное пространство, доступное для управляющего процессора. Работа подчиненных процессоров сводится к выполнению команд, которые дает им управляющий процессор.

Уменьшения числа обращений к системной магистрали в процессе пересылки данных между процессорами в централизованных системах можно достигнуть за счет передачи управления непосредственно пересылкой данных процессору-приемнику информации в соответствии с командой управляющего процессора.

Возможный вариант организации архитектуры

системы показан на рис. 1. Система представлена в виде MSBC-модели на графическом языке [8].

Система рассматривается на функциональном уровне, на котором в качестве объектов видны устройства (процессоры, память, устройства ввода-вывода, системы коммутации и т.д.). Объекты взаимодействуют на уровне протоколов обмена информацией.

Модель представлена двумя компонентами:

- графическим описанием с использованием унифицированных функциональных обозначений;
- картой распределения адресного пространства.

Графическое описание архитектуры является графом, который отображает основные компоненты систем и потоки управляющей информации, которые определяют способ их взаимодействия. Дуги графа ориентированы в направлении от активного элемента, который является инициатором действия, к элементу, с которым это действие должно осуществляться. При этом направление передачи данных может не совпадать с ориентацией дуг.

Управляющий процессор P_1 пересылает команды, соответствующие определенным заданиям, в регистры IR_i ($i = 2, n$) подчиненных процессоров P_i . Запись слова в регистр команд автоматически вызывает прерывание, по которому процессор P_i переходит на подпрограмму выполнения задания (обслуживания прерывания). После выполнения задания подчиненный процессор P_i сообщает об этом управляющему процессору P_1 пересылкой в IR_1 определенного слова, что вызывает соответствующее прерывание. При одновременном обращении процессоров к магистралям арбитраж в соответствии с приоритетами выполняется аппаратными средствами.

Пересылка массива из процессорного модуля P_i в модуль P_j требует четыре обращения к системной магистрали (к регистрам IR_i , IR_j и два раза к IR_1) для синхронизации процесса и M обращений процессора P_j для пересылки массива. Процессор P_i формирует массив в своей коммуникационной памяти, которая доступна P_j . Процессор P_j пересылает массив в свою локальную память. Следовательно, коэффициент эффективности использования магистрали имеет вид

$$K_{CM} = 1 + 4/M. \quad (4)$$

На рис.1а показаны средства прерывания управляющего процессора подчиненными процессорами со своих локальных шин, что не требует двух обращений к системной магистрали для подтверждения выполнения заданий. При таком подходе к организации обмена получим

$$K_{CM} = 1 + 2/M. \quad (5)$$

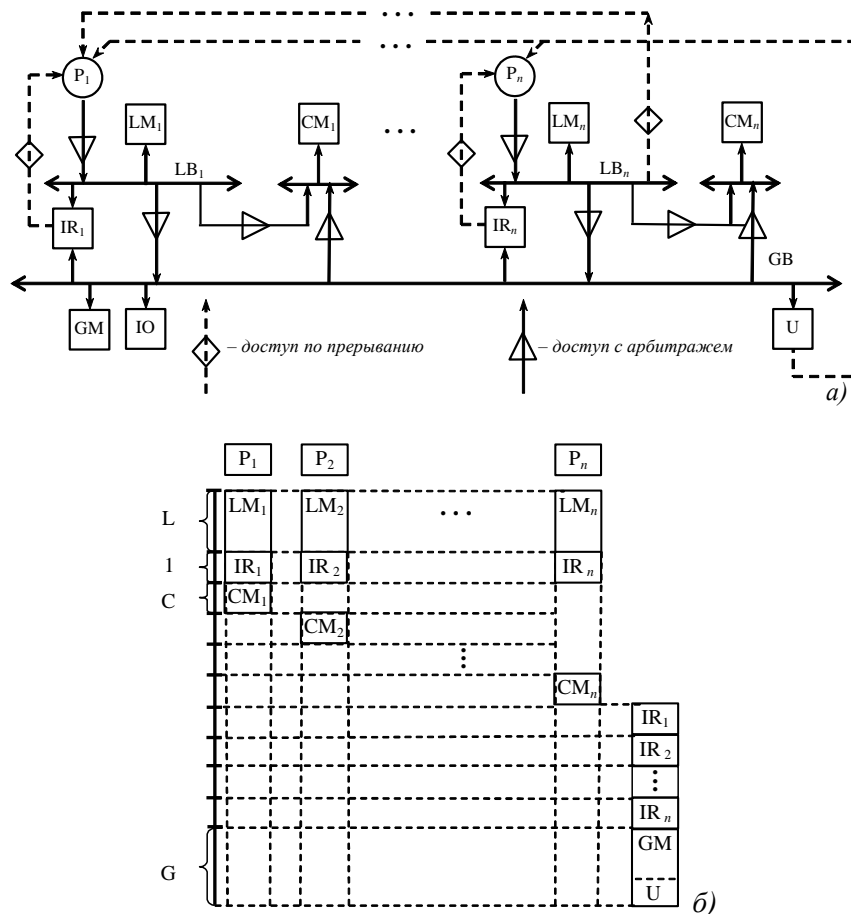


Рис. 1. Модель системи: а – граф управляючих потоків; б – карта розподілення адресного простору; P_i – процесор; LM_i – локальна пам'ять; CM_i – комунікаційна пам'ять; GM_i – загальна пам'ять; LB_i – локальна шина; GB – загальна системна шина; IR_i – реєстр команд; IO – зовнішні пристрої; U – спеціальний процесор; C, I, L, G – позначення об'ємів пам'яті

Сравнивая формулы (2) и (3) с формулами (4) и (5) становится очевидным, что предложенный метод обмена данными уменьшает число обращений к системной магистрали при пересылке массивов. С увеличением длины массива число обращений к магистрали для пересылки одного слова стремится к минимально возможному значению – одному обращению.

Для ускорения обработки данных в состав системы может быть введен специализированный процессор U , реализующий вычисления под управлением потока данных. Наиболее эффективно использование таких устройств при реализации мелкозернистых алгоритмов [9]. Например, в системах автоматического управления часто необходимо выполнять мелкозернистые алгоритмы интерполяции функций, преобразования систем координат в многомерном пространстве. Разработаны методы реализации алгоритмов с формированием внутренних параллельных потоков команд, что ускоряет вычисления [10].

В адресном пространстве общей памяти GM выделяется область адресов внутренней памяти специализированного процессора U для каждого процессора. Процессоры передают в специализированный процессор данные для вычислений по выделенным для них адресам. После выполнения алгоритма специализированный процессор сообщает об этом соответствующему процессору сигналом

прерывания. Процессор считывает результаты из общей памяти.

ЗАКЛЮЧЕНИЕ

В работе проведен анализ архитектур мультипроцессорных систем с однородным и неоднородным доступом к памяти с точки зрения эффективности использования коммуникационной среды систем. Показана необходимость уменьшения загрузки коммуникационной среды для ускорения обработки данных.

Предложенный метод пересылки массивов данных обеспечивает уменьшение числа обращений к системной магистрали при обмене данными между процессорами. Уменьшение загруженности коммуникационной среды при пересылке данных позволяет увеличить число процессорных модулей и, следовательно, повысить производительность систем.

Показана возможность ускорения решения задач за счет параллельной работы процессоров и специализированного процессора, управляемого потоком данных.

Все это создает предпосылки для уменьшения времени обработки информации и, как следствие, расширения области применения систем реального времени.

ЛИТЕРАТУРА

1. Воеводин В.В. Параллельные вычисления / В.В Воеводин, Вл.В.Воеводин. – СПб.: БХВ, 2002. – 608 с.
2. Жабин В.И. Архитектура вычислительных систем реального времени / В.И.Жабин. – К.: ВЕК+, 2003. – 176 с.
3. Flynn M. Some Computer Organisations and Their Effectiveness / M.Flynn // IEEE Trans. Computers. – 1972. – Vol. 21, N 9. – P. 948-960.
4. Жабин В.И. Повышение эффективности параллельной обработки данных и обеспечение отказоустойчивости мультипроцессорных вычислительных систем реального времени / В.И.Жабин // Искусственный интеллект. – 2015. – №1-2. – С. 87-97.
5. De Hon A. Reconfigurable Computing Architectures / A.DeHon, R.Tessier, K.Poczek // Proceedings of the IEEE. – 2015. – Vol. 103, N 3. – P. 332-354.
6. Koch D. Partial Reconfiguration on FPGAs. Architectures, Tools and Applications / D.Koch // – New York: Springer. – 2013. – 295 p.
7. Wilson A.B. More Power to You: Symmetrical Multiprocessing Gives Large-Scale Computer Power at a Lower Cost with Higher Availability / A.B.Wilson // Datamation. –1980. – N 6. – P. 216-223.
8. Жабин В.И. Графическое описание архитектуры вычислительных систем / В.И.Жабин // Вісник Національного технічного університету України «КПІ». Інформатика, управління та обчислювальна техніка. – К: “ВЕК+”, 2001. – № 36. – С. 80–88.
9. Silva J.G.D. Design of processing subsystems for Manchester data flow computer / J.G.D.Silva, J.V.Wood // IEEE Proc. N.Y. – 1981. – Vol. 128, N 5. – P. 218 – 224.
10. Жабина В.В. Параллельное формирование команд в потоковых системах / В.В.Жабина // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. праць. – К.: Век+. – 2009. – № 50. – С. 113-117

Розробка паралельного алгоритму кластеризації даних засобами автоматизації проектування

Прихидько В.С.
КПІ ім. Ігоря Сікорського
Київ, Україна

Дорошенко А.Ю.
КПІ ім. Ігоря Сікорського
Київ, Україна

Яценко О.А.
Інститут програмних систем
НАНУ
Київ, Україна

Анотація. Розроблено паралельні алгоритми для вирішення задачі кластерного аналізу методом К-середніх з допомогою програмного інструментарію проектування й синтезу програм, що використовує алгебро-алгоритмічний підхід для перетворення формальних специфікацій алгоритмів у програмний код. Досліджено та порівняно швидкодію послідовного та паралельних варіацій алгоритму.

Ключові слова: кластерний аналіз кластер, великі обсяги даних, К-середніх, паралельний алгоритм.

ВСТУП

Зростання темпів накопичення корисної інформації в останні роки породжує необхідність вдосконалення методів опрацювання великих обсягів даних. В цьому напрямку своє місце відведено інтелектуальному аналізу даних, що включає, зокрема, методи, основані на моделюванні, ймовірнісних узагальненнях, асоціюванні та пошуках закономірностей. Кластерний аналіз або автоматичне групування об'єктів є частковим випадком такого аналізу. Великою мірою розвитку цієї дисципліни сприяло проникнення в сферу аналізу даних ідей, що виникли в теорії штучного інтелекту.

На сучасному етапі розвитку науки та технологій дослідження і розробка методів та алгоритмів для реалізації інтелектуального аналізу даних є пріоритетними напрямками розвитку та вдосконалення галузі аналізу даних. Зростання обсягу інформаційних потоків зумовлює актуальність даної тематики, оскільки виникають нові задачі та проблеми, пов'язані з обробкою великих обсягів даних різної організаційної структури. Різноманітність даних, що потребують обробки, обмежує застосовуваність традиційних методів аналізу даних, що використовують статистику, інформаційний пошук чи знання експертів. З цього і випливає необхідність переходу на новий рівень в аналізі даних, в основі якого лежить інтелектуальний аналіз.

КЛАСТЕРНИЙ АНАЛІЗ

Кластерний аналіз чи кластеризація — завдання групування набору об'єктів таким чином, щоб об'єкти одної групи (називаються кластером) були більш схожими (в тому чи іншому сенсі) один до одного, ніж до об'єктів в іншій групі. Це головне завдання пошукового аналізу даних, і загальний метод аналізу статистичних даних, що використовується в багатьох

областях, включаючи машинне навчання, розпізнавання образів, аналіз зображень, пошук інформації, стискання даних, комп'ютерну графіку. Кластерний аналіз сам по собі це не один специфічний алгоритм, але загальна задача для вирішення. Вона може бути вирішена різними алгоритмами, які значно відрізняються по своєму уявленню про те, що собою являє кластер і як його ефективно знайти.

Популярним визначенням кластера є група з невеликими відстанями між членами кластера, густі області простору даних, інтервали чи конкретні статистичні розподіли. Тому кластеризація може бути сформульована як задача багатоцільової оптимізації. Відповідний алгоритм кластеризації і параметри (функція схожості (відстані), поріг густоти, кількість очікуваних кластерів) залежать від індивідуального набору даних і способу використання результатів. Кластерний аналіз сам по собі не є автоматичною задачею, а є ітеративним процесом виявлення знань чи задачею інтерактивної багатоцільової оптимізації, яка працює способом спроб і помилок[1,2].

ПОСТАНОВКА ЗАДАЧІ

Більшість завдань кластеризації виконують ітеративні процедури для пошуку локальних чи глобальних оптимальних рішень на великогабаритних наборах даних. Реальні дані дуже важко інтерпретувати для кластеризації. Тому кластеризація вимагає багато експериментів з різними алгоритмами або з різними функціями для одного й того ж набору даних, що вимагає багато обчислювального часу. Тому паралелізація алгоритмів кластеризації є дуже практичним підходом для прискорення обчислень. Більшість паралельних алгоритмів кластеризації використовують протокол передачі повідомлень в архітектурі master-slave. В ієрархічних алгоритмах кластеризації, що використовують певну структуру даних, для збереження часу оновлення використовуються також гіперкуб або мережева база даних. В даному дослідженні за основу було взяти послідовний алгоритм кластеризації К-середніх та з допомогою засобів автоматизованого проектування алгоритмів побудовано дві паралельні версії.

СИСТЕМА ПС

В даній роботі для автоматизації побудови паралельних алгоритмів кластеризації пропонується програмний інструментарій проектування й синтезу

програм (система ППС)[3], що використовує алгебро-алгоритмічний підхід для перетворення формальних специфікацій алгоритмів у програмний код. Інструментарій призначений для конструювання схем алгоритмів, поданих в модифікованих системах алгоритмічних алгебр В.М. Глушкова (САА-М), а також генерації відповідних програм мовами програмування C++, C#, Java та ін. Система ґрунтується на методі діалогового конструювання синтаксично правильних програм (ДСП-методі), основна ідея якого полягає в порівневому проектуванні алгоритмів зверху вниз шляхом суперпозиції мовних конструкцій САА-М, які користувач обирає зі списку та додає в дерево конструювання алгоритму. На основі побудованого дерева та шаблонів програмних реалізацій конструкцій САА-М, що зберігаються у базі даних системи ППС, виконується автоматична генерація тексту програми цільовою мовою програмування. Особливістю інструментарію є інтеграція трьох форм подання алгоритмів у процесі їх проектування: аналітичної (регулярні схеми), природно-лінгвістичної (САА-схеми) та графової (граф-схеми). Для реалізації паралельних алгоритмів кластеризації в якості цільової мови програмування обрано C#.

РЕЗУЛЬТАТИ РОБОТИ АЛГОРИТМІВ

За основу для даної роботи було взято один з найпопулярніших алгоритмів кластерного аналізу К-середніх. Послідовна версія SA (Sequential Algorithm) алгоритму поширена у вільному доступі. З допомогою засобу автоматизації проектування було побудовано дві паралельні версії алгоритму PAM (Parallel Algorithm Manual) - з ручним розкладанням задач по потокам та PAB (Parallel Algorithm Built-in) - з розкладанням на потоки, що пропонує середовище CLR (Common Language Runtime Microsoft .NET Framework). Далі всі алгоритми було застосовано на невеликому наборі даних і отримано наступні результати часу виконання кластеризації в мс.

Таблиця 1. Результати швидкодії алгоритмів

Тип\N	1	2	3	4	5	6	7	8	9	10
SA	359.2	362.2	359.1	411.2	364.2	357.2	378.2	364.2	358.2	374.2
PAM	20.1	44.5	41.8	40.6	23.6	19.9	29.2	20.5	39	40.4
PAB	1	1	1	2	1	1	1	1	0.9	1

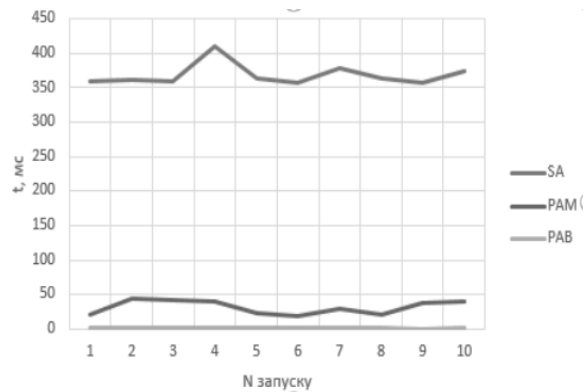


Рис. 1. Порівняння швидкодії алгоритмів

$$tSA = \sum_i tSA_i \quad (1)$$

$$tPAM = \sum_i tPAM_i \quad (2)$$

$$tPAB = \sum_i tPAB_i \quad (3)$$

Розрахуємо середні показники.

$$tSA = 368.8\text{мс}, tPAM = 31.9\text{мс}, tPAB = 1.09\text{мс}.$$

ВИСНОВКИ

За результатами дослідження (середні показники), поданими у табл.1 та на рис.1 видно, що паралельний підхід зменшує кількість часу, необхідного для опрацювання даних та виконання кластерного аналізу, але також зрозуміло, що результат залежить від якості розпаралелювання, оскільки різні версії паралельного алгоритму дали різні результати.

ЛІТЕРАТУРА

1. Cluster analysis [Електронний ресурс] –Режим доступу до ресурсу:https://en.wikipedia.org/wiki/Cluster_analysis.
2. Cluster Analysis: Basic Concepts and Algorithms [Електронний ресурс] – Режим доступу до ресурсу:<https://www.users.cs.umn.edu/~kumar/dmbook/ch8.pdf>.
3. Методы алгебраического программирования: формальные методы разработки параллельных программ/АндонФ.И., Дорошенко А.Е.,ЖеребК.А., Шевченко Р.С., Яценко Е.А.—К.: Наукова думка, 2017.—440 с.

Isoefficiency control tool for distributed computing

Heorhii Loutskii, Artem Volokyta, Pavlo Rehida
 Igor Sikorsky Kyiv Polytechnic Institute
 Kyiv, Ukraine
artem.volokita@kpi.ua, regidapavel@gmail.com

Abstract. This paper describes the implementation of the proposed model for evaluating algorithms for controlling the load isoefficiency of a computer system using the example of two pipelining algorithms. The experimental researches of the model shows the possibility of using this approach for this task. In the paper is proposed the further modifications of the developed model, namely the synthesis of effective topologies and algorithms for a problem of a given dimension.

Keywords: distributed computing, scheduling, isoefficiency.

INTRODUCTION

Efficiency of parallel processing. The efficiency of parallel processing is closely related to degree of parallelism. Existing formulas do not always accurately determine the real performance of the parallel processing acceleration. Formulas do not take into account the communication issues between processors. For fixed-size tasks, the efficiency of parallel computing with increasing number of processors N usually decreases. In addition, it is necessary to evaluate the reasons for the decrease of efficiency due to the restrictions of parallelism and communication costs.

Isoeffective systems. Systems that allows to select the needed efficiency level in advance to solve the problem and to maintain current level of efficiency, are usually called isoeffective systems [1]. The main idea of this work is research the isoefficiency of distributed systems, based on the model of calculations with service agents. This model has ability to control of the size of the input task. This model uses conveyor addition algorithms. Scheduling computing is a critical aspect of increasing the efficiency of distributed computing systems. Accordingly, the possibility of modeling different approaches and algorithms for the effective use of available resources in the decomposition of computations is an actual task for systems, whose computational capabilities increases over real time [2]. An example of an isoeffective system is architecture of a distributed supercomputer Blockchain. Blockchain is a global distributed supercomputer, that uses a distributed database of transactions, which allows to perform calculations with a change of task complexity. Task complexity depends on the total performance of the entire network blockchain.

As a result, the proposed system for modeling will allow us to research the isoefficiency of different algorithms in the given topologies based on time parameters for use in distributed scheduling.

MODEL OF ISOEFFECTIVE COMPUTING

Our goal is to achieve a constant level of efficiency. The parallel computing research system consists of a central node on a separate server and agents. Agents are organized

into a hierarchy and are designed to manage the isoefficiency E_{const} . Method to control of efficiency is proposed in article [3]. The states of the computing nodes are marked with the appropriate color for the active and inactive nodes. The Coordinator performs an efficiency audit and compares the current performance level in distributed systems (Fig. 1).

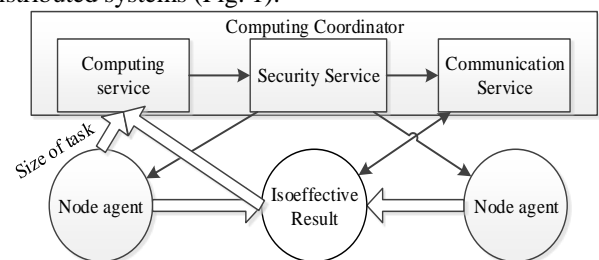


Fig. 1 Model of isoeffective computing

If the efficiency lower than the E_{const} , then the system reduces the size of the task to achieve the appropriate level of efficiency. If the level of efficiency is greater than necessary, the coordinator changes the current number of nodes or increases the size of the task. The proposed system will maintain a needed level of efficiency, which allows to change the size of the task with considering the performance of the nodes and network load.

Description of the system for control the isoefficiency of distributed computing. Let's consider the simplest but rather typical example of conveyor addition. Consider two algorithms for performing this procedure. As a topological organization of the system, we select the hypercube of the second order. In algorithms represent the sequence of steps, where determine the transfer of data from one processor to another, and of the sum is the addition of the value in the receiving processor with the value in the transferring processor.

In algorithm 1, data transfer performed firstly, and then the addition performed. The algorithm 1 is obviously ineffective, as when using it, the acceleration of parallel processing turns out to be less than one. The isoefficiency

formula for algorithm 1: $E = \frac{S}{N} = \frac{n}{2n \log_2 N + n} = 0,2$,

where S – acceleration of algorithm.

Formulas for efficiency E determine the ability to achieve the required efficiency of parallel computing systems. Such an opportunity is provided by varying the parameters n (size of task) and N (number of nodes). Based on this variation, it is possible to achieve linear increase of productivity with increasing number of processors. This means that when performing computational processes, it is possible to determine in advance the necessary efficiency of their implementation.

The algorithm 2, in terms of acceleration and

efficiency, gives quite good results. The second algorithm is effective, as when using it, addition performed firstly, and then the data transfer is performed. The isoefficiency

formula for algorithm 2: $E = \frac{S}{N} = \frac{1}{1 + 2 \frac{N}{n} \log_2 N} = 0,5$. At

the same time, in the existing calculations there is an assumption that the time of the transferring operation and the time of the operation of the addition are the same. In our system, simulation is performed.

EXPERIMENTAL RESEARCH

In order to solve the problem of inefficiency research, the WCF (Windows Communication Foundation) technology was chosen, which provides secure and reliable transactions. Systems that uses WCF, are usually called service-oriented systems. Such systems use services that provide data transportation and execution of tasks in accordance with a given algorithm: the launch of the program on the nodes that will provide services; launch of the main program on a separate server; gradual execution of computational instructions and transactions according to the algorithm. In order to meet the needs of our task, the following system developed.

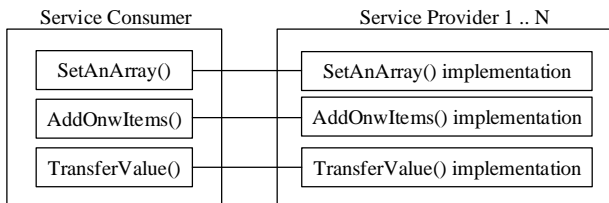


Fig. 2. Client software architecture

SetAnArray (int [] a) sends random values that located in created arrays to each node. AddOnwItems (int index1, int index2) – provides the operation of the addition, in this case, arguments are array indexes. It is important to note that the amount received is stored in the current array by the index1 index. TransferValue (string address, int index) – is used in cases where for some SP it is necessary to get the value of the array of the index [index] from another SP1.

Let's consider the results of experiments for $N = 4$.

TABLE. TIME OF ALGORITHMS 1, 2 EXECUTION, AND VALUES OF ISOEFFICIENCY.

N	Alg1	Alg2	Iso1	Iso2
32	8,41	5,18	6,4	16
64	13,82	6,67	12,8	32
128	22,96	9,64	25,6	64
256	42,04	15,53	51,2	128
512	81,71	29,81	102,4	256

S_1, S_2 – acceleration of algorithms 1,2, T_{S1}, T_{S2} – time of sequential computing, T_{P1}, T_{P2} - time of parallel computing, E_1, E_2 – coefficients of isoefficiency. Thereby:

$$S_1 = \frac{T_{S1}}{T_{P1}}, S_2 = \frac{T_{S2}}{T_{P2}} \text{ since } T_{S1} = T_{S2}, \text{ then } \frac{S_1}{S_2} = \frac{T_{P1}}{T_{P2}}$$

$$E_1 = \frac{S_1}{N}, E_2 = \frac{S_2}{N}, \frac{E_1}{E_2} = \frac{T_{P2}}{T_{P1}}$$

Fig. 3 shows the results of calculations in the form of ratio of the execution time of algorithms 1 and 2, and ratio of the isoefficiency coefficients.

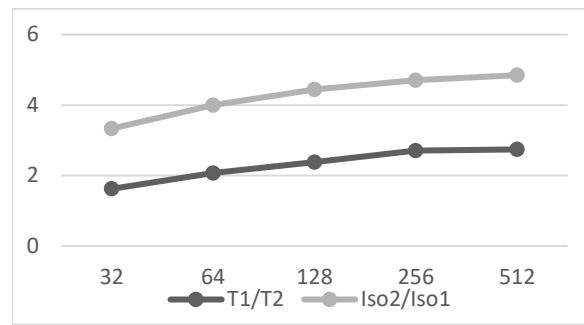


Fig. 3. Computing time for algorithms 1,2

Also Fig. 3 shows that algorithm 2 is faster. The proposed system allows to perform distributed computing modeling with time control of computations.

Fig. 4 shows the theoretical values (upper curves) are

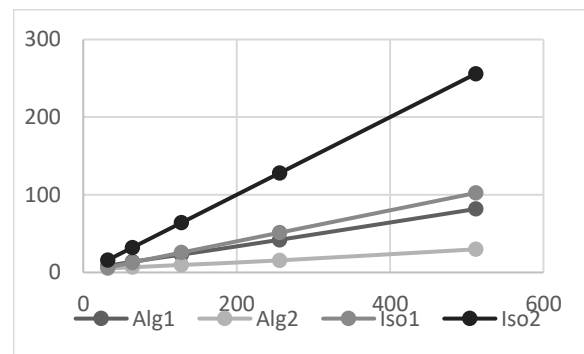


Fig. 4. Acceleration coefficients comparison

higher than the results of calculations, which indicates the need for additional control of the efficiency during calculations for the possibility of adjusting the size of the calculations.

CONCLUSION

This paper proposes a distributed computing model for comparing the conveyor addition algorithms with the use of services. The software simulates the change in the number of nodes for different task sizes. Using the proposed model (the rules for selecting the size of the task) for modeling the load on the distributed system allows you to analyze the expediency of increasing/decreasing the dimension of input data for calculations. For example, in terms of changing the number of resources. The implemented model is promising for further modifications. One of the possible improvements is the synthesis of new types of topologies and the choice of granularity of parallelism.

REFERENCES

[1] Grama A. Isoefficiency function: A scalability metric for parallel algorithms and architectures / Grama A., Gupta A., Kumar V.//IEEE Parallel and Distributed Technology, Special Issue on Parallel and Distributed Systems: From Theory to Practice. – 1993. – V. 1. – №. 3. – P. 12-21.

[2] Loutskii H. Development of real time method of detecting attacks based on artificial intelligence / Loutskii H., Volokyta. A., Yakushev O., Rehida P., Vu Duc Thinh // Technology audit and production reserves - № 3/1(29) – 2016 – P. 40 – 46.

[3] Loutskii H. The system for modeling distributed computing to control the isoeffectiveness / Loutskii H.M., Volokyta A.M., Rehida P.G. // Intellectual Systems for Decision Making and Problems of Computational Intelligence ISDMCI'2017: Conference Proceedings.– Kherson: PP Vyshemirsky V. S., 2017. – P. 235-236.

**ПІДГОТОВКА КАДРІВ У
ГАЛУЗІ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

**TRAINING IN INFORMATION
TECHNOLOGIES FIELD**

Оптимизация учебных планов IT-специальностей

Стенин А.А.
КПИ им. Игоря
Сикорского
Киев, Украина
alexander.stenin@yandex.ua

Пасько В.П.
КПИ им. Игоря
Сикорского
Киев, Украина

Лемешко В.А.
КПИ им. Игоря
Сикорского
Киев, Украина

Шитикова И.Г.
Институт
телекоммуникаций и
глобального
информационного
пространства НАН
Украины
Киев, Украина
irinashitikova54@gmail.com

Аннотация. Предлагается подход к формированию и оптимизации перечня дисциплин вариативной части учебных планов IT-специальностей, основанный на методах экспертных оценок. При этом нет ограничений на количество экспертов, показателей качества и исследуемых дисциплин. Использование системы Интернет-опроса позволяет подключить широкий круг специалистов-экспертов для повышения достоверности вероятности оптимальности полученных результатов.

Ключевые слова: оптимизация, учебный план, Интернет-опрос, множество Парето, экспертные оценки.

ВВЕДЕНИЕ

Одна из проблем подготовки студентов IT-специальностей – недостаток практических навыков работы над проектами и знаний современных инструментов и методов разработки или тестирования программного обеспечения. Именно профессиональная деятельность IT-специалистов задает и определяет цели изучения учебных дисциплин, а значит и содержание, и структуру, и формы соответствующей учебной деятельности студентов, готовящихся к будущей профессиональной работе.

В этом контексте большое значение приобретает формирование вариативной части учебных планов IT-специальностей, которая дает возможность расширения и (или) углубления знаний, умений и навыков, определяемых содержанием базовых дисциплин.

Наиболее предпочтительным для формирования и оптимизации такого плана является метод экспертных оценок [1]. В этом случае, оптимальный по качественному составу дисциплин учебный план будет результатом коллективного труда экспертов как в области будущей сферы деятельности студентов IT-специальностей, так и специалистов высшей школы, обладающих высокой компетентностью в отношении выбранных показателей качества учебных дисциплин.

Поскольку ВУЗы, где есть IT-специальности, и специалисты-эксперты данного профиля профессиональной деятельности разбросаны по всей Украине и не всегда есть возможность собрать их в одном месте, предлагается использовать систему Интернет-опроса для сбора необходимых экспертных оценок с последующей их обработкой предлагаемым ниже методом.

РЕШЕНИЕ ЗАДАЧИ

На базе системы Интернет-опроса занесем оценки экспертов в таблицу бальных рангов учебных дисциплин (табл. 1). Здесь r_{ij}^k – ранг (оценка) i -ой дисциплины ($i = 1, N$) j -м экспертом ($j = 1, M$) по k -му показателю качества ($k = 1, Q$). Ранг представляет собой натуральное число в принятой для оценки шкале бальности.

Таблица 1

Оценки экспертов

Показатель качества	J_1		...		J_Q		
	k_1	...	k_N	...	k_1	...	k_N
Дисциплина	k_1	...	k_N	...	k_1	...	k_N
Эксперт	r_{11}^1	...	r_{1N}^1	...	r_{11}^Q	...	r_{1N}^Q
\mathcal{E}_1	r_{11}^1	...	r_{1N}^1	...	r_{11}^Q	...	r_{1N}^Q
...
\mathcal{E}_M	r_{M1}^1	...	r_{MN}^1	...	r_{M1}^Q	...	r_{MN}^Q
Сумма рангов	\sum_1^1	...	\sum_N^1	...	\sum_1^Q	...	\sum_N^Q
Средний ранг	\bar{r}_1^1	...	\bar{r}_N^1	...	\bar{r}_1^Q	...	\bar{r}_N^Q

Сумма рангов и средний ранг определяются соответственно по формулам:

$$\sum_i^k = \sum_{j=1}^M r_{ij}^k, \quad (1)$$

$$\bar{r}_i^k = \frac{1}{M} \sum_i^k. \quad (2)$$

Не теряя общности результата, предположим, что есть 7 учебных дисциплин, которые в результате ранжирования оценок экспертов по показателям содержательности материала (J_1) и методического уровня изложения (J_2) сформировали следующую систему неравенств относительно среднего ранга дисциплины для:

$$J_1 : k_6 > k_1 > k_2 > k_3 > k_7 > k_5 > k_4, \quad (3)$$

$$J_2 : k_6 > k_2 > k_1 > k_7 > k_3 > k_5 > k_4. \quad (4)$$

Смысл неравенств (3) и (4) заключается в парных предпочтениях одного курса по отношению к другому в рамках выбранного показателя качества.

Известно, корректность полученных при обработке экспертных оценок зависит от согласованности группы экспертов. Для оценки согласованности экспертов вычисляем дисперсионный коэффициент конкордации по выбранным показателям качества [2]:

$$W^k = \frac{12S^k}{M^2(N^2 - N) - M \sum_{j=1}^M T_j^k}, \quad (5)$$

где

$$S^k = \sum_{i=1}^N \left(\sum_{j=1}^M r_{ij}^k - r_0^k \right), \quad (6)$$

где r_0^k – усредненная ранговая оценка по k -му показателю, т.е.

$$r_0^k = \frac{1}{N} \sum_{i=1}^N \bar{r}_i^k, \quad (7)$$

T_j^k – показатель связанных рангов в ранжировке j -м экспертом, который определяется как:

$$T_j^k = \sum_{p=1}^{H_j} (h_p^3 - h_p), \quad (8)$$

где H_j – число групп равных рангов j -го эксперта, h_p – число равных рангов в p -й группе связанных рангов при ранжировании j -м экспертом.

При $\bar{W}^k \geq 0,7$ считается, что группа экспертов работает согласовано. В противном случае необходимо переформировать группу экспертов.

Кроме того, целесообразно различать экспертов по степени компетентности и значимости их мнений, т.е. необходимо провести ранжирование самих экспертов. При ранжировании каждому из них присваивается ранг значимости от 1 (наиболее влиятельный эксперт) до M (наименее влиятельный эксперт). Мнение каждого эксперта учитывается в расчетах с коэффициентом, пропорциональным $1/m_j$, где m_j – ранг j -го эксперта ($j=1, M$).

Для оценки взаимосвязи ранжированных дисциплин учебных курсов вычисляем коэффициент ранговой корреляции Спирмена [3]:

$$\rho = 1 - \frac{6s}{N^3 - N}. \quad (9)$$

Если значение ρ то это характеризует линейную в среднем зависимость между ранжировками по рассматриваемым показателям.

Для рассматриваемых в нашем случае двух показателей это говорит о том, что дисциплины, получившие высокую оценку по содержательности, имеют высокую оценку по методическому уровню.

Далее проводим упорядочение дисциплин по двум показателям качества, выделяя из неравенств (3) и (4) последовательно доминирующие подмножества (множества Парето). Для наглядности дадим

графическую интерпретацию указанного подхода. Для соотношений (3) и (4) на рис. 1 учебные дисциплины представлены согласно табл.1. подмножествами, с центральной точкой, координаты которой определяются местом данной дисциплины среди рассматриваемых дисциплин, согласно его среднего ранга.

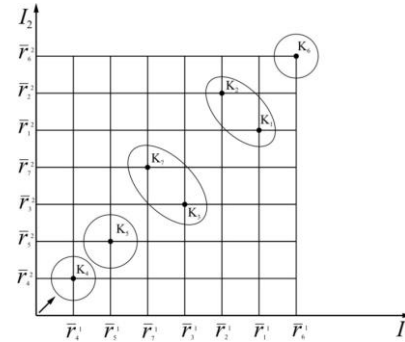


Рис. 1. Графическая интерпретация ранговой корреляции учебных курсов

Будем считать, что показатели качества J_1 и J_2 равноценны, т.е. суммарный показатель качества определяется как:

$$J = J_1 + J_2. \quad (10)$$

В этом случае вектор изменения показателя качества J проходит под углом 45 градусов (рис.1). Тогда, как видно из рис. 1, множество Парето для всей совокупности дисциплин включает только одну дисциплину k_6 . Исключая дисциплину k_6 из рассмотрения, вновь выделяем множество Парето, в которое входят две дисциплины k_1 и k_2 . Каждая из этих дисциплин эффективнее любой другой по обоим показателям, но между собой они несравнимы, поскольку по $J_1: k_1 > k_2$, а по $J_2: k_2 > k_1$. Продолжая эту процедуру получим следующую цепочку предпочтений дисциплин:

$$k_6 > (k_1 k_2) > (k_3 k_7) > k_5 > k_4. \quad (11)$$

Отсюда, если, например, нам необходимо из 7 дисциплин выбрать 5, дисциплины k_5 и k_4 исключаются из рассмотрения. Если же необходимо выбрать 4 дисциплины, то помимо исключения дисциплин k_5 и k_4 необходимо исключить одну из k_3 или k_7 , что можно сделать введением дополнительного показателя. Следует отметить, что при данном подходе при большом количестве количество показателей и дисциплин задача формирования оптимальной вариативной части учебного плана специальности существенно усложняется, что связано с анализом гиперпространства критериев качества.

В идеальном случае многокритериальной задачи можно вести поиск такого решения, которое принадлежит пересечению множеств оптимальных решений всех однокритериальных задач. Однако такое пересечение обычно оказывается пустым множеством, поэтому приходится рассматривать так называемое переговорное множество эффективных решений

(оптимальных по Парето). Критерий оптимальности итальянского экономиста В. Парето применяется при решении таких задач, когда оптимизация означает улучшение одних показателей при условии, что другие не ухудшаются.

Можно выделить следующие основные методы решения задач многокритериальной оптимизации:

1. оптимизация одного, признанного наиболее важным критерия, остальные критерии при этом играют роль дополнительных ограничений;

2. сведение многих критериев к одному введением экспертных весовых коэффициентов для каждого из критериев таким образом, что более важный критерий получает более высокий вес;

3. упорядочение заданного множества критериев и последовательная оптимизация по каждому из них (этот подход лежит в основе метода последовательных уступок) [4].

Наиболее распространенным из этих методов является второй метод, использующий линейную свертку критериев на каждом уровне иерархии. В нашем случае можно выделить два уровня иерархии. На нижнем уровне формируется на основе табл. 1 критерий оценки экспертами i -ой дисциплины по k -му критерию. Обозначим его как E_{ik} . Из табл. 1 видно, что $E_{ik} = \bar{r}_i^k$. Тогда критерий верхнего уровня для оценки i -ой дисциплины по совокупности критериев нижнего уровня будет иметь вид:

$$J_i = \sum_{k=1}^Q \lambda_{ik} E_{ik}, \quad (12)$$

где λ_{ik} – весовые коэффициенты важности k -го

критерия при оценке i -ой дисциплины, которые из табл. 1 определяются как

$$\lambda_{ik} = \left(\sum_{k=1}^Q \bar{r}_i^k \right) / \left(\sum_{i=1}^N \sum_{k=1}^Q \bar{r}_i^k \right) \quad (13)$$

причем

$$\sum_{k=1}^Q \lambda_{ik} = 1, \lambda_{ik} \geq 0. \quad (14)$$

Учитывая целочисленное ранжирование критериев, что приводит к большому диапазону разброса оценок, для более точной оценки предлагается использовать свертку относительных значений критериев i -ых дисциплин относительно их максимальных и минимальных значений [3,5]. Способ свертки состоит в решении задачи минимизации линейной комбинации с неотрицательными весовыми коэффициентами λ_{ik} , обозначающими важность k -го критерия, и удовлетворяющие условию (14):

$$J_1 = \sum_{k=1}^Q \lambda_{ik} (E_{ik \max} - E_{ik}) / (E_{ik \max} - E_{ik \min}). \quad (15)$$

Далее, согласно полученным из (12) значениям критериев верхнего уровня, ранжируем в порядке убывания список дисциплин и отбираем заданное количество дисциплин в вариативную часть учебного плана специальности.

После формирования оптимальной по качественному составу дисциплин вариативной части учебных планов IT-специальностей следующей важной задачей повышения уровня подготовки специалистов является формирование моделей и целевых показателей учебного процесса [6].

Выводы

Предложенный подход к формированию и оптимизации вариативной части учебных планов IT-специальностей носит универсальный характер и может быть использован для решения подобных задач в различных сферах деятельности человека. При этом нет ограничений на количество экспертов, показателей качества и исследуемых объектов. Главная проблема состоит в многокритериальности решаемой задачи. При небольшом числе критериев она решается на основе парной доминантности критериев так, как это показано в данной статье. При большом числе критериев предлагается использовать линейную свертку многоуровневой системы критериев. При этом использование системы Интернет-опроса позволяет подключить широкий круг специалистов-экспертов для повышения достоверной вероятности оптимальности полученных результатов.

ЛИТЕРАТУРА

1. Нейлор К. Как построить экспертную систему: монография / К. Нейлор. – М.: Энергоатомиздат, 1991. – 286 с.
2. Катренко А.В. Теорія прийняття рішень: підруч. / А.В. Катренко, В.В.Пасічник, В.П.Пасько. – Київ: Видавнича група ВНУ, 2009. – 448 с.
3. Бешелев С.Л. Математические методы экспертных оценок: монография / С.Л. Бешелев, Ф.Г. Гурвич. – М.: Статистика, 2010. – 264 с.
4. Добров Г.М. Экспертные оценки в научно-техническом прогнозировании: монография / Г.М. Добров, Ю.В.Ершов. – Київ:Наукова думка, 2008. – 123 с.
5. Орлов А.И. Экспертные оценки: монография / А.И.Орлов. – М.: : Изд-во МГТУ им. Н.Э. Баумана. – 2009. Ч. 2 : – 2011. – 486 с., 2010. – 264 с.
6. Стенін О.А. Моделі і цільові показники учебного процесу в автоматизованих учебных системах / О.А.Стенін, Ю.А. Тімошин, Н.І. Домаскіна, І.П.Шалабай // Адаптивні системи автоматичного управління. –2006. – вип. 9 (29), стор. 140147.

ABSTRACTS

INFORMATION SYSTEMS AND TECHNOLOGIES

Page 11

Creation of robotic system for plasma spraying of implants

Karymsakova Indira, Denissova Natalia
D. Serikbaev East Kazakhstan State Technical
University
Ust-Kamenogorsk, Kazakhstan
indviki@mail.ru, ndenisova@ektu.kz

Krak Iurii
Taras Shevchenko National University
Kyiv, Ukraine
Yuri.krak@gmail.com, Krak@univ.kiev.ua

Abstract. The paper is devoted to exploring characteristics of implants according to physical dimensions, the type of prosthetics of various organs, manufacturing materials and companies, etc., are suggested. The problem of applying biocompatible coatings to implants using robotic complex based on an industrial robot is considered. For the coating application, the technology of plasma spraying and the corresponding equipment of robotic complex are described. Implants for subsequent plasma spraying which correspond to possibilities to maintain biocompatible coating are selected.

Keywords: implant, plasma spraying, robotic plasma deposition plants

Page 14

Module of calibration and positioning of computer vision system for digital cutting of materials

Doroshenko A.Y.
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Tumanov V.V.
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

The principle of operating the calibration and positioning module for the computer vision system used in the digital cutting of materials is described. The nature and influence of optical distortions on the photos of the cutting surface and methods of their elimination were studied. Also the positioning system that guarantees high accuracy of determining the physical location of objects by their position in the photo was developed.

Keywords: computer vision system, digital cutting, OpenCV, camera calibration

Page 17

System for organizing interactive quest-games based on Bluetooth beacons

Andrii Konovalenko
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine
konovalenko.andrii@gmail.com

Abstract. Interactive quest-games system based on Bluetooth-beacons and a mobile application are presented. The mobile application allows a visitor to participate in an interactive quest searching for beacons and answering questions. The system is represented as a Web service that handles and stores data and a hybrid mobile application for a visitor (user). The mobile application is implemented using Angular 4, TypeScript, HTML5 and Ionic Framework 3.

Keywords: *interactive quest, quest-game, entertainment, Bluetooth, hybrid mobile app*

Page 19

Microcontroller for the serial port communications

Anatoliy Sergiyenko
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
aser@comsys.kpi.ua

Maria Orlova,
Alexey Molchanov
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. A microcontroller structure, which is based on the eight-bit stack processor architecture is proposed, and is intended for the FPGA implementation. The microcontroller has small hardware costs, reduced software amount, and flexibility.

Keywords: *stack processor, Forth, FPGA, VHDL*

Page 21

Development of the architecture of automated data collection, processing and analysis system based on Big Data technology

Borys Ryzhko
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Ostap Smolynets
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Abstract. In order to reduce the cost of building high-performance cluster systems, it is proposed to use open software based on Big Data technology. Described method of development of the architecture of automated data collection, processing and analysis systems.

Keywords: data processing, big data

Page 24

Analysis and optimization of autonomous heat supply systems

Stenin A., Pasko V.
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
alexander.stenin@yandex.ua

Shitikova I.
Institute of Telecommunications and Global Information
Space National Academy of Sciences of Ukraine
Kyiv, Ukraine
irinashitikova54@gmail.com

Abstract. This work is devoted to an actual problem of improve the efficiency of operation of heat supply systems through the development and introduction of new resource-saving technologies, optimization of existing and introduction of new designs of heat exchangers. Proposed the mathematical models for calculating the base constructive parameters of heat exchangers. Proposed the dynamical mathematical model, which makes it possible to optimize the parameters of heat processes. Suggested simulation system on base SolidWork gives a wide range of heat processes with a graphical interpretation of the results and allows you to optimize the parameters of heat exchangers online.

Keywords: heat exchanger, parametric identification, dynamic model, simulations

Page 28

Data security in cloud environments

Oleksiy Pyrozhkov
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
a.y.pirozhkov@gmail.com

Olena Savchuk
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
savchuk_11@ukr.net

Abstract. The paper is devoted to the concept of ensuring the security and confidentiality of data in cloud computing, using data-oriented approach. The concept of data-oriented security approach is presented, described the solution to this problem. Access control procedure to cloud resources is presented.

Keywords: cloud security, data security, data-oriented security

CONTROL SYSTEMS

Page 33

The evaluation of the possibility of creating additional propulsion for the control of spacecraft using permanent magnets

Erik Lapkhanov
Institute of technical mechanics
National Academy of Sciences of Ukraine and State Space Agency of Ukraine
Dnipro, Ukraine
ernando@i.ua

Abstract. The article is devoted to the evaluation of the possibility of creating additional propulsion for the control of spacecraft using permanent magnets. The main attention is paid to calculations of the traction force (propulsion) during the interaction of the magnetosphere, created by permanent magnets with a stream of charged particles of the ionosphere and solar wind.

Keywords: spacecraft, permanent magnets, propulsion Earth's magnetic field (EMF), the flow of charged particles

Page 33

On video frame designing of human-machine interface for process control systems

Leonid Yurchuk
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
leonidyu0@gmail.com

Bublinskyi Stanislav
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
bublinsky.s.m@gmail.com

Abstract. The development problems video frame in mimic diagram designing of human-machine interface for process control systems according to the requirements and recommendations of the International Standard ANSI / ISA-101.01-2015-“Human MachinInterfaces for Process Automation Systems” are considered.

Keywords: man-machine interface, life cycle, standardizing organization, ergonomics

PROGRAMMING TECHNOLOGIES

Page 41

High-preformance matrices multiplication in scope of Android NDK and JNI

Illia Achour
Anatoliy Doroshenko
Igor Sikorsky Kyiv Polytechnic Institute
Kiev, Ukraine

Abstract. This work presents high-performance matrices multiplication approach implementation based on Android NDK, JNI for Android OS devices. Approach and its alternative solutions performance was precisely measured, all used features and technics combinations were considered and benchmarked.

Keywords: Android NDK, JNI, affine transformation, OpenGL ES, Java

INFORMATION PROCESSING IN COMPLEX SYSTEMS

Page 47

The Method of an Accumulated Impact for Cognitive Mapping Analysis

Oleh Dmytrenko
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv
dmytrenko.o@gmail.com

Dmytro Lande
Igor Sikorsky Kyiv Polytechnic Institute
Institute for Information Recording NASU
Ukraine, Kyiv
dwlande@gmail.com

Abstract. In this article is introduced a new method of cognitive mapping analysis, that is called the method of an accumulated impact. Showed, that in the proposed method, there is no problem, compared with a widely used method – the impulse method, such as: using the proposed method always give a result regardless of whether impulse process is a stable or not. Unlike the impulse method, the method of an accumulated impact does not violate the scale invariance. Also in this article the advantages introduced method are presented on the examples of analysis of cognitive maps.

Keywords: cognitive map, directed graph, weighted directed graph, adjacency matrix, influence matrix, the impulse method, the method of an accumulated impact

DISTRIBUTED AND PARALLEL COMPUTING

Page 53

Acceleration of Data Exchange in Multiprocessor Systems

Valerii Zhabin
“Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
vizhabin@comsys.kpi.ua

Valentina Zhabina
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
val.zhabina2@gmail.com

Abstract. Methods of inter-processor data exchange in the common bus systems with different data access schemes are analyzed. Possibility to synchronize the procedure of exchange through data bus and control bus is considered. Method for data exchange acceleration based on each processor direct access to other processors local memory addressing space is proposed for the host systems.

Keywords: multiprocessor systems, memory organization, data exchange

Page 57

Design of parallel algorithm for cluster analysis using instrument of design automation

Prykhidko V.S.
Igor Sikorsky Kyiv Polytechnic
Institute
Kyiv, Ukraine

Doroshenko A.Y.
Igor Sikorsky Kyiv Polytechnic
Institute
Kyiv, Ukraine

Yatsenko O. A.
Igor Sikorsky Kyiv Polytechnic
Institute
Kyiv, Ukraine

Abstract. K-means parallel algorithms for cluster analysis were designed using instrument for programs synthesis, which uses an algebra-algorithmic approach for transforming formal specifications of algorithms into program code. The performance of sequential and parallel variations of the algorithm is studied and compared.

Keywords: cluster analysis, cluster, big data, K-means, parallel algorithm

Page 59

Isoefficiency control tool for distributed computing

Heorhii Loutskii
Igor Sikorsky Kyiv Polytechnic
Institute
Kyiv, Ukraine

Artem Volokyta
Igor Sikorsky Kyiv Polytechnic
Institute
Kyiv, Ukraine
artem.volokita@kpi.ua

Pavlo Rehida
Igor Sikorsky Kyiv Polytechnic
Institute
Kyiv, Ukraine
regidapavel@gmail.com

Abstract. This paper describes the implementation of the proposed model for evaluating algorithms for controlling the load isoefficiency of a computer system using the example of two pipelining algorithms. The experimental researches of the model shows the possibility of using this approach for this task. In the paper is proposed the further modifications of the developed model, namely the synthesis of effective topologies and algorithms for a problem of a given dimension.

Keywords: distributed computing, scheduling, isoefficiency

TRAINING IN INFORMATION TECHNOLOGIES FIELD

Page 63

IT-specialty Curriculum Optimization

Stenin A., Pasko V., Lemeshko V.
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine
alexander.stenin@yandex.ua

Shitikova I.
Institute of Telecommunications and Global Information
Space National Academy of Sciences of Ukraine
Kyiv, Ukraine
irinashitikova54@gmail.com

Abstract. Proposed an approach to the formation and optimization of the qualitative composition of the disciplines of the variable part of the curriculum of the specialty, based on methods of expert assessments. Using the Internet survey system allows you to connect a wide range of experts to increase the confidence in the probability of optimality of the results. At the same time, there are no restrictions on the number of experts, quality indicators and disciplines studied. The main problem is the multicriteria nature of the problem being solved. With a small number of criteria, this task may be solved by the pairwise dominance of the criteria. With a large number of criteria, in this article proposed to use the linear convolution of criteria. An integer ranking of criteria leads to a large range of estimates spread. For a more accurate assessment, it is suggested to use the convolution of the relative values of the criteria for the i -th disciplines relative to their maximum and minimum values.

Keywords: optimization, curriculum, Internet Survey, Pareto set, expert assessments
