НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

SUMMER INFOCOM ADVANCED SOLUTIONS 2016

МАТЕРІАЛИ
ІІ МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

CONFERENCE PROCEEDINGS
II SCIENTIFIC AND PRACTICAL CONFERENCE

КИЇВ, УКРАЇНА
1-3 ЧЕРВНЯ 2016 РОКУ

УДК 004

Summer InfoCom 2016: Матеріали ІІ Міжнародної науково-практичної конференції, м. Київ, 1-3 червня 2016 р. – К.: Вид-во “Інжиніринг”, 2016. – 116 с. – Мови укр., рос., англ.

Конференція зареєстрована в українському інституті науково-технічної експертизи та інформації (УкрІНТЕІ). Посвідчення № 197 від 6 квітня 2016 р.
Проведення конференції регламентоване наказом ректора НТУУ “КПІ” № 3-146 від 22 квітня 2016 р.

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 978-966-2344-50-9

ПРОГРАМА КОНФЕРЕНЦІЇ

ПРОГРАМА

Секція 1 / Section 1
Інфокомунікаційні технології/ Infocommunication Technologies

Голова

Ролік Олександр Іванович
д.т.н., проф. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

Секретар

Галушко Дмитро Олександрович
ас. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

1 червня/June

- **Клейменов Р.** Система адміністрування спортивними залами як невід'ємна складова покращення студентського життя
- **Litvinov K.A.** Computer system for controlling of ultrasonic level gauge for liquids with circular motion of ultrasonic pulse
- **Shapovalov O.I.** Method of compensation of temperature errors of magnetostrictive level instrument
- **Степанюк А.І., Полторак В.П**. Модернізація серверного ПЗ для оптимізації віддачі контенту за запитом

2 червня/June

- **Полторак В.П., Микосовський В.І.** Застосування криптографії на еліптичних кривих в смарт-картах
- **Стась Д.О.** Розробка та моделювання Цифрового генератора синусоїдального сигналу при апаратній реалізації на ПЛІ
- **Полторак В.П., Макоївець Д.В.** Передача мультимедійних даних в каналах з низьким співвідношенням сигнал/завада
- **Рудницьких Д.О.** Вдосконалення системи бронювання та пошуку квитків в онлайн сервісах України

3 червня/June

- **Корчагин К.П., Полторак В.П.** Влияние характеристик QoS на качество речевого трафика в пакетных сетях
- **Данчул В.С., Полторак В.П.** Забезпечення інфозахисту команд управління пересувним об›єктом
- **Ференс Д.А., Дорогий Я.Ю.** Структурна оптимізація штучних нейронних мереж
- **Чеповой І.В., Пишняк Д.В., Юрчук Л.Ю.** Система інформаційної підтримки освітнього процесу
- **Мокін В., Варчук І.** Геоінформаційна технологія оптимізації топологічної спостережуваності багатозв'язних просторово-розподілених систем

Секція 2 / Section 2
Системи керування/ Control Systems

Голова

Репнікова Наталія Борисівна
к.т.н., доц. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

Секретар

Дорошенко Катерина Сергіївна
ас. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

1 червня/June

- **Репнікова Н.Б., Шумада К.О.** Синтез спостерігаючого пристрою нелінійної системи керування
- **Писаренко А.В., Тишенко Д.В.** Применение байесовских сетей в задачах диагностики и адаптивного управления
- **Шумейко М.С.** Інформаційні системи керування навчальним процесом у вищій школі

Секція 3/ Section 3
Технології програмування/ Programming Technologies

Голова
Дорошенко Анатолій Юхимович
д.ф.-м.н., проф. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

Секретар
Хмелюк Марина Сергіївна
ас. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

1 червня/June

- **Старушик А.М.** Розробка автоматизованої системи надання та пошуку рекомендацій медіаконтенту на основі вподобань інших користувачів
- **Туманов В.В.** Застосування сучасних засобів opengl в Qt
- **Магдич Б.В.** Система пошуку плагіату для заданої предметної області
- **Верес Д. С.** Генеративне та багатоетапне програмування. Lightweight Modular Staging
- **Вовк Є. А., Март Б. А.** Система активного моніторингу мережі, з відображенням показників у реальному часі

2 червня/June

- **Вальчук Х.І., Дорогий Я.Ю.** Порівняльний аналіз стікості водяних знаків до модифікацій контейнера
- **Прохорова К.С.** .NET Core як потужний інструмент оптимізації програмного забезпечення і спрощення процесу розробки
- **Гончаренко О. Р.** Таблиці пошуку для підвищення швидкості обчислень у комп'ютерній графіці та їх перспективи у майбутньому
- **Сімоненко В.П., Дрегало Т.В.** Система аутентифікації підвищеної надійності

Секція 4 / Section 4
Оброблення інформації в складних системах / The information processing in complex systems

Голова
Теленик Сергій Федорович
д.т.н., проф., зав. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

Секретар
Резник Дмитро Ігорович
ас. каф. автоматики та управління в технічних системах, НТУУ "КПІ"

1 червня/June

- **Lande D.V., Andrushchenko V. B.** Formation of subject area and the co-authors network by sounding of Google Scholar Citations service
- **Галкін О.А.** Дослідження задачі класифікації великих масивів медико-статистичних даних на основі вибірки за значимістю
- **Осідач А.О.** Оцінка ефективності правильності визначення логічної структури документа
- **Сушко С., Чемерис А.** Сравнение эффективности автоматической оптимизации на основе полиэдральной модели
- **Gagarin O.O., Toporivskyi B. P.** Research issues of mining big data streams

2 червня/June

- **Моргаль О.М., Савчук О.В., Латаш І.О.** Використання нечітких множин в технічній діагностиці
- **Жабина В.В., Жабин В.И.** Повышение эффективности мультипроцессорных систем, управляемых потоком дескрипторов данных
- **Свинаренко Д.** Розробка рекомендаційного алгоритму книжок
- **Ріпневський О.О.** Використання гібриду CPU/GPU у криптографічних високопродуктивних обчисленнях

- **Мороз И. Д., Дорогой Я. Ю.** Определение контуров лица методом ограниченного среднего сдвига

3 червня/June

- **Мороз И. Д., Дорогой Я. Ю.** Определение контуров лица методом ограниченного среднего сдвига
- **Дзідзоєв А.Ю.** Дослідження використання фолксономій як інструмента для побудови рекомендаційних моделей в системах соціального тегування
- **Vu Duc Thinh, A.Volokyta, P. Rehida** Access model based on mobile agents for the protection of cloud computing
- **Жабін В.І., Кохан О.С., Токар А. Г.** Реалізація степеневої функції з плаваючою комою в неавтономному режимі
- **Шаповал О.С.** Алгоритм формування рейтингових списків абітурієнтів з урахуванням пріоритетності заяв

ЗМІСТ

Тези конференції 11

Інфокомунікаційні технології 13
Клейменов Роман. Система адміністрування спортивними залами як невід'ємна складова покращення студентського життя 13
Litvinov K.A. Computer system for controlling of ultrasonic level gauge for liquids with circular motion of ultrasonic pulse 15
Shapovalov O.I. Method of compensation of temperature errors of magnetostrictive level instrument 17
Степанюк А.І., Полторак В.П. Модернізація серверного ПЗ для оптимізації віддачі контенту за запитом 18
Полторак В.П., Микосовський В.І. Застосування криптографії на еліптичних кривих в смарт-картах 21
Стась Д.О. Розробка та моделювання Цифрового генератора синусоїдального сигналу при апаратній реалізації на ПЛІС 23
Полторак В.П., Макоївець Д.В. Передача мультимедійних даних в каналах з низьким співвідношенням сигнал/завада 24
Рудницьких Д.О. Вдосконалення системи бронювання та пошуку квитків в онлайн сервісах України 26
Корчагин К.П., Полторак В.П. Влияние характеристик QoS на качество речевого трафика в пакетных сетях 28
Данчул В.С., Полторак В.П. Забезпечення інфозахисту команд управління пересувним об›єктом 30
Ференс Д.А., Дорогий Я.Ю. Структурна оптимізація штучних нейронних мереж 32
Чеповой І.В., Пишняк Д.В., Юрчук Л.Ю. Система інформаційної підтримки освітнього процесу 34
Мокін В., Варчук І. Геоінформаційна технологія оптимізації топологічної спостережуваності багатозв'язних просторово-розподілених систем 37

Системи керування 39
Репнікова Н.Б., Шумада К.О. Синтез спостерігаючого пристрою нелінійної системи керування 39
Писаренко А.В., Тишенко Д.В. Применение байесовских сетей в задачах диагностики и адаптивного управления 42
Шумейко М.С. Інформаційні системи керування навчальним процесом у вищій школі 44

Технології програмування 46
Старушик А.М. Розробка автоматизованої системи надання та пошуку рекомендацій медіаконтенту на основі вподобань інших користувачів 46
Туманов В.В. Застосування сучасних засобів opengl в Qt 48
Магдич Б.В. Система пошуку плагіату для заданої предметної області 50
Верес Д. С. Генеративне та багатоетапне програмування. Lightweight Modular Staging 53
Вальчук Х.І., Дорогий Я.Ю. Порівняльний аналіз стікості водяних знаків до модифікацій контейнера 55
Прохорова К.С. .NET Core як потужний інструмент оптимізації програмного забезпечення і спрощення процесу розробки 58
Гончаренко О. Р. Таблиці пошуку для підвищення швидкості обчислень у комп'ютерній графіці та їх перспективи у майбутньому 60
Сімоненко В.П., Дрегало Т.В. Система аутентифікації підвищеної надійності 63

Оброблення інформації в складних системах 66
Lande D.V., Andrushchenko V. B. Formation of subject area and the co-authors network by sounding of Google Scholar Citations service 66
Галкін О.А. Дослідження задачі класифікації великих масивів медико-статистичних даних на основі вибірки за значимістю 69
Осідач А.О. Оцінка ефективності правильності визначення логічної структури документа 71
Сушко С., Чемерис А. Сравнение эффективности автоматической оптимизации на основе полиэдральной модели 74
Gagarin O.O., Toporivskyi B. P. Research issues of mining big data streams 76
Моргаль О.М., Савчук О.В., Латаш І.О. Використання нечітких множин в технічній діагностиці 79
Жабина В.В., Жабин В.И. Повышение эффективности мультипроцессорных систем, управляемых потоком дескрипторов данных 81

Свинаренко Д. Розробка рекомендаційного алгоритму книжок....85
Ріпневський О.О. Використання гібриду CPU/GPU у криптографічних високопродуктивних обчисленнях....88
Мороз И. Д., Дорогой Я. Ю. Определение контуров лица методом ограниченного среднего сдвига....90
Дзідзоєв А.Ю. Дослідження використання фолксономій як інструмента для побудови рекомендаційних моделей в системах соціального тегування....95
Vu Duc Thinh, A.Volokyta, P. Rehida Access model based on mobile agents for the protection of cloud computing....97

Статті конференції....101

Технології програмування....103
Вовк Є. А., Март Б. А. Система активного моніторингу мережі, з відображенням показників у реальному часі....103

Оброблення інформації в складних системах....107
Жабін В.І., Кохан О.С., Токар А. Г. Реалізація степеневої функції з плаваючою комою в неавтономному режимі....107
Шаповал О.С. Алгоритм формування рейтингових списків абітурієнтів з урахуванням пріоритетності заяв....109

ТЕЗИ КОНФЕРЕНЦІЇ

ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

Administration system of gyms as an inseparable part of student life

Roman Kleimenov
student of NTUU "KPI"
Ukraine, Kiev

Considers the problems of the relevance and necessity of the administration system gyms campus. Showing the main features of the existing system of administration, their advantages and disadvantages. Proposed its own system administration.

Keywords: administration system, web-application, health, gym, campus

Система адміністрування спортивними залами як невід'ємна складова покращення студентського життя

Клейменов Роман
студент НТУУ «КПІ»
Україна, Київ

Розглядається проблема актуальності і необхідності системи адміністрування спортивними залами студмістечка. Показуються основні особливості існуючих систем адміністрування, їх переваги та недоліки. Пропонується власна система адміністрування.

В сучасну еру інформаційних технологій популярність веб-додатків різноманітних типів зростає з кожним днем все більше і більше, адже доступ в інтернет з кожним днем стає все простіше, швидкість з'єднання все вище і все більше людей починають розуміти, що для того, щоб отримати певний товар або послугу за найкоротший проміжок часу зовсім не обов'язково виходити з дому. Використовуючи різноманітні веб-додатки ми економимо життєво важливий час, адже кожен з нас намагається провести більше часу з сім'єю, вивчити щось нове, добре відпочити та поліпшити своє фізичне самопочуття різноманітними способами. З давніх часів фізичні вправи були одним з основних та найдієвіших способів підтримання належного стану здоров'я та самопочуття людини, саме тому відвідуючи спортивну залу хотілося б не витрачати час на оплату послуг та підписання різноманітних угод та контрактів з цим закладом, а більше уваги приділити процесу тренування, зміцнюючи цим самим своє здоров'я.

Більшість студентів має обмежений час для своїх повсякденних потреб та забаганок, тому було вирішено зробити сервіс з адміністрування спортивними залами гуртожитків, який дозволив би студентам витрачати більше часу саме на тренування та зміцнення свого організму, а адміністратору цього закладу – швидше та ефективніше вдосконалювати спортивний інвентар та зал в цілому.

Система адміністрування – інформаційна система або комп›ютерна програма, яка використовується для забезпечення і організації спільного процесу створення, редагування і управління контентом [1].

Оглянувши готові рішення щодо систем адміністрування можна сказати, що даний сервіс можна порівняти з багатьма інтернет-магазинами, кожен з яких має свою адміністративну частину та частину користувача. Відмінності між даними сервісами в основному лежать у способі оплати та в послугах, що надаються.

Одним з найяскравіших прикладів системи адміністрування є інтернет-магазин *rozetka.com.ua*. Даний сервіс дозволяє користувачам придбати різноманітні типи товарів та послуг, використовуючи при цьому зручні для клієнта способи оплати та доставки товару або послуги. Для системи адміністрування спортивною залою сервіс *rozetka.com.ua* є гарним прикладом, але даний магазин надає досить багато функціоналу у клієнтській частині, що не потрібен у системі адміністрування спортивною залою. При опла-

ті абонементів не потрібний «кошик» для товарів, адже дана система надає товар тільки одного типу – абонемент до залу, тобто немає необхідності накопичувати однотипні товари для придбання в один клік. Виходячи з цієї ж причини не потрібно організовувати у клієнтській частині складні багатошарові меню, адже весь функціонал клієнта можна зручно і наочно розташувати на декількох сторінках, створивши зручну та інтуїтивно зрозумілу систему переходів між сторінками. Важливою складовою клієнтської частини сервісу *rozetka.com.ua* є особистий кабінет клієнта, що дозволяє слідкувати за історією придбання товарів, аналогічний функціонал реалізується й системою адміністрування спортивною залою, адже облік грошей, що студент витрачає, в тому числі й на відвідування зали, є досить важливою складовою студентського життя.

Найбільш наближеним до системи адміністрування спортивною залою для гуртожитків є сервіс *www.sportlife.ua*. Даний сервіс надає величезну кількість послуг, що пов'язані напряму зі здоров'ям людини (різноманітні типи абонементів), має зручний інтерфейс та функціональність. Враховуючи потреби і можливості студмістечка, сервіс адміністрування спортивною залою кампусу повинен мати схожий функціонал з сервісом www.sportlife.ua, за виключенням того, що система адміністрування не буде надавати деякі послуги, що присутні у *www.sportlife.ua*.

В результаті проведеного опитування студентів та інших зацікавлених осіб, аналізу існуючих аналогів та предметної області, було прийнято рішення розробити власну систему адміністрування спортивними залами гуртожитків НТУУ «КПІ» що відповідає наступним вимогам:

- клієнтська частина повинна мати зручний інтерфейс для оплати послуг та використовувати електронні платежі;
- клієнтська частина повинна мати функціонал для зміни персональних даних та перегляду статистичних даних;
- адміністративна частина повинна мати весь необхідний функціонал для управління спортивною залою та веб-додатком в цілому (можливість редагування користувачів, контенту веб-додатку, управління електронними рахунками і т.д.);
- додаток повинен мати зручне API для збереження даних;
- додаток повинен мати зручні та наочні грошові звіти як для клієнта, так і для адміністратора;
- додаток повинен бути платформонезалежним.

Поставлена задача була розділена на 3 підзадачі:

- розробка database-layer для збереження даних, якими оперують як клієнт, так і адміністратор;
- створення business-layer для надання даному сервісу бізнес-характеристик, таких як способи нарахування грошей та підключення систем безготівкової оплати;
- створення presentation-layer для зручного та зрозумілого відображення даних системи.

Засоби розробки обиралися відповідно до вимог адміністраторів спортивних залів НТУУ «КПІ», актуальності сучасних технологій, цінових політик сучасних фреймворків та ПО в цілому.

Мовою написання програмного продукту було вибрано Java з огляду на зручність використання єдиної мови програмування на всіх рівнях програмного продукту. На стороні back-end використовується фреймворк hibernate [2] для взаємодії з БД. На стороні front-end використовується технологія jsp [3], що взаємодіючи з проміжними шаром програмного продукту, побудованим за допомогою servlets, демонструє гарні показники надійності та продуктивності.

Фреймворк hibernate був обраний з огляду на те, що система адміністрування спортивними залами має не складну базу даних, внаслідок чого продуктивність запитів до даної бази має гарні показники ефективності та надійності. Використовуючи даний фреймворк, немає необхідності писати рутинні запити до бази даних, всі залежності між таблицями та їх обмеження легко переносяться на java код. Даний фреймворк забезпечує високий ступінь незалежності від конкретної СУБД, а також дуже зручний у взаємодії з MySQL, яку було вибрано в якості сховища даних через простоту у встановленні та використанні, підтримку необмеженої кількості користувачів, що одночасно працюють із БД, безкоштовність, високу швидкість виконання різноманітних запитів та команд, наявність простої та ефективної системи безпеки.

Для створення front-end частини проекту була задіяна технологія jsp з огляду на те, що вона має низьку вартість підтримки проектів, є кросплатформенною технологією, дуже широко поширена на багатьох існуючих проектах, має відкритий вихідний код, а також за даною технологією можна знайти велику кількість готових рішень та бібліотек.

Бізнес-логіка реалізована за допомогою Java Servlet [4]. Сервлет взаємодіє з клієнтами за допомогою принципу запит-відповідь. Великою перевагою технології Java Servlet є швидкість роботи, гарна масштабованість, надійність і безпека, незалежність від платформи, безліч інструментів моніторингу та налагодження і легка інтегрованість back-end та front-end частин.

Також для створення даного проекту було застосовано фреймворк Bootstrap, перевагами якого в даному контексті є простота, безкоштовність, інтегрованість.

В результаті роботи було створено просту у використані систему, що надає необхідний функціонал як для клієнтської, так і для адміністративної частини, має зручний API для збереження даних, що надходять в online-режимі в єдину БД, має інтуїтивно зрозумілий інтерфейс і не залежить від платформи використання. Найближчим часом розроблена система адміністрування буде встановлена до спортзалу гуртожитку №8.

ПЕРЕЛІК ПОСИЛАНЬ

1. ECM Enterprise Content Management, Ulrich Kampffmeyer. Hamburg 2006, ISBN 978-3-936534-09-8.
2. https://ru.wikipedia.org/wiki/Hibernate
3. https://ru.wikipedia.org/wiki/JSP
4. http://java-course.ru/student/book1/servlet/

Рецензент: к.т.н. доц. каф. ТК НТУУ «КПІ» Т.А. Ліхоузова

Computer system for controlling of ultrasonic level gauge for liquids with circular motion of ultrasonic pulse

Litvinov K.A.
post-graduate student
Volodymyr Dahl East Ukrainian National University

The paper considers a new way of measuring the level of liquids by ultrasonic method, which is based on the circular motion of ultrasonic pulse. The level gauge measuring circuit has two ultrasonic transmitters, one of which is located on a float which floats on the surface of a liquid. The level gauge block diagram is presented, and its operating principle is described. The level gauge has a small dead zone, and twice as wide measuring range.

Keywords: ***Ultrasound, level, liquid, measuring, transmitter, environment, gas, control, error, sensitivity***

The ultrasonic method (USM) of level monitoring is based on determination of time in which an ultrasonic pulse (USP) passes the distance from the ultrasonic transmitter (UST) to the liquid surface [1-3]. Level gauges, which are based on this method, have a sufficiently high precision of measurement (from ±0.25%) and a sufficiently broad measuring control range (MCR) (some of them - up to 120 m) [4,5]. The disadvantages should include the considerable dependence of level monitoring on gas environment (GE) parameters, a sufficiently large dead zone (up to 0.6 meters), the influence of internal structural elements and many others [6-7]. The proposed ultrasonic level gauge (USLG) [8] can have one or two piezoceramic transmitters (UST). Main UST1 is located on the top of the reservoir with liquid and excited by an electric pulse (EEP), which is formed by an electric measuring circuit (EMC). Secondary UST2 is located on the top of the float which is floating on the liquid surface. UST2 ultrasonic transmitter is excited by an ultrasonic pulse (USP), which is formed by UST1 ultrasonic transmitter. Main and secondary USTs are connected to each other by an insulated metal towrope. The ultrasonic pulse (USP), which is emitted by the secondary UST2 ultrasonic transmitter, passes through the gas environment (GE) in the reservoir and is perceived by an ultrasonic receiver (USR), which converts these pulses into electromotive force (EMF) with an amplitude proportional to the GE thickness, and the electrical signal goes to the programming microcontroller (PMC) of the computer control and management system, which processes the measurement information signal, determines the amplitude of the perceived ultrasonic signal (USS), calculates the time of movement of the ultrasonic pulse (USP) from UST1 ultrasonic transmitter to USR ultrasonic receiver and transmits information to the real-time monitor for display, and sends it to the database formation unit. Due to the fact that the emitted and perceived channels are divided between themselves and create a corresponding circular shape, the following results are achieved:

- the dead zone of the ultrasonic level gauge (USLG) is reduced almost 5 times due to the absence of a reference mechanical device;
- the measuring control range (MCR) of liquid level is increased up to 2 times due to single passing of ultrasonic pulse through the gas environment;
- the precision of liquids level measurement is increased almost twice due to: a decrease in the distance of passing of an ultrasonic pulse (USP) in the gas environment (GE) by 2 times; a decrease in the secondary ultrasonic effects inside of the reservoir, and the absence of a reference device on the path of an ultrasonic pulse (USP) in the GE.

Ultrasonic level gauges with two ultrasonic transmitters (UST) are designed for measuring control of fuel level in warehouses, refueling stations, etc., as well as other highly inflammable liquids, because ultrasonic pulses of mechanical origin propagate through liquid and gas environment. In cases when nonflammable liquids level is measured, ultrasonic level gauges (USLG) can be built with one ultrasonic transmitter (UST), which is located on the float. A significant advantage of the USLG with circular motion of ultrasonic pulse (USP) is the absence of contact of this pulse with the liquid surface. Fig. 2.1 presents the scheme of an ultrasonic level gauge (USLG), on which the following is marked: 1 – UST1 primary transmitter, UST2 secondary transmitter, USR receiver, CU control unit and IDU information display unit.

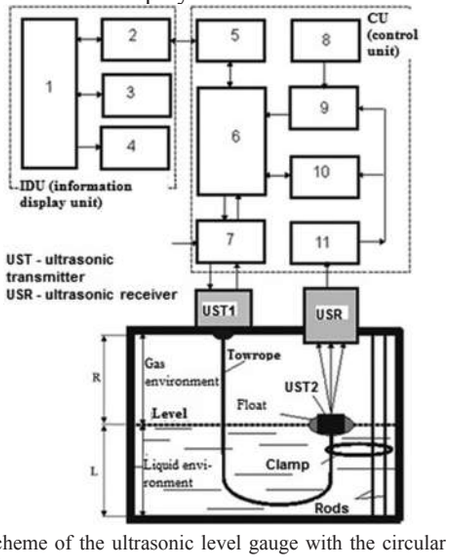


Figure 1 - Scheme of the ultrasonic level gauge with the circular motion of ultrasonic pulse

Measuring of liquid level in the reservoir is performed by means of the secondary UST2 which emits ultrasonic pulses (USP) and transmits them through the gas environment (GE) with R thickness to the ultrasonic receiver (USR). Since the distance from the primary UST1 to UST2 is constant and determined by the towrope length, and the distance from UST2 ultrasonic transmitter to USR ultrasonic receiver is determined by R thickness, then the time of ultrasonic pulse (USP) passing of this distance will depend on L liquid level. Since the UST1 output signal is proportional to the UST2 signal, and the time of USP passing along the metal towrope is constant and much less than the time of USP passing through the gas environment (GE), then this time has almost no effect on the measurement result. The time by which the liquid level is determined, consists of τ_1 time of feeding of a single electric pulse (EEP) to the piezoceramic element (PCE1) of UST1, τ_2 time of ultrasonic pulse (USP) formation by the first transmitter, τ_3 time of USP passing along a metal towrope to UST2, τg time of USP passing of the gas environment (GE), and τ_4 time of perception by the ultrasonic receiver (USR), transformation into voltage and amplification of the latter. UST2 ultrasonic transmitter has receiving and emitting metal membranes in its structure, between which there is a PCE2 which serves as a receiver-transmitter of initial ultrasonic pulses (USP). Due to single passing by USP of R distance in the GE, the liquid level measurement error, which is caused by changes in temperature, pressure and composition, is reduced almost twice. Since ultrasonic pulse (USP) weakening in the GE decreases almost twice, then the level measurement range is increased by the same amount compared to an analog. The computer measuring system consists of the following two units: the control unit (CU) and the measurement information display unit (IDU). The IDU includes: microcontroller (MC) 1; signal conversion unit 2; memory unit 3 and real-time monitor 5. The control unit includes: signal conversion unit 5; managing microcontroller (MMC) 6; adjustable source of excitation pulses (ASEP) 7; reference-voltage source (RVS) 8; comparator 9; USR EMF amplitude determining unit 10 and amplifier 11. The ultrasonic level gauge (USLG) operates in the following way. After its putting into operation, managing microcontroller (MMC) 6 produces a control signal to adjustable source of excitation pulses (ASEP) 7, which forms and produces a single electric pulse (EEP) to UST1 ultrasonic transmitter. Simultaneously, the timing-pulses counter (TPC) resets to zero and turns on. At that, UST1 generates an ultrasonic pulse (USP) at maximum amplitude, which goes to UST2 along the insulated metal towrope. The latter creates USP, which is sent to the GE. After passing through the GE, this USP is perceived by the ultrasonic receiver (USR), in which EMF with $\mathring{a}_R$ voltage is created. The latter, after amplification in amplifier 11, is fed to comparator 9, where it is compared with a pre-set $\mathring{a}_{0R}$ reference voltage, which is formed by reference-voltage source (RVS) 7. If the difference in these voltages is not zero ($\mathring{a}_R - \mathring{a}_{0R} \neq 0$), then comparator 9 produces a signal to the adjustable source of excitation pulses (ASEP), which reduces or increases the amplitude of the single electric pulse (EEP) until $\mathring{a}_R - \mathring{a}_{0R} = 0$ equation is valid. When $\mathring{a}_R - \mathring{a}_{0R} = 0$ then the comparator permits managing microcontroller (MMC) 6 to turn the timing-pulses counter (TPC) on and calculate the liquid level value. The timing-pulses counter (TPC) is working until the perceiving signal of the next level-measuring cycle comes to the comparator, and the measured time, the calculated level value, the temperature of the GE and the liquid, as well as the pressure in the reservoir and the atmospheric pressure are displayed on the monitor screen 4. The absolute error of the measurement control is also displayed on the monitor screen.

LITERATURE

1. Zhdankin V.K. Ultrasonic sensors for control systems [articles] // Modern automation technology. - Moscow: STA-PRESS, 2003. – No.1. - P. 68 - 79; No.4. - P. 48 - 62.

2. Pat. 2004/088253 VOIS, IPC G01F, G01F296. Level measuring device operating with ultrasound / Eckert Manfred (Germany), Faber Harald (Germany), Spanke Dietmar (Germany); Endress+Hauser GmbH+Co.KG (Germany) and others. – No. EP2004/003405; appl. March 31, 2004; published on October 14, 2004.

3. Ultrasonic converters / Edited by E. Kikuchi, translated from English. - Moscow: Science, 1972. - 386 p.

4. Stencel I.I., Thomson A.V., Shapovalov A.I., Litvinov K.A. MEASURING LEVEL OF LIQUID MEDIUMS WITH IRREGULAR SURFACE BY THE ULTRASOUND LEVEL CONTROL DEVICE. “Development of Scientific Research 2012”: Proceedings of the Eighth International Scientific and Technical Conference, Poltava, November 19-22, 2012, p. 78-81

5. Babikov O.I. Level control using ultrasound. - Leningrad: Energy, 1971. - 98 p.

6.Siemens. Control and Measuring Instruments. Level: Catalogue FI 01 / Siemens AG. - 2007. - 188 p.

7. Vzliot ultrasonic flow meter. Digital ultrasonic level gauge [Electronic resource]: Catalogue: Products / “Vzliot” CJSC. – Access mode: www.vzljot.ru/catalogue/84/ - Name from the title screen.

8. Patent of Ukraine for invention No. 110220 “The ultrasonic device for liquid media level monitoring”. Bulletin No. ...

9. Patent of Ukraine for utility model No. 103916 “The device of liquid media level monitoring with circular motion of an ultrasonic pulse”. Bulletin No. ...

Рецензент: д.т.н., проф., зав. каф. комп’ютерно-інтегрованих систем управління Східноукраїнського національного університету ім. В. Даля Й. І. Стенцель

Method of compensation of temperature errors of magnetostrictive level instrument

Shapovalov O.I.
PhD student
East Ukraine Volodymyr Dahl National University

This work considers factors that cause temperature measurement errors of the magnetostrictive level instrument. It considers the ways to reduce temperature errors caused by simultaneous change of electrical and mechanical parameters of the waveguide. There is also a block diagram of the method of compensation of temperature errors and the description how it works.

Keywords: ***magnetostriction, waveguide, domains, ultrasound, measurement, compensation, error.***

One of the important disadvantages of the known magnetostrictive level instruments (MLI) is the inadequacy of compensation of measuring temperature control since it is a non-linear error [1-4]. Besides, the effect of gas environment (GE) and liquids temperature is not considered, since the influence can be different, for example, in summer and winter seasons, or during the night and day and daily changes affect the electrical resistance of the waveguide, which is located in two environments. As the theoretical and experimental studies [5] proved, the change in temperature greatly affects the length l_P of the waveguide, which is changed according to the formula

$$l_P = t_P v_0 \sqrt{1 + 2\frac{\alpha_l(\dot{O}_N - \dot{O}_0)}{1 + \alpha_l(\dot{O}_N - \dot{O}_0)}}, \quad (1)$$

where t_P - is the time of the movement of electric current impulse (ECI) from the generator to the electromagnetic transformer (EMT), which floats on the surface of the liquid;

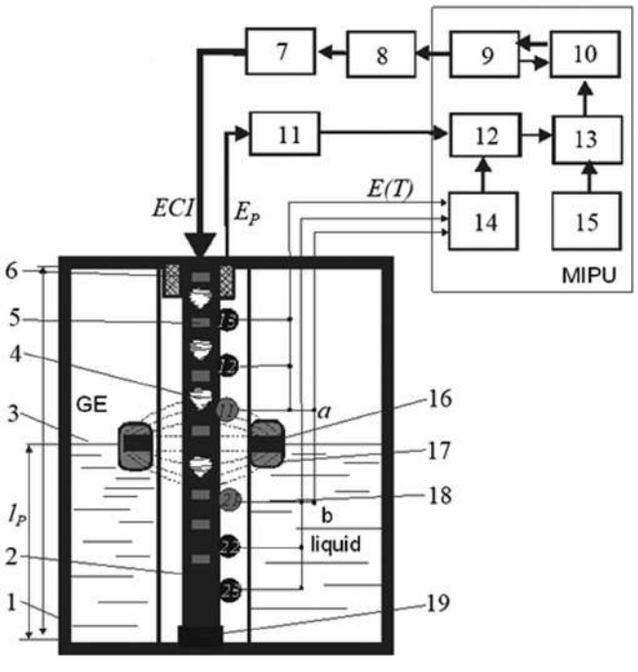
v_0 - the velocity of ECI movement along the magnetostrictive waveguide;

α_l - the temperature coefficient of the waveguide elongation;

$\dot{O}_N$ - the temperature of the environment where the waveguide is located;

$\dot{O}_0$ - nominal environment temperature.

Reducing the temperature error is made due to the time of ECI passing of the distance from the generator ECI to EMT, which is on the surface of the liquid, and ultrasonic torsion pulse (UTP) from EMT to ultrasound receiver (USR) with compensation of the nonlinear temperature influence and compensation of the temperature error caused by uneven GE and fluid temperature changes [6,7]. Unlike gauges, whose temperature sensors are located along the length of the waveguide, and amendment to the temperature error of normalized values is formed according to the average readings of the sensors [6,7], the proposed MLI temperature sensors are located along the waveguide so that near the float with EMT at least two temperature sensors are located, one of which measures the temperature of the GE, and another – of the fluid. According to output signals of the measurement information processing unit (MIPU) the average of the temperature is calculated, and subsequently - the adjusting signal to the measurement result. This allows to reduce the temperature error of MLI by 2 times due to the influence of the temperature of the GE on the work of EMT, reduce the temperature error of MLI due to the influence of GE and fluid temperature by 2.5 times on the change of the active resistance of the waveguide and reduce static nonlinearity characteristics when the temperature of the liquid is from minus 40^{0}C up to 120^{0}C. Fig. 1 shows the MLI circuit, namely: 1 - the technological apparatus; 2 - waveguide; 3 - the surface of the liquid; 4 - UTP; 5 - ECI; 6 - UTP receiver; 7 - ECI amp; 8 - generator; 9 - microprocessor (MP) 10 - meter clock; 11 - UTP receiver amplifier; 12 - adder; 13 - comparator; 14 - signal processing unit of the thermocouples (SOUT) 15 - reference voltage source (RVS) 16 - permanent magnet; 17 - float; 18 - protective tube; 19 - shoe.



Picture 1 - MLI with compensation of temperature error

Unlike others, the proposed MLI uses chromel-copel thermocouple that are evenly placed along the length of the waveguide so that the distance between both of them in GE and in the fluid is the same and equal to the amount of 5 to 20 percent depending on the length of the waveguide. All thermocouples

are arranged in certain chain and periodically they are subject to the survey, and their signals in the form of thermal electromotive force $E(T)$ in the proper order are written down in the block 14. Thermocouples 11-13 are located in the GE and 21-23 thermocouples are located in liquid environment. Thus separately thermocouple signals are formed in GE and in liquid. Thermocouple signals of 11 and 21 are sent to SOUT to form amendments due to the changes in shift module, material density and its linear waveguide extension. Signals of the thermocouples 11-13 and 21-23 form the amendment due to the temperature change of the active resistance of the waveguide.

The number of thermocouples that are located in GE and liquid depend on the level. The countdown of the thermocouples begins on the position of the float of the EMT, i.e. the level towards GE or liquid. Thermal electromotive force $E_{1\hat{A}}(\dot{O})$ of the thermocouple that is in GE and $E_{1D}(\dot{O})$ of the thermocouple that is in fluid are coming to MIPU, where their difference is determined $\Delta E_1(\dot{O}) = E_{1\hat{A}}(\dot{O}) - E_{1P}(\dot{O})$. Based on the results of measuring the temperature with thermocouples thermoelectric averages are determined. According to the calculated average values $\overline{E}_{\hat{A}}(\dot{O})$ and $\overline{E}_{D}(\dot{O})$the average temperature of the waveguide is determined. Since during the calibration of MLI the GE and fluid temperature, and accordingly the waveguide were equal to some normalized temperature $\dot{O}_{\delta 0}$, the temperature deviation of the waveguide, which causes the temperature error:

$$\Delta\overline{T}_{\delta} = T_{x0}\left[\left(1 - \overline{\delta}_{\hat{A}}\right) - \overline{\delta}_{C} l_P / l_x\right], \qquad (2)$$

where $\overline{\delta}_{\hat{A}} = \Delta\overline{T}_{\hat{A}} / T_{x0}$ - relative average change in GE temperature;

$\overline{\delta}_{C} = \Delta\overline{T}_{C} / T_{x0}$ - relative change in the average temperature of the liquid.

-------ffff----------

According to the calculated value of temperature error $\Delta\overline{T}_{\delta}$ the correction is formed as an electrical signal that goes to MIPU. MLI works as follows. After MLI starts working MP 9 issues control signals to the generator 8 which generates ECI. After amplification in the amplifier 7 ECI is sent to the input waveguide 2. Simultaneously the electric measuring circuit (EMC) is lowered to the zero state and meter clock (MC) 10 is switched on. Thus, ECI, moving along the waveguide 2 at the speed v_0 reaches the area of magnetic field of the permanent magnet 16, which is floating on the surface 3 of the float 17. At the same time the electrodynamic force (EDF) appears in EMT, which changes the position of the waveguide material domains. Once the ECI passes the area of the magnetic field of the EMT, the domains start the damped oscillatory process with ultrasonic frequency (USF), thus forming UTP 4. Since the vibrations of the domains take place in the magnetic field of a permanent magnet, there is also an induction current with USF. Since UTP is repelled by the magnetic field, its motion is opposite to the movement of the ECI. This means that the UTP momentum returns back to the beginning of the waveguide 2. At its entrance a USR receiver 6 is located, which converts UTP in electrical voltage which goes to the input of the amplifier 11 and then into the adder 12. The latter is designed to sum up USR signal and temperature probes signal corrections which is formed with the block 14. From the unit adder the value of the measuring signal is sent into the comparator 13, where it is compared with a reference voltage - RVS 15. In case of equality of these voltages MC 10 stops its operation. According to the number of measures impulses the time is determined by which the current value of the liquid is calculated. After the received measurement result is saved in MIPU, a new measurement cycle starts.

LITERATURE

1. Catalogue Fl 01 firm Siemens. Control and Measuring instruments. Level. 2007. - 188 c.

2. Ultrasonic transducers / edited by E. Kikuchi, translated from English .. - M .: Nauka, 1972. - 386 p.

3. Shapovalov O.I. Mathematical model magnetodynamic flow in the area rheological transition magnetostrictive transducers. Herald of Khmelnytsky National University: Engineering. Khmelnitsky, 2014, №2 (211). - p. 240-245.

4. Stenzel Y.I., Thomson A.V., Shapovalov A.I. Analysis of magnetostrictive liquid level controls environments. East European Journal of advanced technologies. Kharkov, № 3/5 (45) 2010. - p. 53- 56.

5. Stenzel Y.I., Shapovalov A.I. Experimental study of ultrasound signals magnetostriction means to control the level of liquid media. Bulletin of National Technical University "KPI". Proceedings of "Power and transformational technology." №12. 2010. - p. 15-21.

6. Ukraine patent for utility model UA 98707 UMPK G01F 23/28 (2006.01). Magnetostrictive means to control the level of liquid media / Stenzel Y.I., Shapovalov A.I., Ryabichenko A.V., Leprosy O.I. .; Appl. 09/19/2014; Publish. 05/12/2015. Bull. №9.

7. Stenzel Y.I., Shapovalov A.I., Thomson A.V., Yanishyna A.S. Basic theory of magnetostriction means to control the level of liquid media. Proceedings of the National Technical University "KPI". Collected Works. "Electricity and preobrazovatelnaya technics." - Kharkov: NTU "KPI" - №19. 2011, p. 45-54.

Рецензент: д.т.н., проф., зав. каф. комп'ютерно-інтегрованих систем управління Східноукраїнського національного університету ім. В. Даля Й. І. Стенцель

Improving server settings software to optimize content return by request

Stepanyuk A.I.
Ukraine, Kyiv

Poltorak V.P.
Ukraine, Kyiv

Annotation: In theses analyzed the influence of certain criteria server software settings and their effect on the rate of return for content. The situation was simulated load on servers apache-server and nginx- server modified main options in database and monitor their impact on the processing of requests for selected criteria - speed page load

Tags: Web-server, nginx, apache, improving server settings, page speed load improving.

Модернізація серверного ПЗ для оптимізації віддачі контенту за запитом

Степанюк А.І.
Україна, Київ

Полторак В.П.
Україна, Київ

Анотація: В тезах був проаналізований вплив деяких критеріїв налаштування програмного забезпечення серверу та їх вплив на швидкість віддачі контенту за запитом. Була змодельована ситуація навантаження на сервери apache-сервер та nginx-сервер змінено основні параметри в налаштуваннях бази даних та моніторинг їх впливу на обробку запитів за обраним критерієм – швидкість завантаження сторінки.

Ключові фрази: Веб-сервер, бази даних, nginx-сервер, apache-сервер, модернізація серверного ПЗ, покращення швидкості завантаження сторінки.

ВСТУП

В процесі розробки високонавантаженого сервісу з передачі медіа, а саме аудіо та відео потоків в режимі реального часу постала проблема швидкості завантаження контенту зі сторони користувача. Було вирішено провести дослідження які параметри налаштування ПЗ необхідно змінити для більш швидкої віддачі контенту за запитом.

Дослідження проводились таким чином щоб всі користувачі знаходились в Україні. Дев'яносто п'ять відсотків навантаження на сервери імітувалось скриптами. Інші п'ять відсотків реальні користувачі. Отже загальна кількість умовних користувачів на сайті було 10 тисяч. Середня кількість звернень до БД та apache-серверу ~ 1000.

Отже на нашу думку у використанні високо навантаженого проекту значну роль відіграє налаштування серверів, та ПЗ, а надто у сервісу типу онлайн стріму – система що віддає відео та аудіо зображення в режимі реального часу, та й являється одночасно хостингом відео, з можливістю його онлайн перегляду. Умовно поділимо множину файлів нашого проекту на складові частини, яких дві: статичний та динамічний контент. Статичний контент обробляється nginx-сервер – веб сервером, який є високо продуктивний, та буде займатися віддачею статичного контенту, а саме: картинок, файлів формату html та css, аудіо та відео файлів. Динамічним контентом або бекендом всього сайту, що складає переважно файли формату php, буде оброблятися apache-сервером. Таке розподілення обов'язків знімає з apache-сервера усе навантаження по запитах статичного контенту адже передає його nginx-серверу який в свою чергу їх обробляє поки apache-сервер обробляє динамічний контент.

Всі користувачі сайту використовують ресурси сервера, переглядають відео, або слухають онлайн лекції, це створює значне навантаження на сервер та серверну базу. Apache-сервер та база даних БД (в нашому випадку MySQL 5.5) постійно змагаються за серверні ресурси.

Отже умовно будемо вважати що кожним розділом обробки даних займається свій сервер – обробкою php файлів та динамічного контенту буде займатися apache-сервер, а обробкою статичного контенту, html, css файлів, картинок та відео буде займатися nginx-сервер.

Умовно розділимо статичний контент на дві частини, це необхідно для того щоб краще зрозуміти які параметри краще редагувати:

««Легкий» контент на який припадають файли типу html, css, js, xml, rss, txt – добре піддається стисканню, отже будемо стискати їх використовуючи функцію gzip.

«Важкий» контент відео та аудіо-файли. Такі типи файлів значно залежать від дискової системи, кількості оперативної пам'яті, та пропускної властивості каналу, адже вони великі за розміром, тому необхідні велика швидкість їх передачі.»[2] Задача роздачі такого типу контенту поділяється на дві підзадачі – зберігання та роздача контенту. В випадку з нашим проектом, проект займається роздачею відео, та передачі медіа в режимі реального часу отже основне звернення буде саме до важкого контенту - відео, тобто необхідно швидко обробити великі об'єми інформації та зберегти на диск, та так само швидко віддати за запитом користувача. Додатково підключаємо ще один резервний канал зі швидкістю 1 ГБ\с (резервний канал), що спілкується з іншим сервіс провайдером. Підключення налаштоване за допомогою динамічного протоколу маршрутизації BGP, що дає змогу за замовчення використовувати обидва канали, і відповідати на запити по тому каналу звідки прийшов запит. Налаштовуємо на сервері RAID масив 10 з чотирьох дисків. Обрали саме 10-й рейд масив адже він являється найнадійнішим варіантом для зберігання даних, це зумовлено тим що запис даних на цю систему відбувається послідовно на декілька дисків. Швидкість зчитування, та запису даних на диск є високою [1].

Конфігурація тестових серверів обрана з доступних варіантів. Отже кінцевий варіант конфігурації серверу наведений в таблиці 1.

Табл. 1 – Набір характеристик тестового серверу.

Характеристика	*Значення параметру*
Процессор	3.2 Ггц 4 потоки
ОЗУ	32 GB
Пам'ять	4xSATA 128 GB
Рейд	10
Пропускна здатність каналу	100мб
Швидкість підключення	1 гб\с

Критерієм оцінки якості обрано швидкість завантаження сторінок, адже швидкість віддачі контенту за запитом це показник оптимізації серверного забезпечення та сервера для користувача. Показник високої швидкості віддачі контенту сторінки користувачу єдине що його цікавить. Чим менше користувач чекає на необхідну йому інформацію тим краще. На момент проведення дослідження показники швидкості віддачі контенту наведені в таблиці 2 без попередньої оптимізації серверного ПЗ.

Табл. 2 – Середні показники швидкості завантаження сторінки вебсайту.

Характеристика	*Швидкість (мс)*
Сторінка з відео	400 ms
Сторінка зі стрімом	465 ms
Сторінка з контентом	225 ms

Необхідна оптимізація серверного програмного забезпечення, в даному дослідженні полягає в тому щоб покращити швидкість віддачі контенту за запитом використовуючи обрані нами потужності серверу.

Під необхідною оптимізацією будемо розуміти покращення значень першочергових критеріїв оцінки продуктивності – тобто ми будемо редагувати налаштування які впливають на об'єми обробки інформації одночасно, кількості запитів до сервісів нашого серверу, відповідно до потужностей нашого серверу, оптимальної кількості користувачів та їх середньої кількості запитів.

Дослідження системи проводилось на реальному сервері шляхом зміни наступних параметрів які впливають тим чи іншим чином на ПЗ щодо звернення до нього за запитом.

Оптимізація бази даних MySQL полягає у тому щоб зупинити значний потік запитів до БД який може не витримати навантаження, а, також, в оптимізації віддачі запитів типових запитів, а саме зберігання таблиць в кеші і їх миттєвої віддачі за наступним таким самим запитом.

Всі показники змінювалися відповідно до потужностей серверу, та кількості відвідувачів під час нашого імпровізованого навантаження наступним чином:

back_log - кількість з›єднань, які можуть знаходиться в черзі до того моменту, як сервер перестане відповідати на нові запити. Так як проект вважається високо навантаженим, то обираємо 6000 звернень. Як показують дослідження більша кількість звернень до нашого серверу погіршує швидкість обробки запитів.

max_connections = 1700 - скільки підключень може бути прийнято сервером, оптимальний показник який ми використовуємо відповідно до потужностей серверу та кількості користувачів.

table_cache = 3096 Відкриття таблиці вимагає деяких ресурсів, отже цей параметр відповідає за кількість відкритих таблиць які очікують наступного з'єднання деякий час після виконання останнього.

tmp_table_size = 64M Параметр відповідає за максимальний розмір тимчасової таблиці, що вміщується в пам›яті. Якщо таблиця його досягає вона зберігається на диск. Отже необхідно намагатися що б таблиць на диску створювалося як можна менше.

thread_cache_size = 4096 Максимальна кількість потоків, які залишаються для повторного використання після виконання запиту. Корисно тримати достатнім для того що б MySQL якомога менше робив нових потоків і використовував старі.

Частина оптимізуючих налаштувань що відноситься до налаштувань Nginx-сервер виконувались наступним чином:

Gzip стискання допомагає швидше передавати дані статичного контенту шляхом зменшення ваги файлів що в свою чергу пришвидшує їх передачу. Отже включаємо gzip стискання для нашого проекту.

```
"gzip_static on:
    location /js/ {
gzip_static on;
 /var/www/metida.ua/view/js
}
```

Вмикаємо online пакування динамічних файлів:

```
    location / {
gzip on;
gzip_min_length 1100;
gzip_buffers 16 8k;
gzip_comp_level 3;
gzip_types text/plain application/xml application/x-javascript
text/css;
/var/www/metida.ua/view/js
} "[3]
```

Після проведення вказаних вище маніпуляцій з серверним ПЗ ми отримуємо наступні показники швидкості завантаження сторінок, що наведені у Табл. 3.

Табл. 3 - Оптимізовані середні данні швидкості віддачі контенту за запитом

Характеристика	*Швидкість (мс)*
Сторінка з відео	340 ms
Сторінка зі стрімом	400 ms
Сторінка з контентом	200 ms

ВИСНОВОК

Проведене дослідження показало, що серверне ПЗ, оптимізоване належним чином, дає змогу підвищити швидкість обслуговування клієнтів у сенсі видачі відео контента на 15% у порівнянні з не оптимізованими налаштуваннями ПЗ сервера.

Програмна оптимізація серверу являється невід'ємною частиною для високо навантажених проектів, вони дозволяють більш гнучко використовувати доступні

потужності, та досягати значного покращення результатів зі швидкості доступу до різних частин сайту при невеликих обсягах тієї ж самої потужності серверів.

ПЕРЕЛІК ПОСИЛАНЬ

1. RAID [Електронний ресурс] / Вікіпедія – 2016 – Режим доступу: https://ru.wikipedia.org/wiki/RAID#RAID_0.2B1

2. Черный О. Тюнинг nginx-сервер [Електронний ресурс] / Черный Олег // Хабрахабр – 2009 – Режим доступу: https://habrahabr.ru/post/56497/

3. Ставинский А. Оптимизация связки Nginx-сервер, Apache, PHP, MySql [Електронний ресурс] / Ставинский Антон // Хабрахабр – 2012 – Режим доступу: https://habrahabr.ru/post/146179/

4. Nick Kew / The Apache Modules Book. Application development with apache / Nick Kew; - Indiana – 2007 – 558c.

Use of elliptic curve cryptography in smart-cards

Vadim Poltorak
ACTS NTUU "KPI"
Ukraine, Kyiv

Vitaliy Mykosovskyy
ACTS NTUU "KPI"
Ukraine, Kyiv

Summary

Argumentation the possibility of using elliptic curve cryptography in systems with limited processing power. Description of possible advantages of using elliptic curve cryptography in smart-cards.

Keywords: ***elliptic curves, elliptic curve cryptography, smart-cards, RSA.***

Застосування криптографії на еліптичних кривих в смарт-картах

Полторак Вадим Петрович
к.т.н., доцент кафедри АУТС НТУУ "КПІ"
Україна, Київ

Микосовський Віталій Ігорович
студент кафедри АУТС НТУУ "КПІ"
Україна, Київ

Анотація

Аргументація можливості використання криптографії на еліптичних кривих в системах з обмеженою обчислювальною потужністю. Опис можливих переваг застосування еліптичної криптографії в смарт-картах.

Смарт-карти є одними з найбільш широко застосовуваних електронних компонентів сьогодення. Смарт-карти застосовуються для різного роду посвідчень, кредитних карт, електронних квитків, паспортів і т.д. Смарт-карта це мікрокомп›ютер, що здатний зберігати дані і запускати команди. Він має розміри не більше 25мм і розміщений на пластиковій карті, розміром як стандартна кредитка[1]. Смарт-карти стійкі до механічного втручання, а дані, що зберігаються зашифровані і захищені від зловмисників. На відміну від карт з магнітною стрічкою, смарт-карти можуть не тільки зберігати дані, але й виконувати обчислення, таким чином вони не потребують доступу до віддаленої бази даних під час транзакції.

Вони можуть застосовуватись для автоматизованих електронних транзакцій. Їх непросто скопіювати, тому застосування смарт-карт робить операції більш захищеними. Також смарт-карти можуть зберігати дані і містити ряд захисних функцій і алгоритмів.

Смарт-карти портативні і можуть бути застосовані в багатьох програмах та сервісах, і зменшити витрати в порівнянні з системами заснованими на використанні імені користувача та паролю.

Для шифрування даних смарт-карти використовують алгоритм RSA[2]. Оскільки смарт-карти обмежені в обчислювальній потужності, а вимоги до криптозахисту даних ростуть з кожним днем, необхідне нове рішення для збільшення криптостійкості даних. Збільшення криптостійкості, при використанні RSA потребує збільшення обчислювальної потужності смарт-карт. Це призведе до збільшення собівартості карток. Для вирішення цієї проблеми пропонується застосувати криптографію на еліптичних кривих [5].

Завдяки властивостям еліптичних кривих, криптографія на них дозволяє значно зменшити довжину ключа, в порівнянні з застосовуваними зараз алгоритмами, не втрачаючи при цьому криптозахищеності. Застосування еліптичної криптографії в смарт-картах має наступні переваги [3]:

- Масштабованість. Зі збільшенням вимог до надійності захисту даних, еліптична криптографія дозволяє значно збільшувати криптостійкість, при незначному збільшенні розміру ключа. Що дозволяє забезпечувати

вищі рівні захисту, без модифікації самої смарт карти, а отже, без збільшення її вартості.

• Менший час операцій та менші затрати пам'яті: Ключі невеликих розмірів потребують менше пам'яті, також менше даних передається між картою і програмою, тому операції проходять швидше.

• Відсутність сопроцессора: Так як еліптична криптографія не потребує значної обчислювальної потужності, для неї не потрібний спеціальний крипто-сопроцессор, що дозволяє зменшити вартість смарт-карти на 20-30%

Для застосування еліптичної криптографії в смарт-картах пропонується наступний алгоритм. В якості зовнішніх параметрів вибираються еліптична крива E і точка P високого порядку із групи точок E(F) [2].

Абонент A(термінал) вибирає секретний ключ $k_A \in Z^*_N$ (мультиплікативна група кільця лишків за модулем N), обчислює і оголошує свій відкритий ключ (E, P, Y), де $Y = k_A P$.

Абонент B(смарт-карта) для передачі абоненту A секретного повідомлення m

• отримує авторизовану копію відкритого ключа (E, N, P, Y);

• «вкладає» повідомлення m в точку $M \in E(F)$ за певним алгоритмом[4];

• вибирає випадкове число $r \in Z^*_N$ (процедура рандомізації);

• обчислює сеансовий ключ $\Delta = rY$;

• обчислює криптограму $C = (C_1, C_2) = (rP, M + \Delta)$;

• відправляє криптограму C абоненту A.

Для розшифрування криптограми абонент A, використовуючи свій секретний ключ k_A,

- обчислює $k_A C_1$ (отримує в результаті $k_A rP$), інвертує цей результат [5];

- складає отриману точку $-k_A rP$ із точкою C_2, відновлюючи точку M:

$$-k_A rP + M + \Delta = M + rk_A P - k_A rP = M.$$

Криптоаналітику(потенційному зловмиснику) відомі відкритий ключ (E, P, Y) і криптограма (C_1, C_2), таким чином для отримання точки M йому необхідно обчислити точку $k_A rP$. Для цього йому доведеться вирішити задачу Діффі-Хелмана для еліптичної кривої[2]: знайти цю точку за відомими точками rP та $Y = k_A P$, або йому прийдеться вирішувати задачі дискретного логарифмування, обчислюючи секретний ключ k_A та рандомізатор r за точками $k_A P$, rP і P відомої йому еліптичної кривої E та відомому йому порядку точки P.

Слід зазначити, що повторне використання рандомізатора r недопустимо [4]. Якщо криптоаналітику вдалось розшифрувати одну криптограму (C_1, C_2) чи дізнатись точку M іншим способом, то він з легкістю отримає і інші повідомлення, зашифровані тим же рандомізатором r, що і криптограма (C_1, C_2). Перші точки в цих парах однакові, а другі зв'язані співвідношеням

$$C'_2 = C_2 - M + M'.$$

Тому $M' = C'_2 - C_2 + M$ і друге повідомлення знайдено.

Застосування даного алгоритму дозволить зменшити витрати на смарт-карти, збільшивши при цьому криптостійкість даних. При цьому слід дотримуватись наступних вимог[6]:

• Криві розглядаються або над простими полями (порядок q яких дорівнює простому числу p), або над полями характеристики два (у яких $q = 2^m$).

• Для представлення елементів поля використовується або стандартний базис, породжений тричленом або п'ятичленом, або гаусівський нормальний базис.

• Крива E задається вибором двох елементів a, b поля GF(q). В випадку $p>2$ вона має вид $Y^2 = X^3 + aX + b$, а в випадку $p = 2$ вид $Y^2 + XY = X^3 + aX + b$ Таким чином, рекомендуються тільки несуперсингулярні криві.

В наступній таблиці приводиться необхідна довжина ключа(у бітах) для забезпечення однакової криптоскійкості, застосовуючи алгоритм RSA та алгоритм на еліптичних кривих [3]:

Табл. 1 – Порівняння довжини ключа RSA і еліптичної криптографії

RSA	*Еліптична криптографія*
512	106
768	132
1024	160
2048	210
21000	600

Порівняння швидкодії алгоритмів при однаковій довжині ключа за матеріалами [3] дозволяє зробити ряд висновків:

Табл.2 – Порівняння RSA і еліптичної криптографії за швидкодією

Характеристики алгоритмів	*RSA*	*Еліптична криптографія*
Довжина відкритого ключа, біт	1024	160
Швидкість генерації ключа, мс	1432	65
Швидкість шифрування, мс	4,28	1,4
Швидкість розшифрування, мс	48,5	26,7

Шифрування на еліптичних кривих значно перевершує використовуваний зараз алгоритм RSA за швидкодією, а також потребує значно менше пам'яті. Отже, застосування алгоритму шифрування на еліптичних кривих може дозволити значно зменшити витрати на смарт-карти, не втративши при цьому криптостійкість даних.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Rankl, W.; W. Effing (1997). Smart Card Handbook. John Wiley & Sons. ISBN 0-471-96720-3. Guthery, Scott B.; Timothy M. Jurgensen (1998). SmartCard Developer's Kit. Macmillan Technical Publishing. ISBN 1-57870-027-2.

2. Брюс Шнаер. Прикладная криптография [Текст] / Шнаер Б. – Новгород: Триумф, 2012. – 784с. – ISBN 978-5-89392-527-2

3. Karu P. Practical comparison of fast public-key cryptosystems / P. Karu, J. Loikkanen. – Helsinki: Helsinki University of Technology, 2001. – 93 с.

4. A. Menezes, P. van Oorschot, S. Vanstone. Handbook of Applied Cryptography. — CRC-Press, 1996. — 816 p. — (Discrete Mathematics and Its Applications

5. Болотов Анатолий Александрович. Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы / Анатолий Александрович Болотов, Сергей Борисович Гашков, Александр Борисовч Фролов. – М.: КомКнига, 2006. – 328 с. - ISBN 5-484-00443-8

6. Мао Венбо. Современная криптография. Теория и практика. — М.: Вильямс, 2005. — 768 с. ISBN 5-8459-0847-7. - ISBN 0-13-066943-1.

Design and simulation of sinusoidal signal digital generator with hardware implementation on FPGA

Stas Danyil Oleksandrovych
student of NTUU "KPI", faculty of informatics and computer techniques, department of computer techniques.
Ukraine, Kyiv.

This paper developed and explored digital generator sinusoidal signal with its hardware implementation on FPGA synthesizer developed structural description language VHDL in FPGA development environment, and then conducted simulations in ModelSim program and results that correspond to the principle of the scheme.

Keywords— FPGA, VHDL, Direct Digital Synthesizers.

Розробка та моделювання Цифрового генератора синусоїдального сигналу при апаратній реалізації на ПЛІС

Стась Даниїл Олександрович
студент НТУУ «КПІ», факультет інформатики та обчислювальної техніки, кафедра обчислювальної техніки.
Україна, Київ.

В даній роботі розроблений та досліджений цифровий генератор синусоїдального сигналу з його апаратною реалізацією на ПЛІС, розроблено структурний опис синтезатора на мові VHDL в середовищі розробки ПЛІС, після чого проведено моделювання в програмі ModelSim і отримані результати, які відповідають принципу роботи схеми

В даний час прикладеться чимало зусиль щоб замінити всі (або майже всі) аналогові компоненти системи зв'язку за допомогою архітектури з використанням цифрового обладнання, функціональні можливості якого можуть бути налаштовані за допомогою ПВМ на спеціалізованому програмному забезпеченні. Так і в системах радіо зв'язку їде перехід з аналогової елементної бази на цифрову [1].

Цифрова реалізація радіо компонентів має чималу кількість важливих переваг над їх аналоговою реалізацією[2]. Найбільш важливими перевагами є:

- істотно краща повторюваність і стабільність (старіння і узгодження компонентів вже не є факторами обмеження продуктивності);
- здійснення функцій обробки сигналу, які неможливі з аналогового пристрою (наприклад, FIR-фільтри);
- апаратні засоби «тонкого налаштування» замінюється налаштуванням програмного забезпечення;
- розробка більш економічних та багатофункціональних радіостанцій, що підтримують різних типів модуляцію сигналу та ширину смуги пропускання;
- великий потенціал зниження собівартості продукції і часу її розробки.

Однією з важливих частин пристроїв передачі радіосигналів, є «генератор з цифровим управлінням» (NCO), який повинен забезпечувати стабільну частоту гармонійного сигналу. Один з найпопулярніших способів вирішити цю проблему сьогодні є використання DDS (Direct Digital Synthesizers) техніки [3], які можуть бути реалізовані в ПЛІС.

Метою роботи є розробка цифрового генератора синусоїдального сигналу на базі ПЛІС.

Перш ніж розглядати структуру DDS, нам необхідно зрозуміти принцип, за яким працює синтезатор.

Він полягає в наступному: прямий цифровий синтезатор - синтезатор частот, в якому цифровими методами безперервно формуються відліки (коди) вихідного сигналу, які потім перетворюються в аналоговий сигнал певної частоти. Функціональний блок, в якому відбуваються ці відліки, називається накопичувачем фази (НФ). Зміни в

акумуляторі фази відбуваються під впливом тактової частоти і коду частоти (код тієї частоти, яка буде генеруватися синтезатором). Головним параметром цього блоку є розрядність. Це характеристика впливає на параметри частотного синтезатора, а саме: в залежності від тактової частоти визначається частотне розширення і діапазон одержуваних частот, а також якість високих частот.

Значення амплітуди сигналу, відповідні поточної фазі формованого сигналу, можуть обчислюватися в синтезаторі або вибиратися з відповідного ЗУ. Тут краще другий метод, так як безпосереднє обчислення фази буде знижувати швидкодію схеми. Таблиця (Look Up Table), з якої вибираються значення sin, найчастіше розміщується в ПЗУ.

Вибір з ПЗУ проводиться так: відліки з накопичувача фази формують адреса для ПЗУ, за яким вибираються осередки з потрібною фазою. Розмір ПЗУ визначається за кількістю значень функції sin в ньому. Чим вище це значення, тим чіткіше буде отриманий сигнал. Однак, в свою чергу, така можливість обмежена можливостями ЦАП.

Підвищити кількість значень в ПЗУ можна також використовуючи симетрію функції sin, а саме половину фази або чверть фази. В роботі використано четверть фази.

Далі значення з осередків пам'яті йдуть на цифро-аналоговий перетворювач (ЦАП), де формується синусоїдальний сигнал, що має вигляд «сходинок».

На виході ЦАП повинен стояти ФНЧ - фільтр нижніх частот (anti-aliasing filter), яких згладжує вихідний сигнал (прибирає «сходинки»).

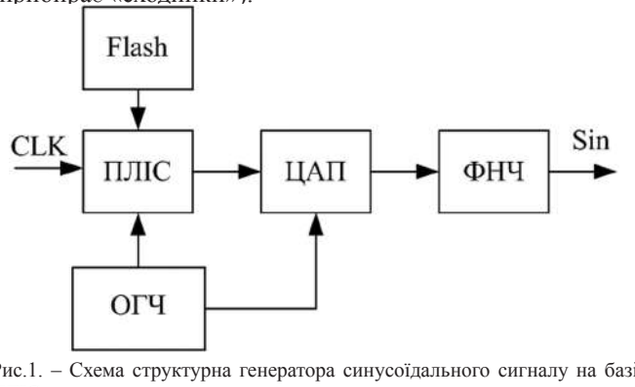


Рис.1. – Схема структурна генератора синусоїдального сигналу на базі ПЛІС

У підсумку ми повинні отримати сигнал потрібної нам частоти (яка вибирається кодом частоти CLK). Структурна схема такого генератора показана на рис. 1.

В ході виконання даної роботи були розглянуті принципи прямого цифрового синтезу, його переваги і недоліки в порівнянні з іншими видами синтезу частоти. Розглянуті схемотехнічні реалізації прямого цифрового синтезатора частоти, оцінені їхні переваги й недоліки, і вибрана відповідна реалізація, яка послужила основою проектованого пристрою. Розроблено структурний опис синтезатора на мові VHDL в середовищі розробки ПЛІС, після чого проведено моделювання в програмі Modelsim і отримані результати, які відповідають принципу роботи схеми. Для наочності, на основі результатів, записаних в файл, за допомогою Excel з пакету Microsoft Office 2007, були отримані графіки і гістограми синусоїди, що генерується.

ВИСНОВОК

Розроблений цифровий генератор синусоїдального сигналу має високу швидкодію та точність, та високу ефективність використанні ресурсу ПЛІС. Може бути використаний як елемент пристроїв передачі радіосигналів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Mikael Olofsson. Introduction to Digital Communication. Department of Electrical Engineering, Linköpings universitet, 2010.

2. Roy Blake. Electronic Communication Systems 2nd edition. Delmar, Thomson learning, 2002.

3. Stanford Telecom. Numeric Modulation in DDS Systems. The DDS Handbook, sixth edition, 1990.

Transmitting multimedia data via channels with low signal to noise ratio

Vadim Poltorak
ACTS NTUU "KPI"
Ukraine, Kyiv

Dmytro Makoivets
ACTS NTUU "KPI"
Ukraine, Kyiv

Summary

Argumentation the possibility of usage cascade codes for transmitting multimedia data via channels with low signal to noise ratio. Cascade series-parallel coding structure.

Keywords

Forward error correction, cascade codes, turbo-codes, signal to noise ratio

Передача мультимедійних даних в каналах з низьким співвідношенням сигнал/завада

Полторак Вадим Петрович
к.т.н., доцент кафедри АУТС НТУУ "КПІ"
Україна, Київ

Макоївець Дмитро Володимирович
студент кафедри АУТС НТУУ "КПІ"
Україна, Київ

Анотація

Аргументація можливості використання каскадних кодів для передачі мультимедійних даних в каналах з низьким співвідношенням сигнал/завада. Послідовно-паралельна каскадна кодова структура.

Більшість сучасних систем завадостійкості забезпечують практично безпомилкову передачу даних, але майже всі вони розраховані на застосування для передавання по високоякісних каналах, які гарантують досить незначний вплив шуму в каналі на сигнал, що передається. Однак досі існують канали з високим рівнем шуму відносно сигналу. Це може бути також звичайний канал зв'язку, але зі збільшеною відстанню передачі фізичним середовищем (довший кабель, ніж прописано в стандарті, або приймач розташований далі від радіо транслятора, ніж було передбачено). При цьому збільшується вплив шуму в каналі на сигнал, що спричинює виникнення помилок. У випадку передачі відео або інших мультимедійних даних такими каналами зазвичай немає можливості на повторну пересилку пошкодженого пакету, бо це суттєво збільшить затримку. До того ж немає гарантії, що шум не спотворить повторно пересланий пакет. Для забезпечення достатньої якості зв'язку затримка при передачі звуку не повинна перевищувати 250-300 мс [1], оскільки відеодані повинні доставлятись синхронно з аудіоданими, то ця межа справедлива і для відеоданих. При цьому для відеозв'язку з роздільною заданістю 720p необхідна швидкість не менше 512 Кбіт/с. Тому очевидно, що у випадку мультимедійних даних зазвичай просто немає часу на повторну пересилку. У такій ситуації є потреба в потужній системі завадостійкості, що дозволяє виправляти помилки відразу на стороні приймача.

При виникненні значної кількості помилок один завадостійкий код може не впоратись з виправленням. До того ж кожен з кодів має свої сильні та слабкі місця.

Застосування каскадного принципу дозволяє досить суттєво збільшити завадостійкість системи. Застосування кількох ступенів кодування дозволяє збільшити мінімальну кодову відстань і, як наслідок, збільшити здатність коду виправляти помилки [2].

Класичними вважаються послідовні каскадні коди. Оптимальним в цьому випадку є використання каскадного кодування на основі двох кодів, що називаються зовнішнім і внутрішнім. Вихідне повідомлення кодується спочатку зовнішнім кодером, а вже потім закодоване повідомлення подається на вхід внутрішнього кодера і лише після такого подвійного кодування передається в канал зв'язку [3]. Використання більше ніж двох ступенів при каскадному кодуванні звісно дозволяє ще більше покращити завадостійкість, проте досить суттєво зростає і надлишковість. Збільшення надлишковості призводить до збільшення обсягів даних, що передаються, тобто зростання трафіку.

Однак останнім часом набуває поширення застосування турбо-кодів. По своїй суті турбо-код є паралельним каскадним блоковим кодом. Турбо-код складається з каскаду паралельно з›єднаних систематичних кодів. Ці складові називаються компонентними кодами. В якості компонентних кодів можуть використовуватися згорткові коди, коди Хеммінга, Ріда — Соломона, Боуза — Чоудхурі — Хоквінгема та інші. Залежно від вибору компонентного коду турбо-коди діляться на згорткові турбо-коди та блокові турбо-коди добутку [4].

Згортковий турбо-код знайшов застосування в мережах LTE. В цьому випадку він здійснює завадостійке кодування з кодовою швидкістю 1/3 за допомогою схеми із двох паралельно зв'язаних згорткових кодерів із внутрішнім перемішувачем біт. Ця схема гарантує імовірність виникнення помилки на рівні $9 \cdot 10^{-7}$ при співвідношенні сигнал/завада на рівні 13 дБ. Ці результати справедливі для використання при передачі 16QAM.

Проте при використанні каналу зі значно нижчим значенням співвідношення сигнал/завада така система кодування не буде справлятися і імовірність виникнення помилки значно зросте, що призведе до суттєвого зниження якості передачі. Тому постає проблема потреби підвищення потужності механізму завадостійкості.

І тут можна повернутись до згаданого раніше принципу послідовного каскадного кодування. Тільки тепер одним із послідовних кодів буде турбо-код. Таким чином отимаємо послідовно-паралеьну каскадну структуру. Даний принцип не набув поширення здебільшого через складність процесу кодування/декодування, а саме – витрат часу на нього [5].

Проте обчислювальна потужність комп'ютерів в наш час та потенційне застосування для передачі в сильно зашумлених каналах нівелює цей недолік. Це пояснюється тим, що в таких каналах передача на надвисоких швидкостях неможлива через обмежену пропускну здатність каналу, так як співвідношення сигнал/завада набуває малих значень:

$$C = \Delta f_e \log_2 \left(1 + SNR\right),$$

де SNR – співвідношення сигнал/завада.

Тому обчислювальна техніка цілком впорається з таким потоком даних не вносячи додаткової затримки.

Запропонована послідовно-паралельна схема дозволяє забезпечувати імовірність виникнення помилки на рівні близько $1 \cdot 10^{-8} - 1 \cdot 10^{-7}$ для порівняно невисоких корисних швидкостей (близько 2 Мбіт/с) при співвідношенні сигнал/завада близьких до 0 дБ. Звісно, враховуючи низьке значення співвідношення сигнал/шум в каналі, за таку швидкість доведеться платити досить широкою смугою частот, необхідною для передачі. Проте для певних мереж передачі даних застосування такого підходу було б аргументованим за рахунок можливості суттєвого збільшення відстані, на яку передаватимуться дані. Тому системи з послідовно-паралельною структурою каскадного кодування мають значний потенціал і потребують подальшого детального дослідження для знаходження оптимальної комбінації парламентів складових кодів.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Обеспечение качества IP-телефонии [Електронний ресурс] : [Веб-сайт] – Електронні дані. – [Україна: ТОВ "Яліта", 2009]. – Режим доступа: http://voip.jalita.com/literature/book_1/5.shtml (дата звернення 20.04.2016). - Назва з екрану.

2. Richardson T. Modern Coding Theory / T. Richardson, R. Urbanke. – Cambridge: Cambridge University Press, 2007. – 576 p. – ISBN 978-0-521-85229-6.

3. Жураковський Юрій Павлович. Теорія інформації та кодування: Підручник. / Ю. П. Жураковський, В. П. Полторак. – К.: Вища шк., 2001. – 255 с.: іл. – ISBN 966-642-031-7.

4. Теория кодирования / Т.Касами, Н. Токура, Е. Ивадари, Я. Инагаки. – М.: Мир, 1978. – 576 с. – ISBN 5-8459-0887-2.

5. Золотарев В.В.. Помехоустойчивое кодирование. Методы и алгоритмы: Справочник./ В.В. Золотарев, Г.В. Овечкин. – М.: Горячая линия – Телеком., 2004. – 126 с.: іл. – ISBN 5-93517-169-4.

Improvement of searching and booking tickets in online services of Ukraine

Rudnytskykh Dmytro Olesandrovich
Student NTUU «KPI»
Kyiv, Ukraine

Summary. The article deals about general description of searching and booking tickets to the online services of Ukraine. There have been identified disadvantages and basic solutions, including the use of Dejkstra algorithm and A* algorithm, that make additional assumptions or conduct so-called informative search in the analysis of the main features ticket services online. It was concluded about the lack of a universal algorithm for solving a wide range of tasks, which leads to the need to find the optimal combination of instruments and tools for software and play all reasonable search customer needs.

Keywords: services search and booking tickets, graph theory, algorithms Dijkstra and A * search routes online.

Вдосконалення системи бронювання та пошуку квитків в онлайн сервісах України

Рудницьких Дмитро Олександрович
студент НТУУ «КПІ»
Київ, Україна

Анотація. У тезах наукової доповіді здійснюється загальна характеристика систем бронювання та пошуку квитків в он-лайн-сервісах України. При здійсненні аналізу основних можливостей сервісів продажу квитків он-лайн були виділені недоліки та основні шляхи їх вирішення, серед яких використання алгоритмів Дейкстрі та А*, які роблять додаткові припущення, або проводять так званий інформативний пошук. Був зроблений висновок щодо відсутності універсального алгоритму для вирішення широкого спектру задач, що зумовлює потребу пошуку оптимального поєднання інструментів та засобів для забезпечення та відтворення усіх обгрунтованих пошукових потреб споживача.

Ключові слова: сервіси бронювання та пошуку квитків, теорія графів, алгоритми Дейкстрі та А*, пошук маршрутів онлайн.

У сучасному українському суспільстві такі сервіси, як Яндекс.Карти та Google.Maps вже змінили наше уявлення про побудову маршрутів, як використовуючи власний транспорт, так і громадський. На сьогодні звичним стали системи побудови шляхів в навігаторах та інших пристроях та придбання за обраними маршрутами квитків он-лайн. Водночас окремі системи для покупки квитків, зокрема офіційний сервіс ПАТ «Укрзалізниці» – booking.uz.gov.ua, недостатньо задовольняють потреби споживача та не відповідають сучасним вимогам, хоча ПАТ «Укразалізниця» є монополістом у сфері надання послуг залізничних перевезень і більша частина споживачів користується саме цим сервісом. Аналізуючи основні можливості сервісу продажу квитків он-лайн на сайті booking.uz.gov.ua, можна виділити такі недоліки:

- відсутність публічного API для отримання інформації про маршрути та наявність квитків;
- неможливість побудови маршрутів, якщо між двома пунктами відсутні прямі сполучення;
- не існує єдиної бази для приміських та національних маршрутів;
- можливість шукати квитки тільки на точну дату;
- відсутність механізму повідомлення користувачів про появу нових квитків на вибрані маршрути;
- відсутність інтеграції з іншими системами сполучення (авіа- та автотранспорт);
- відсутність мобільних додатків.

Логічно, що в Україні з'явилося багато систем, які намагаються вирішити одну чи декілька з вищеперерахованих проблем. Такі сервіси, наприклад, *bilet.privatbank.ua* та *e-kvytok.kiev.ua* створюють для зручності користування мобільні додатки та інтегруються з іншими сервісами. Сервіс *tickects.ua* додатково надсилає споживачу послуги нагадування про появу нових квитків за допомоги e-mail та sms та надає більш гнучкий пошук.

Проблему з відсутності відкритого API та відсутності інтеграції з іншими системами сполучення вирішила ТОВ «Argest group» (http://www.argest.com.ua/) з сервісом Uticket-api, які дають інформацію про наявні квитки, однак не дають додаткової інформації про усі можливі маршрути, якщо квитків не має в наявності.

На рівні з вищевикладеним залишаються дві проблеми, які до сьогодні не вирішені на рівні жодного з он-лайн-сервісів продажу квитків – відсутність єдиної системи для всіх видів залізничного транспорту та неможливість побудови складних маршрутів. Сьогодні інформацію про приміські потяги наявна на одному з шести підрозділів сайту ПАТ «Укрзалізниці». Цю проблему дуже складно вирішити за допомоги зовнішніх розробників, але розробити систему для розрахунку маршрутів з пересадками цілком можливо при наявності відкритого API. Подібну задачу вирішує будь-який сервіс пошуку авіаквитків та багато залізничних кас Європи та США. Водночас така система пошуку залізничних маршрутів враховуючи можливі пересадки, все одно має певні обмеження. Так, користувачу необхідно надати декілька варіантів маршруту; треба враховувати, наявність квитків у продажу за обраними маршрутами, оскільки користувач можливо захоче отримувати повідомлення, коли з'являться додаткові місця; пересадки між рейсами повинні враховувати час відправлення від кожної станції та деяку дельту для зміни транспорту; кількість пересадок не може бути більше деякого заданого числа, наприклад п'яти, оскільки при більшому значенні це може дуже ускладнити пересування пасажира; між двома станціями може бути більше, ніж один маршрут; якщо використовувати різні метрики в якості ваги ребер, наприклад ціна, то додатково необхідно враховувати різні типи місць в рамках одного вагону; необхідно підтримувати актуальність даних кожного дня, оскільки багато маршрутів мають дуже специфічний розклад руху та сезонність, які не можливо наперед передбачити.

Побудова системи, яка б враховувала вищевказані обмеження, можлива за допомогою вирішення класичної задачі з теорії графів – а саме задачі пошуку найкоротшого шляху. Оскільки реально допустимий маршрут в результаті не може складатися з великої кількості пересадок, тому не має сенсу розглядати дуже складні алгоритми. Відповідно варто надати перевагу алгоритмам Дейкстрі та **A***, які роблять додаткові припущення, або проводять так званий інформативний пошук.[1]

Класичний алгоритм Дейкстри працює тільки для графів без циклів від'ємної довжини. У процесі роботи алгоритму послідовно позначаються розглянуті вершини графа. Причому вершина, позначена останньої (на даний момент) розташована ближче до вихідної вершині, ніж всі непозначених, але далі, ніж всі помічені. Спочатку позначається вихідна вершина; наступної, очевидно, буде позначена вершина, найближча до початкової, і суміжна з нею. Нехай на якомусь кроці вже позначений кілька вершин. Відомі найкоротші шляхи, що ведуть з вихідної вершини до поміченим. Для кожної з непозначених вершин проробимо наступне: 1. Розглянемо всі дуги, провідні з помічених вершин в одну непозначених. Кожна така дуга є останньою дугою на шляху з вихідної вершини в цю непозначених. 2. Виберемо з цих шляхів найкоротший. А потім виберемо серед них самий короткий до всіх непозначених вершин, і позначимо вершину, до якої він веде. Алгоритм завершиться, коли будуть помічені всі досяжні вершини. У результаті роботи алгоритму Дейкстри будується Дерево найкоротших шляхів.[2]

Алгоритм **A*** належить до евристичних алгоритмів пошуку. Алгоритм використовує допоміжну функцію (евристику), аби скеровувати напрям пошуку та скорочувати його тривалість. Алгоритм повний в тому сенсі, що завжди знаходить оптимальний розв'язок, якщо він існує. Алгоритм **A*** спершу відвідує ті вершини, які ймовірно ведуть до найкоротшого шляху до мети. Аби розпізнати такі вершини, кожній відомій вершині x зіставляється значення $f(x)$, яке дорівнює довжині найкоротшого шляху від початкової вершини до кінцевої, який пролягає через обрану вершину. Вершини з найменшим значенням f обираються в першу чергу. Функція $f(x)$ для вершини x визначається так: $f(x)=g(x)+h(x)$, де: $g(x)$ функція, значення якої дорівнюють вартості шляху від початкової вершини до x, $h(x)$ – евристична функція, оцінює вартість шляху від вершини до кінцевої [3]. Використана евристика не

повинна давати завищену оцінку вартості шляху. Прикладом оцінки може служити пряма лінія: загальний шлях не може бути коротшим за пряму лінію. Головною різницею цих алгоритмів є наявність евристичної функції, яка дає виграш у швидкості, бо алгоритмічна складність Дейкстрі та $A^* O(|E|+|V|*\log|V|)$ та $O(|V|*\log|V|)$ відповідно.

Виникає додаткова задача визначення цієї самої евристичної функції і якщо для пошуку за критерієм відстані евристикою може слугувати пряма відстань до кінцевої точки, то для інших метрик визначити евристичну функцію може бути складно, або навіть не можливо. В останньому випадку однозначно треба використовувати алгоритм Дейкстрі.

Варто погодитись з тим, що класичні алгоритми допомагають у вирішенні окремих завдань, однак безсилі перед іншими (необхідність знаходити декілька розв'язків, враховувати більше, ніж одну дугу між вершинами та нелінійність маршруту з точки зору різного часу відправлення). Такий факт пояснюється відсутністю універсального алгоритму, який би вирішував усі необхідні задачі. Отже, нині вкрай актуальним завданням як для теоретичного дослідження, так і практичного застосування є пошук оптимального поєднання інструментів та засобів для забезпечення та відтворення усіх обґрунтованих пошукових потреб споживача, що в кінцевому підсумку значно покращить якість систем бронювання та пошуку квитків он-лайн.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pathfinding[Електронне видання]. – Електр. дан. – Режим доступу:

2. http://en.wikipedia.org/wiki/Pathfinding Молчановський О. Курс з дискретної математики. / О.Молчановський // [Електронний ресурс]. – Режим доступу:

http://oim.asu.kpi.ua/files/DM/30_Shortest_path_algorithms.pdf

3. Жигаревич О.К. Методи та засоби проектування та розробки системи оптимізації транспортних маршрутів /О.К.Жигаревич // Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». – Луцьк, 2013. – Випуск №11.

Рецензент: к.т.н. доц. каф. ТК НТУУ «КПІ» Т.А. Ліхоузова

Influence of QoS characteristics on the quality of voice traffic over packet networks

Kirill Korchagin
Luxoft
Ukraine, Kiev

Vadym Poltorak
NTUU KPI
Ukraine, Kiev

Annotation

This article analyzes the main methods for evaluating the quality of voice traffic in the network, the quality of the evaluation criteria of network traffic according to QoS and E-model. Comparison of subjective and objective methods of assessing the quality of the voice traffic. The influence of the technical characteristics of the network on the subjective assessment of the quality of voice traffic in the network. Based on studies conducted by the authors have shown that the QoS parameters have a significant influence on the quality of the voice traffic in the network

Key words: QoS, MOS, E-model, R-factor, Quality, network.

Влияние характеристик QoS на качество речевого трафика в пакетных сетях

Корчагин К.П.
Luxoft
Украина, Киев

Полторак В.П.
НТУУ КПИ
Украина, Киев

В данной статье проанализированы основные методы оценки качества речевого трафика в сети, критерии оценки качества сетевого трафика согласно QoS и Е-модели. Произведено сравнение субъективной и объективной методик оценки качества речевого трафика. Проанализировано влияние технических характеристик сети на субъективную оценку качества речевого трафика в сети. На основе проведенного исследования авторами было показано, что параметры QoS оказывают значительное влияние на качество речевого трафика в сети.

Ключевые слова: QoS, MOS, Е-модель, R-factor, Качество, сеть.

Наиболее распространенный и точный критерий оценки качества речи в VoIP сетях – это восприятие услуги пользователем. Наиболее широко используемой методикой является методика субъективной оценки качества которая известна как MOS (Mean Opinion Score). В соответствии с данной методикой качество предоставляемого сервиса рассчитывается как среднее арифметическое от всех оценок, выставляемых экспертами после прослушивания тестового звонка. Экспертные оценки составляются в соответствии со следующей шкалой: 5 — отлично, 4 — хорошо, 3 — приемлемо, 2 — плохо, 1 — неприемлемо. Оценка 3.5 и выше характеризирует хорошее качество речи.

Помимо субъективной оценки качества также существуют объективные методики оценки качества сети, которые ориентируется на технические характеристики. Одной из таких методик является Е-модель. Данная методика описана в рекомендации ITU-T Rec. G.107 и называется Е-модель. «Данная модель основана на математическом алгоритме, с помощью которого отдельные параметры передачи преобразуются в различные отдельные «факторы ухудшения состояния», которые считаются аддитивными на психологическом уровне. Алгоритм Е-модели также учитывает комбинированные эффекты нарушений в связи, которые происходят одновременно, а также некоторые эффекты маскирования. Е-модель основана на моделировании результатов большого количества субъективных тестов, проведенных в прошлом, в широком диапазоне параметров передачи. Основным результатом расчетов Е-модели является скалярная величина оценки качества известная как «Transmission Rating Factor, R-factor, R». Представляется Е-модель в следующем виде:

$$R = R_0 - L_s - L_e - L_d + A,$$

где:

R_0 – максимальное значение r-factor которое возможно достичь при использовании выбранного кодека.

L_s – искажения, вносимые кодеками и шумами в канале;

L_e – искажения, вносимые оборудованием, включая и потери пакетов;

L_d – искажения за счет суммарной сквозной задержки («из конца в конец») в сети;

А – фактор пеимущества который предназначается для корректирования значения r-factor`a.»[1]

Согласно рекомендации ITU-T Rec. G.107, значения r-factor`a и MOS можно свести в таблицу

Значение R-фактора	*Категория качества и оценка пользователя*	*Значение оценки MOS*
90 < R < 100	Наилучшая	4,34-4,5
80 < R < 90	Высокая	4,03-4,34
70 < R < 80	Средняя	3,60-4,03
60 < R < 70	Низкая	3,10-3,60
50 < R < 60	Неприемлимая	2,58-3,10

Е-модель предоставляет критерии качества сети и опирается на технические составляющие, которые включают в себя и параметры задержки и параметры голосовых кодеков. Помимо Е-модели существуют и другие критерии оценки качества трафика в сети, такие как QOS.

QoS представляет собой набор критериев по которым можно оценить качество сети и качество передаваемого речевого трафика. «Согласно QoS все сети можно разделить на три класса качества:

I класс сети характеризуется следующими техническими параметрами – потеря пакетов не должна превышать 0.5% от общего количества пакетов сети и вариативная задержка не должна превышать 10мс. Сети первого класса качества предоставляют наилучшее качество речи в сети.

II класс - потеря пакетов не превышает 1% от общего количества пакетов в сети и вариативная задержка не превышает 20мс. Сети второго класса качества предоставляют высокое качество речи в сети.

III класс - потеря пакетов не превышает 2% от общего количества пакетов в сети и вариативная задержка не превышает 40мс. Сети третьего класса качества предоставляют среднее качество речи в сети.» [2]

Помимо критериев качества QoS так же включает в себя набор сервисов которые, при должной настройке, приведут к улучшению качества голосового трафика в сети.

Сопоставим критерии качества QoS и оценки MOS Оценим взаимозависимость критериев качества QoS и оценки MOS. Для этого обратимся к значению r-factor›a и его корреляции с оценками MOS.

Для этого преобразуем формулу расчета R-factor`a к следующему виду:

$$R = R_0 - L_s - L_e - L_d + A = R_0 - 2L_s - D_P - \Delta R_i + \Delta S_i - J_{i-1} - \frac{|\Delta R_{i-1} - \Delta S_{i-1}| - J_{i-1}}{16} - L_e + A, \quad (2)$$

Где:

$L_D = D_{ïê} + D_{ïï} + D_{ää} + D$ – сумма всех задержек в сети

$D_{ïï} = D_i = (R_i - R_{i-1}) - (S_i - S_{i-1}) = \Delta R_i - \Delta S_i$ – сквозная задержка (R – время прибытия пакета в метках времени RTP, S – временная метка RTP, взятая из пакета.)

$D_{ïê} \sim L_s$ – задержки вносимые джитером

D_P – задержка распространения

$D_{ää} = J_i = J_{i-1} + \frac{|D_{i-1}| - J_{i-1}}{16}$ – «задержка в джитер буфере» [3]

Анализ полученного выражения показывает, что подавляющее количество параметров влияющих на значение r-factor`a являются параметрами задержки. При применении и соответствующей настройке сервисов QoS можно улучшить значения параметров задержки и увеличить значение r-factor`a, что приведет к увеличению значения R. Одним из основных принципов QoS является приоритизация трафика и выделение отдельной полосы под каждый вид трафика, что приводит к уменьшению количества потерянных пакетов и уменьшению вариативной задержки в сети.

Основываясь на этих возможностях можем сделать вывод, что применение сервисов QoS позволит значительно улучшить параметр r-factor`a и, как следствие, улучшить значение MOS в сети за счет уменьшение вариативной задержки и количества потерянных пакетов.

ЛИТЕРАТУРА:

1. «E-model» / [Електронный ресурс]. – Режим доступа: https://www.itu.int/ITU-T/studygroups/com12/emodelv1/tut.htm

2. «Оценка качества VoIP» / [Електронный ресурс]. – Режим доступа: http://www.ixc.ua/110

3. «Облегченный протокол пользовательских дейтаграмм (UDP-Lite)» / [Електронный ресурс]. – Режим доступа: http://rfc.com.ru/rfc3828.htm

Information security providing in case of controlling mobile object

Danchul Volodymyr
ACTS NTUU "KPI"
Ukraine, Kyiv

Vadym Poltorak
ACTS NTUU "KPI"
Ukraine, Kyiv

Summary

A problem of controlling mobile objects via the insecure channels was examined. A way to improve security while using short keys was suggested. A protocol with authentication combined with usage of commutative encryption was developed.

Keywords

Mobile object, authentication, information security, commutative encryption

Забезпечення інфозахисту команд управління пересувним об›єктом

Данчул Володимир Сергійович
студент кафедри АУТС НТУУ "КПІ"
Україна, Київ

Полторак Вадим Петрович
к.т.н., доцент кафедри АУТС НТУУ "КПІ"
Україна, Київ

Анотація

Розглянуто проблему управління пересувними об'єктами через незахищений канал. Запропоновано спосіб підвищення рівня захисту шляхом забезпечення аутентифікації при використання у протоколі на основі комутативного шифрування для убезпечення передачі малих за розміром команд управління пересувним об'єктом.

Використовувані в системах захисту інформації криптографічні схеми симетричного шифрування забезпечують гарантовану стійкість шифрування при застосуванні ключів достатнього розміру, наприклад 256 біт [1]. Але часто виникає необхідність швидкої і надійної передачі інформації при наявності у відправника і отримувача лише ключів малого розміру (32-56 біт). Використання ключів такого розміру безпосередньо в симетричному шифруванні дозволяє зловмиснику визначити ці ключі методом повного перебору за прийнятний період часу. В цьому випадку виникає необхідність забезпечення певного рівня стійкості для таких систем, наприклад, операцій.

Особливо гостро ця проблема постає при керуванні рухомими пристроями. Оскільки керування проходить в режимі реального часу, вимагається прийнятна швидкість передачі в умовах жорсткого обмеження потужності передавачів, що встановлюються на рухомий пристрій, а вимоги до захисту інформації залишаються. Команди управління, зазвичай, не великого розміру і можуть бути швидко зашифровані ключами малого розміру.

Одним зі способів вирішення задачі є використання додаткового алгоритму комутативного шифрування. Алгоритм є комутативним, якщо результат послідовного шифрування повідомлення М на ключах К1 і К2 не залежить від порядку використовуваних ключів $E_{K2}\left(E_{K1}(M)\right)=E_{K1}\left(E_{K2}(M)\right)$ де $E_K(M)$ – результат шифрування повідомлення М на ключі К, $D_K(M)$ – результат дешифрування повідомлення М на ключі К [1]. Недоліком цього підходу є неможливість забезпечення аутентифікації повідомлення. Для забезпечення аутентифікації можна застосувати додатковий спільний секретний ключ невеликого розміру, завдяки якому зловмисник не зможе видавати себе за відправника повідомлення. Цей ключ передається або через захищений канал, або з використанням асиметричного шифрування і використовується протягом сеансу зв'язку.

В якості процедури комутативного шифрування пропонується застосувати трьох кроковий протокол Шаміра [3]. Це дозволить позбутися процедури розподілу ключів. Основою подібних протоколів є алгоритм комутативного шифрування, у якому відправник і отримувач мають власні секретні ключі К1 і К2 відповідно.

При застосуванні подібного протоколу передача повідомлення М по відкритому каналу відбувається за наступ-

ною процедурою [3]:

1. Відправник шифрує М своїм ключем К1 і відправляє криптограму C_1 отримувачу:

$$C_1 = E_{K1}(M)$$

2. Отримувач шифрує криптограму C_1 своїм ключем К2 і відправляє назад криптограму C_2:

$$C_2 = E_{K2}(C_1) = E_{K2}(E_{K1}(M))$$

3. Відправник дешифрує криптограму C_2 за своїм ключем і відправляє криптограму C_3 отримувачу :

$$C_3 = D_{K1}(C_2) = D_{K1}(E_{K2}(E_{K1}(M))) = D_{K1}(E_{K1}(E_{K2}(M))) = E_{K2}(M)$$

Отримувач використовує процедуру дешифрування за своїм ключем $D_{K2}(E_{K2}(M))$ і отримує вихідне повідомлення М.

Великою перевагою є те, що ключі К1 і К2 можуть обиратися довільно і для кожного нового повідомлення можливий вибір нових пар ключів. У процесі передачі повідомлення не відбувається обміну ключами і цей протокол може називатись безключовим шифруванням.

Основним недоліком є вразливість подібних протоколів до атаки «людина по середині», коли зловмисник видає себе за одного з учасників обміну.

Шифрування повідомлень спільним ключем малого розміру не є безпечним, оскільки після перехоплення криптограми є можливим повний перебір простору ключів. Для забезпечення належного рівня надійності доцільним є використання безключового шифрування у парі з аутентифікацію повідомлень по спільному ключу малого розміру. Таке застосування спільного ключа принципово відрізняється від його застосування у схемах симетричного шифрування, оскільки у зловмисника буде лише одна спроба вгадати ключ і передати власне хибне повідомлення, в той час як при шифруванні існує можливість багатократного підбору значень ключа. При використанні для аутентифікації ключа довжиною всього лише 32 біти вірогідність обману складає усього лише 2^{-32}, що є прийнятною вірогідністю. Для практичного застосування методу необхідно забезпечити поєднання процедур аутентифікації і шифрування у єдиному протоколі.

Задача аутентифікації. Значення криптограм C_1, C_2, C_3 отримувані в ході виконання протоколу обчислювально не відрізняються від випадкових значень. Шифрування криптограм за спільним коротким ключем з використанням симетричного алгоритму $G_K(C)$, де G_K – алгоритм симетричного шифрування за ключем К, не дозволяє потенційному зловмиснику знайти значення короткого спільного ключа за прийнятний період часу, а легальному отримувачу повідомлення дозволяє вчасно ідентифікувати відправника. При використанні цього підходу пропонована процедура обміну повідомленнями набуває наступного вигляду :

1. Відправник шифрує повідомлення своїм секретним ключем К1: $C_1 = E_{K1}(M)$, отримана криптограма C_1 шифрується спільним ключем К: $S_1 = G_K(C_1)$, криптограма S_1 надсилається отримувачу.

2. Отримувач розшифровує криптограму S_1 використовуючи спільний ключ К: $C_1 = G_K^-(S_1)$, отримана криптограма C_1 шифрується секретним ключем отримувача К2: $C_2 = E_{K2}(C_1)$, після чого криптограма C_2 шифрується спільним ключем К: $S_2 = G_K(C_2)$, і S_2 відправляється відправнику.

3. Відправник розшифровує шифрограму S_2 використовуючи спільний ключ К: $C_2 = G_K^-(S_2)$. Після цього до C_2 застосовується процедура дешифрування за секретним ключем К1: $C_3 = D_{K1}(C_2) = E_{K2}(M)$ і результат пересилається отримувачу. Отримувач розшифровує повідомлення М: $M = D_{K2}(E_{K2}(M))$. Використання в перших двох кроках протоколу додаткового симетричного шифрування за спільним ключем виконує взаємну аутентифікацію повідомлення відправником і отримувачем.

В якості функції $E_K(M)$, що забезпечує властивість комутативності можна використовувати алгоритм шифрування Поліга-Хеллмана [2], що базується на обчислювальній складності задачі дискретного логарифмування за модулем простого числа.

Результатом аутентифікації є відносне убезпечення від атак типу «людина посередині». Значення криптограм, що отримуються в процесі роботи протоколу обчислювально не відрізняються від випадкових значень, їх шифрування на спільному ключі виключає можливість визначення короткого ключа методом повного перебору усіх можливих комбінацій спільного ключа. Це позбавляє зловмисника ефективного засобу відбракування невірних значень ключа методом тестування розшифрованих ним значень на читабельність.

В статті розглянуті можливості побудови протоколів шифрування з спільним секретним ключем малого розміру. В основі ідеї лежить використання безключового шифрування в комбінації з аутентифікацією повідомлень. В якості механізму аутентифікації використовується алгоритм симетричного шифрування частини власних криптограм, що використовуються в процесі комутативного шифрування, на короткому спільному ключі. Запропоновано конкретне протокольне рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Брюс Шнаер. Прикладная криптография [Текст] / Шнаер Б. – Новгород: Триумф, 2012. – 784с. – ISBN 978-5-89392-527-2

2. Patent 4424414, US. - Exponentiation cryptographic apparatus and Method / Martin E. Hellman, Stephen C. Pohlig.

3. Молдовян Н.А. Введение в криптосистемы с открытым ключом [Текст] / Молдовян Н.А. – СПб : БХВ, 2007. – 286с.

Structural optimization of artificial neural networks

Ferens Dmytro
ACTS NTUU "KPI"
Ukraine, Kyiv

Yaroslaw Dorogyy
ACTS NTUU "KPI"
Ukraine, Kyiv

Summary

A problem of increasing the efficiency of neural networks has been examined. The original algorithm of structural optimization based on genetic algorithm and immutable data structures was proposed. A system which can be used to enhance the effectiveness of existing neural network technologies was developed.

Keywords

Neural networks, data science, genetic algorithms, immutable data structures

Структурна оптимізація штучних нейронних мереж

Ференс Дмитро Андрійович
студент кафедри АУТС НТУУ «КПІ»
Україна, Київ

Дорогий Ярослав Юрійович
к.т.н., доцент кафедри АУТС НТУУ «КПІ»
Україна, Київ

Анотація

Розглянуто проблему відвищення ефективності нейронних мереж. Запропонований оригінальний алгоритм структурної оптимізації. Розроблена система що може використовуватись для підвищення якості роботи існуючих нейромережевих технологій.

Штучні нейронні мережі — математичні моделі побудовані за принципом функціонування біологічних нейронних мереж [1]. На сьогоднішній день апарат нейронних мереж широко використовується для вирішення задач класифікації, кластеризації та регресії. Проте, найбільш значимим недоліком нейронних мереж є складність вибору оптимальної структури мережі – кількості шарів, нейронів та зв'язків між ними. Проблема вибору структури тісно зв'язана з проблемами недонавчання та перенавчання. Занадто прості мережі не здатні виявляти закономірності у реальних задачах. Занадто складні мережі мають багато надлишкових параметрів що в процесі навчання налаштовуються не лише на виявлення закономірностей але і на відтворення шуму.

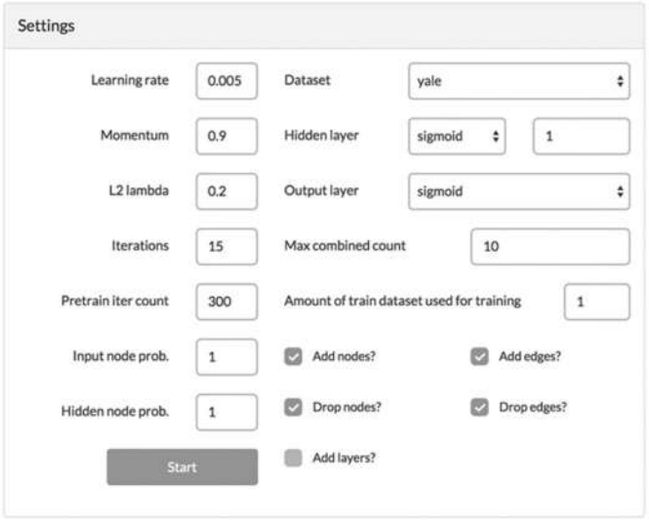
На сьогоднішній день, підбір архітектури нейронної мережі, кількості шарів та нейронів, топології, швидкості навчання та інших гіперпараметрів мережі виконується шляхом проб та помилок. Взаємозв'язок між гіперпараметрами та здатністю до узагальнення невідомий. Саме тому, генетичні алгоритми є ідеальними кандидатами для вирішення таких задач [2].

Запропонований алгоритм структурного навчання використовується на багатошарових мережах прямого поширення та має ітеративний характер: на кожній ітерації виконується пошук структури мережі кращої за попередню. Пошук мережі виконується шляхом перебору усіх можливих мутацій мережі, вибору та комбінації кращих (селекція та схрещення).

Програмна реалізація системи має вигляд клієнт-серверного додатку з асинхронною комунікацією шляхом протоколу WebSocket. Для серверної частини була використана мова програмування Clojure [3] – функціональна мова загального призначення. Для реалізації мережі використовуються незмінні версії двохмірних матриць – при спробі змінити об'єкт такої матриці користувач отримує новий об'єкт, повністю незалежний від оригінального. Таким чином, різні мутації над однією мережею можна обчислювати паралельно і незалежно одна від одної, що значно підвищує ефективність системи.

Для клієнтської частини була використана мова програмування ClojureScript - підмножина мови програмування Clojure що виконується в середовищі веб-переглядача. Це дає змогу значну частину програмного коду використовувати як на клієнтській так і на серверній частинах, що значно спрощує програмування.

На початку роботи, користувачу необхідно задати параметри навчання, параметри генетичного алгоритму та конфігурацію початкової мережі. Оскільки алгоритм орієнтований на оптимізацію існуючих мереж, рекомендується заздалегідь підібрати початкову кількість навчальних епох, протягом яких мережа навчається у звичайному режимі. Також, необхідно вибрати набір даних на яких відбуватиметься навчання.



Панель налаштування алгоритму

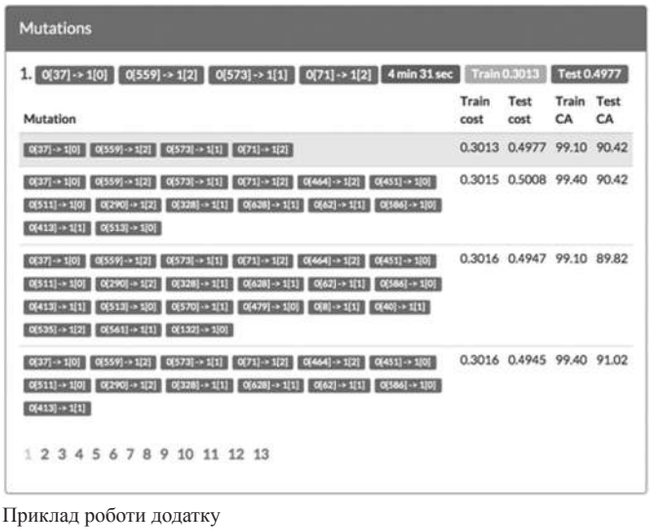
Для навчання мережі використовується алгоритм зворотнього поширення помилки що включає такі методи регуляризації як затухання ваг та інерційність градієнтного спуску. Корекції ваг та зміщення обчислюються:

$$\Delta w_{i,j}[t+1] = -\eta y_i \delta_i + \mu \Delta w_{i,j}[t] - \frac{\eta \varepsilon}{N} w_{i,j}[t]$$

$$\Delta b_i = -\eta \delta_j$$

Окрім того, під час навчання використовуються методи DropOut [4] та DropConnect [5]. Інтерфейс користувача надає змогу задати вірогідність активації кожного нейрону у методі DropOut та вірогідність активації (присутності) у мережі кожного зв'язку у методі DropConnect.

Оскільки кількість ітерацій генетичного алгоритму може бути необмеженою а кількість можливих мутацій збільшується експоненційно по відношенню до розміру мережі, час роботи алгоритму може бути значним. У зв'язку з цим, доцільно використовувати асинхронну модель шляхом протоколу WebSocket – сервер після обчислення кожної мутації та кожної ітерації відправляє результати асинхронно на клієнт що динамічно їх відображає.



Приклад роботи додатку

На рисунку зображений інтерфейс програми під час роботи алгоритму: послідовність вибраних мутацій кожної ітерації, значення ціни на навчальній та тестовій вибірках, витрачений час на обрахунок. Для кожної ітерації алгоритму можливо переглянути усі перевірені мутації у порядку зростання значення ціни. Також, графічний інтерфейс надає змогу порівняти мережу одержану за допомогою структурної оптимізації та після традиційного навчання еквівалентної кількості епох.

Було проведено моделювання алгоритму та дослідження його роботи для вирішення різних задач класифікації та розпізнавання. При вирішенні завдання MONK's-3 [6] алгоритму вдалось зменшити тривалість навчання мережі більш ніж удвічі у порівнянні із традиційним алгоритмом. У задачі TwoSpirals [7] алгоритм зумів побудувати складну багатошарову мережу що виконує класифікацію з точністю 92.7% При класифікації обличч [8] алгоритм зменшив швидкість навчання та збільшив точність класифікації з 93.4% до 95.8% , видаливши при цьому близькько 10% зв'язків початкової мережі.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Хайкин С. Нейронные сети, полный курс [Текст] / Саймон Хайкин. – 2-е изд., перед. – М. : Вильямс, 2008. - 1103 с. – ISBN 5-8459-0890-6

2. Dasgupta D. Designing Application-Specific Neural Networks using the Structured Genetic Algorithm [Текст] / Dasgupta D., McGregor D.: Proceedings of International Workshop on Combinations of Genetic Algorithms and Neural Networks. – 1992.

4. Clojure – home [Електронний ресурс] : [Інтернет-портал]. – [USA, 2008-2014]. – Режим доступу: www.clojure.org (дата звернення 2.05.2015 р.). – Назва з екрана.

5. Sristava N. Dropout: A Simple Way to Prevent Neural Networks from overfitting [Текст] / Sristava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. // Journal of Machine Learning Research. – 2014. –Vol. 15.

6. Wan L. Regularization of Neural Network using DropConnect [Текст] / Wan L., Zeiler M., Zhang S., LeCun Y., Fergus R. : International Conference on Machine Learning. – 2013.

7. Thrun S. The monk's problems: A performance comparison of di erent learning algorithms [Текст] : Technical Report CMU-CS-91-197. – Carnegie Mellon University. – 1991.

8. Lang K. Learning to Tell Two Spirals Apart [Текст] / Lang K., Witbrock M.: Proceedings of the Connectionist Models Summer School. – 1988.

9. The Yale Face database B [Електронний ресурс] : Електронні дані. — [USA, 2001-2015]. – Режим доступу: www.vision.ucsd.edu/~iskwak/ExtYaleDatabase/Yale%20Face%20Database.htm (дата звернення 1.05.2015 р.). – Назва з екрана.

УДК 377.131.11

Information system support of educational process

Ivan V. Chepovyi
Bachelor in Systems Engineering, student
National Technical University of Ukraine 'Kyiv Polytechnic Institute, Kyiv, Ukraine
ichepovoy@gmail.com

Darina V. Pyshnyak
student
National Technical University of Ukraine 'Kyiv Polytechnic Institute, Kyiv, Ukraine
pyshnyak.darina.@mail.ru

Leonid Yu. Yurchuk
associate professor, PhD (technical sciences), associate professor
National Technical University of Ukraine 'Kyiv Polytechnic Institute',Ukraine
leonidyu0@gmail.com

Abstract The basic requirements are determined and the structure of the educational process infotainment system is proposed.

Keywords: informational system; Individual educational route; educational process

Система інформаційної підтримки освітнього процесу

Чеповой Іван Володимирович
бакалавр системної інженерії, студент
НТУУ "КПІ", м. Київ, Україна
ichepovoy@gmail.com

Пишняк Дарина Володимирівна
студент
НТУУ "КПІ", м. Київ, Україна
pyshnyak.darina.@mail.ru

Юрчук Леонід Юрійович
доцент, к.т.н., доцент
НТУУ "КПІ", м. Київ, Україна
leonidyu0@gmail.com

Анотація. Визначені основні вимоги та запропонована структура системи інформаційної підтримки освітнього процесу

Ключові слова: інформаційно-довідкова система, індивідуальний освітній маршрут, освітній процес

Однією з найважливіших проблем у сучасному суспільстві є проблема працевлаштування. Науково-технічний прогрес поряд з позитивними наслідками має і негативні: підвищення вимог до професійної підготовки у сферах класичного виробництва, виникнення багатьох нових професій, швидке старіння отриманих знань – в деяких випадках нові професії зникають швидше, ніж встигає розгорнутися підготовка відповідних фахівців та інше. Швидка зміна виробництва призводить до зміни до реорганізації підприємств, що призводить до плинності кадрів – практично не вдається працювати на одному місці все життя. Дуже часто виробництво створюється у вигляді «тимчасового трудового колективу» - залучення фахівців для реалізації конкретного проекту. Наслідками цих обставин для працівника стає необхідність займатися професійним розвитком на протязі всього життя, отримувати кілька професій, шукати робочі місця.

Інтелектуалізація виробничої діяльності вимагає отримання більш високого рівня професійної підготовки. У багатьох країнах протягом 10-15 років у 4-5 разів розширилась підготовка фахівців з вищою освітою [1]. Наявність вищої освіти дозволяє легше знайти робоче місце, отримувати більшу заробітну плату, займати більш високі посади у порівнянні з особами, що її не мають. Відповідна тенденція проглядається і в Україні. Так у [2] визначено «за даними національного вибіркового Обстеження з питань економічної активності населення у 2012 р. проаналізовано взаємозв'язок між рівнем освіти та статусом на ринку праці. Аналіз взаємозв'язку дозволяє зробити висновок про наявність прямої залежності між зростанням рівня освіти та рівнем зайнятості, рівнем оплати праці».

У зв'язку з цим виникає необхідність дослідження засобів та методів організації навчання особи для отримання нею нових форм мислення, поведінки і співпраці, відповідальності за власні дії.

Одним з аспектів сучасної парадигми освіти є її індивідуалізація з урахуванням можливостей та вподобань особи, інтересів роботодавців та суспільства в цілому.

Основою реалізації такого підходу може бути концепція створення індивідуальних освітніх маршрутів-трєкторій. Деякі фахівці розглядають ці терміни як синоніми, автори відповідно до [3] вважають, що індивідуальний освітній маршрут (ІОМ) становить змістовий компонент, а також розроблений спосіб його реалізації на основі індивідуальної освітньої траєкторії.

На сьогодні в теорії та практиці національної освіти накопичені значні наукові напрацювання, які можуть слугувати основою для створення індивідуальних освітніх маршрутів. Але у своїй більшості вони пов'язані з суто педагогічними питаннями і практично ніде не торкаються питань організації інформаційного забезпечення формування ІОМ на різних рівнях освіти та з урахуванням інтересів учнів.

У той же час у ЄС з прийняттям у 1999 році Болонської декларації провадяться багато різних організаційних програм, що забезпечують можливість зацікавленим особам отримати інформацію про навчальні заклади, необхідні компетенції для отримання певних кваліфікацій та інш.

Інструментальною підтримкою цих програм виступають інформаційно-довідкові системи. Огляд функціональних можливостей деяких з цих систем наведено у [4].

На теренах України таких систем практично не існує. В той же час необхідність їх розробки стає все більш актуальною з подальшим розвитком Болонського процесу, до якого Україна підключилась у 2005 р. [5].

Для розробки інформаційно-довідкових систем (ІДС) (як і для інших) необхідно визначити цільову аудиторію, її інтереси, визначити концепцію побудови та функціональність самої системи.

ЦІЛЬОВА АУДИТОРІЯ ТА ЇЇ ІНТЕРЕСИ

Оскільки освітній процес охоплює практично всі верстви населення і сьогодні продовжується все життя, то можна виділити наступні основні групи зацікавлених користувачів (не торкаючись категорії керівників освітою).

- Школярі та їх батьки при вивченні потреб ринку праці з метою вибору майбутньої професії.
- Вступники до навчальних закладів з метою вибору закладу.
- Студенти для формування та корекції індивідуального освітнього маршруту (ІОМ).
- Особи, що мають бажання отримати додаткову освіту.
- Особи, що бажають підвищити кваліфікацію.
- Навчальні заклади з метою отримання учнів (студентів).
- Праценадавачі з метою реалізації їх вимог до підготовки робітників.

КОНЦЕПЦІЯ ПОБУДОВИ

Концептуальною основою побудови ІДС, враховуючи інтереси цільової аудиторії, може бути система інформаційного забезпечення, що дозволяє формувати та в подальшому підтримувати індивідуальні освітні маршрути (СІП ІОМ). В залежності від наповнення баз даних та комплектації вона може забезпечувати як інтереси окремої особи, так і більш складні структури – навчальні заклади і національну освіту взагалі.

Враховуючи досвід інформаційного забезпечення Європейської системи освіти,загальні принципи побудови ІДС можна сформулювати основні принципи побудови системи:

- ***модульність*** - система повинна мати модульну структуру, що дозволить легко проводити її розвиток модернізацію;
- ***простота використання*** - має бути зрозумілою для будь-яких користувачів. Інтерфейс системи має бути детально спланований з урахуванням ергономічних правил;
- ***відкритість*** - система має бути відкритого типу, для безперешкодного розширення її функціональних можливостей, зв'язку з іншими системами, зокрема системами ЄС;
- ***комплексність*** - має надавати користувачам комплексну інформацію, без необхідності відвідування інших ресурсів та ручного пошуку;
- ***універсальність*** - забезпечувати інтереси різних категорій користувачів;
- ***працювати*** у ВЕБ просторі.

У загальному випадку система має бути гнучкою і мати різні модифікації для забезпечення інтересів різних категорій користувачів.

ОСНОВНІ ФУНКЦІЇ СИСТЕМИ

Система має мати наступні основні функціональні можливості:

- ознайомлення з професіями та загальними та професійними компетенціями, необхідними для успішного працевлаштування та роботи за обраною професією;
- ознайомлення з навчальними закладами, що забезпечують отримання обраних професій;
- ознайомлення з програмами дисциплін, що забезпечують можливість отримання необхідних компетенцій;
- визначення психо-фізіолочічних характеристик користувача з метою визначення можливості успішної роботи за обраною професією;
- визначення при необхідності рівня підготовки користувача (від конкретної теми до загальних компетенцій);
- можливість побудови ІОМ;
- інформаційне супроводження проходження ІОМ.

СТРУКТУРА СИСТЕМИ

Система складається з модулів двох типів - функціональні модулі та бази даних. ФМ забезпечують відповідну функціональність. Бази даних можуть бути сконфігуровані під конкретного користувача.

З урахуванням визначених вище вимог базова структура системи має складатись з наступних модулів (рис.1):

- ***ВЕБ-портал*** – представляє собою ВЕБ-інтерфейс і виконує представницькі та загальні управлінські функції - реєстрацію користувачів, забезпечення доступу до обраних програм роботи з системою, відображення інтерфейсів модулів в уніфікованому вигляді;
- ***модуль зв'язку та діагностики*** забезпечує можливість вільного розширення системи, сумісність з іншими системами, проводить діагностику всіх під'єднаних до системи модулів;
- ***модуль професій*** забезпечує можливість перегляду інформації про сфери діяльності, професії, можливі посади, загальні та професійні компетенції, навчальні заклади, що забезпечують відповідну підготовку, програми навчання та дисциплін;
- ***модуль знань*** дозволяє накопичувати та систематизувати опис знань необхідні для визначених професій;
- ***модуль здібностей*** містить перелік здібностей пов'язаних з напрямками професійної підготовки, та у зв'язці із модулем тестування дозволяє визначити до яких професій користувач має хист;
- ***модуль тестування*** – є функціональним модулем системи або може використовуватися зовнішня система,

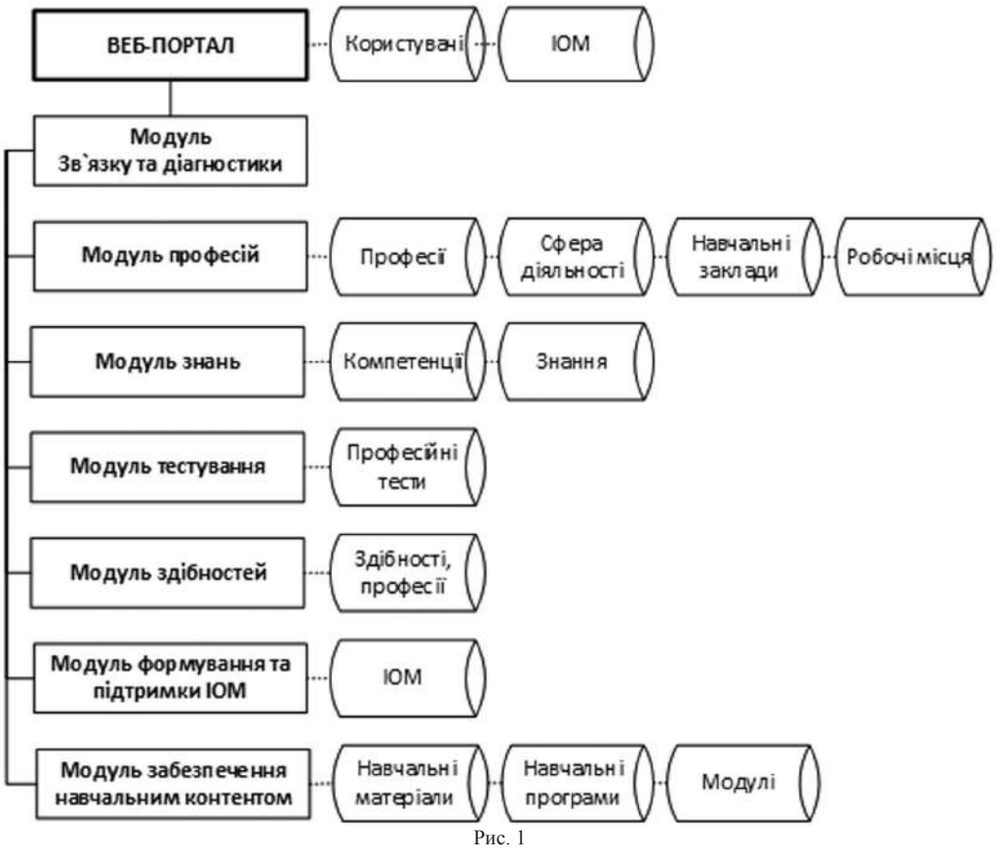


Рис. 1

наприклад, moodle, та набори тестів.

- ***модуль формування та підтримки ЮМ*** – дозволяє на основі отриманих під час роботи із системою даних формувати індивідуальні освітні маршрути користувачів та відслідковувати їх проходження;
- ***модуль забезпечення навчальним контентом (МЗНК)***, завдяки якому зареєстрований користувач отримує доступ до навчальних матеріалів, що необхідні для просування по ЮМ.

Зараз різні модулі системи знаходяться на різних етапах розробки – від уточнення характеристик до дослідної експлуатації [4,6].

ПЕРЕЛІК ПОСИЛАНЬ

1. Рашкевич Ю. М. Болонський процес та нова парадигма вищої освіти: монографія / Ю. М. Рашкевич. – Львів: Львівська політехніка, 2014. – 168 с.

2. Огай М., Романчук Н. Освітні траєкторії населення та їх вплив на професійну мобільність - Аспекти праці - N5, 2014 с. 20-27

3. Міністерство освіти і науки України. Наказ № 368 від 04.04.2016 р. [Електронний ресурс] – Режим доступу: http://old.mon.gov.ua/files/normative/2016-04-19/5432/nmo-368.pdf

4. Богданов А. В. Система інформаційного забезпечення формування та підтримки індивідуального освітнього маршруту[Електронний ресурс] / А.В. Богданов, І.В.Чеповой, П.С.Ухань, Л.Ю.Юрчук // Інформаційні технології і засоби навчання. – 2016. – У друку – Режим доступу: http://journal.iitta.gov.ua/index.php/itlt/article/

5. Впровадження Болонського процессу/ [Електронний ресурс]. – Режим доступу : http://www.osvita.org.ua/bologna/vprov/

6. Здібності для життя. [Електронний ресурс]. – Режим доступу : http://abitask.com/.

Geoinformation technology of topological observability optimization of multiply spatially distributed systems

Vitalii Mokin
Prof. Head of the SACMEG department
Vinnytsia National Technical University
Ukraine, Vinnytsia

Ilona Varchuk
post-graduate student
Vinnytsia National Technical University
Ukraine, Vinnytsia

Summary. Proposed GIS technology to optimize the topological observability multiply spatially distributed systems (MSDS) based on the analysis of the model in the form of a graph of bichromatic this system. It is proposed to carry out the construction of the graph by combining well-known information technology formalization of mathematical models MSDS geoinformation in the parameter space and the integration of mathematical models and geographic information, which significantly accelerates the efficiency and level of automation of the proposed technology.

Keywords: GIS model, geoinformation technology, mathematical model, multiply spatially distributed system, topological observability.

Геоінформаційна технологія оптимізації топологічної спостережуваності багатозв'язних просторово-розподілених систем

Віталій Мокін
Проф. Завідувач кафедри САКМІГ
Вінницький національний технічний університет
Україна, Вінниця

Ілона Варчук
Аспірант
Вінницький національний технічний університет
Україна, Вінниця

Анотація. Запропоновано геоінформаційну технологію оптимізації топологічної спостережуваності багатозв'язних просторово-розподілених систем на основі аналізу моделі у вигляді біхроматичного графу цієї системи. Запропоновано здійснювати побудову цього графу шляхом поєднання відомих інформаційних технологій формалізації математичних моделей БПРС у геоінформаційному просторі параметрів та інтегрування математичних і геоінформаційних моделей, що значно пришвидшує оперативність та рівень автоматизації запропонованої технології.

Ключові слова: геоінформаційна модель, геоінформаційні технології, математична модель, багатозв'язна просторово-розподілена система, топологічна спостережуваність.

Технічні системи, такі як електроенергетичні та електричні системи, дорожньо-транспортні або річкові системи тощо, є багатозв'язними просторово-розподіленими системами. Стан окремих ділянок та складових цих систем характеризується великою кількістю часто взаємозалежних параметрів, що змінюються в часі та просторі. Відповідно трапляються ситуації, коли на одних ділянках чи в окремі інтервали часу ці параметри є спостережуваними або ні.

Для розв'язання задач спостережуваності є досить розвиненим математичний апарат для електроенергетичних систем (ЕЕС) [1, 2]. Для спостережуваності ЕЕС, як геометричних мереж, використовують спеціальний термін — топологічна спостережуваність. Зазвичай, топологічну спостережуваність ЕЕС визначають, використовуючи біхроматичний граф та класичні методи його аналізу та оптимізації — шляхом пошуку максимальних паросполучень між вершинами графа різного типу та оптимізації їх кількості [1].

Але однією з основних проблем застосування цього математичного апарату для ЕЕС чи інших багатозв'язних просторово-розподілених систем (БПРС) є задача побудови відповідного біхроматичного графу. А якщо процеси у БПРС описуються не тільки аналітично, а й алгоритмічно, тоді задача ще ускладнюється.

Для розв'язання даної проблеми пропонується поєднувати інформаційну технологію (ІТ) інтегрування математичних та геоінформаційних моделей [3] та інформаційну технологію формалізації математичних моделей БПРС у геоінформаційному просторі параметрів (ГПП) [4]. ІТ інтегрування математичних та геоінформаційних моделей, запропонована Мокіним В. Б. та Крижановським Є. М., використовує нові підходи щодо проведення аналогії між формалізованим описом математичних моделей та описом просторових об'єктів у геоінформаційних системах (ГІС) і базах даних (БД), що дозволяє прискорити їх інтегрування та розширити аналітичні можливості ГІС за рахунок спеціалізованих обчислювальних пакетів, куди автоматизовано передаються дані ГІС та структура і параметри цих математичних моделей [3].

ІТ формалізації математичних моделей БПРС у ГПП, запропонована Мокіним В. Б. та Гавенком О. В., основана на формалізації аналітичних та алгоритмічних залежностей між параметрами БПРС як системного шару ГІС цих систем, що дозволяє автоматизувати формалізацію та збереження аналітичних та алгоритмічних зв'язків між атрибутивними та просторовими параметрами цих об'єктів і збільшити швидкість адаптації ГІС БПРС до заданої комбінації шарів об'єктів, їх параметрів і залежностей між ними під час моделювання процесів у них [4]. У кожній залежності, яка формалізується у ГПП, усі параметри поділяються на вхідні, та один вихідний, який обчислюється через вхідні, тобто кожна залежність є розв'язком математичної моделі відносно однієї вихідної змінної або алгоритмом обчислення цієї змінної із вхідних змінних [4].

Поєднання цих двох технологій дозволяє автоматизувати формалізацію і збереження усі даних математичних та геоінформаційних моделей, тобто і вхідні дані, і математичні співвідношення та алгоритми їх обробки [5]. Результат їх застосування пропонується доповнити інформаційною технологією оптимізації топологічної спостережуваності цих БПРС (рис. 1). Ця технологія основана на авторському методі трансформації графа, яким, по суті, можна представити ГПП, у біхроматичний граф, для якого є класичні методи аналізу та оптимізації топологічної спостережуваності, що дозволяє поєднувати переваги обох типів методів [6].

Отже, пропонується єдина технологія, яка поєднує 3 вище згадані, структура якої подана на рис. 1. У базі даних зберігаються параметри та вхідні дані математичних та геоінформаційних моделей. ІТ інтегрування математичних та геоінформаційних моделей дозволяє побудувати інтегровану ГІС, в якій можна забезпечити автоматизовану ідентифікацію усіх параметрів за певними співвідношеннями та алгоритмами (за математичними моделями) за наявною атрибутивною (у базі даних) та просторовою (у шарах ГІС) інформацією. А ІТ формалізації математичних моделей БПРС у геоінформаційному просторі параметрів дозволяє у зручному для обробки вигляді у системному шарі чи шарах ГІС зберігати різні комбінації параметрів та структури можливих співвідношень (моделей) між усіма параметрами системи. Розроблена ж авторами ІТ (назвемо її геоІТ) оптимізації топологічної спостережуваності БПРС на основі ГПП перетворює граф моделі, формалізованої у ГПП, у біхроматичний граф (БГ), на основі якого здійснюється подальша перевірка та, за необхідності, – оптимізація топологічної спостережуваності системи.

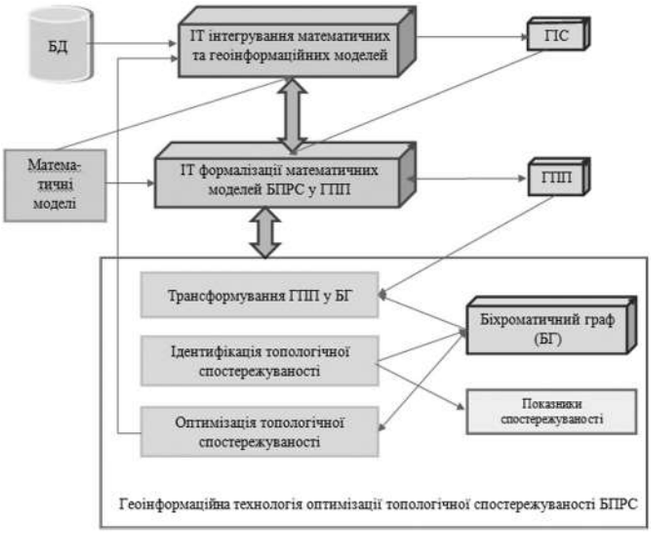


Рисунок 1 – Функціональні зв'язки складових геоінформаційної технології

Алгоритм етапів реалізації запропонованої об'єднаної технології подано на рис. 2.

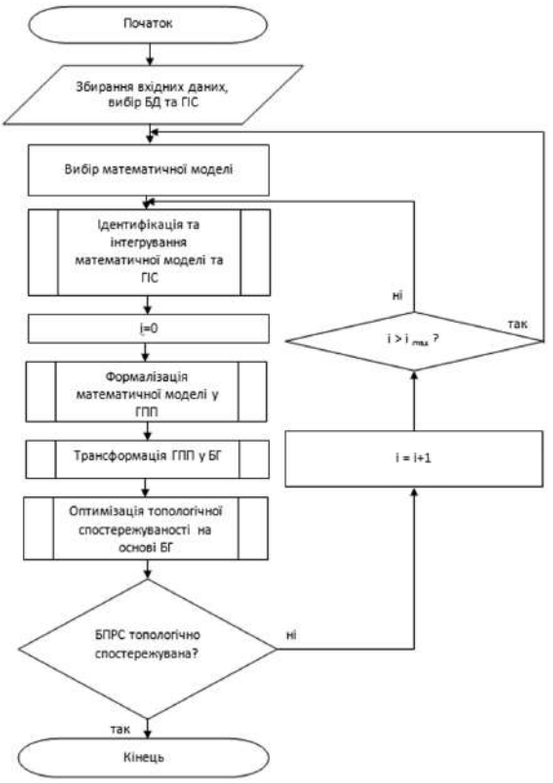


Рисунок 2 – Алгоритм роботи запропонованої технології

Після ідентифікації спостережуваності системи та виявлення ділянок (чи підсистем) та діапазонів часу, в яких система є спостережуваною, здійснюється перевірка того, чи уся система вже є спостережуваною на заданій кількості ділянок та в заданих часових інтервалах. У разі, якщо система ще не є спостережуваною на усіх заданих ділян-

ках та в усьому діапазоні часу, тоді слід повернутись на етап формалізації математичних і геоінформаційних моделей у ГПП, взяти іншу комбінацію вхідних параметрів та структури моделей і провести усі розрахунки заново. При цьому варто підраховувати кількість таких циклів і, якщо їх кількість перевищить задану I_{max}, тоді краще варто повернутись до етапу збирання вхідних даних та вибору математичних і геоінформаційних моделей, де варто пошукати більше даних та принципово інші геоінформаційні та/або математичні моделі.

Таким чином, у роботі охарактеризовано дві інформаційні технології побудови геоінформаційних моделей багатозв'язних просторово-розподілених систем по математичних моделях процесів у них: технологію інтегрування математичних та геоінформаційних моделей та технологію формалізації математичних моделей БПРС у геоінформаційному просторі параметрів (ГПП). Запропоновано яким чином можна, поєднавши ці дві технології, здійснювати автоматизацію синтезу геоінформаційної моделі розподіленої системи по математичних моделях процесів у ній, що дозволить прискорити роботу з геоінформаційними системами і базами даних, а також забезпечить зберігання математичних моделей та усіх вхідних даних у типових форматах, у т.ч. різні варіанти структури і параметрів моделей, в залежності від різних вхідних умов.

Запропоновано доповнити ці дві інформаційні технології третьою – ІТ ідентифікації та оптимізації топологічної спостережуваності БПРС на основі ГПП, яка перетворює граф моделі, формалізованої у ГПП, у класичний біхроматичний граф, на основі якого здійснюється подальший аналіз та оптимізація топологічної спостережуваності системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гамм А.З. Сенсоры и слабые места в электроэнергетических системах / А.З. Гамм, И.И. Голуб. – Иркутск: СЭИ СО РАН, 1996. – 99 с.

2. Савина Н.В. Системный анализ потерь электроэнергии в распределительных электрических сетях в условиях неопределенности: автореф. дис. на соискание учен. степени канд. техн. наук: спец. 05.14.02 «Электростанции и электроэнергетические системы» / Савина Наталья Викторовна. – Благовещенск, 2010. – 20 с.

3. Інформаційна технологія інтегрування математичних моделей у геоінформаційні системи моніторингу поверхневих вод : монографія / В. Б. Мокін, Є. М. Крижановський, М. П. Боцула. - Вінниця: ВНТУ, 2011. – 150 с.

4. Мокін В. Б. Технологія автоматизованої побудови інформаційної моделі для моделювання процесів у багатозв'язних просторово-розподілених системах / В. Б. Мокін, О. В. Гавенко // Вісник Вінницького політехнічного інституту. – Вінниця. – 2013. – № 2. – С. 73-80.

5. Mokin V. B. Method For Determining And Optimization Of Observability Of Multivariable Spatially Distributed Systems Using Geoinformation Parameter Space / V. B. Mokin, I. V. Varchuk // Scientific Bulletin of National Mining University. — 2015. — Issue 5. — Pages 105-111.

6. Варчук І. В. Технологія синтезу геоінформаційної моделі розподіленої системи за математичними моделями процесів у ній / Ілона Вячеславівна Варчук // Вісник Вінницького політехнічного інституту. – 2016. – №2. – С. 20–25.

СИСТЕМИ КЕРУВАННЯ

Synthesis of watching linear device of management system

Repnikova N.B.
National Technical University of Ukraine "Kyiv Polytechnic Institute", Faculty of Informatics and Computer Science
Ukraine, Kyiv

Shumada K.O.
National Technical University of Ukraine "Kyiv Polytechnic Institute", Faculty of Informatics and Computer Science
Ukraine, Kyiv

Summary

The paper presents a new approach to the synthesis of watching device nonlinear system using the method of «backstepping». Analytical relations for the calculation of the matrices, the model of the nonlinear system in an environment Matlab / Simulink and the results of synthesis were developed.

Key words: nonlinear control systems, Lyapunov function, watching device, feedback vector.

Синтез спостерігаючого пристрою нелінійної системи керування

Репнікова Н.Б.
НТУУ «КПІ», факультет Інформатики та Обчислювальної Техніки
Україна, Київ

Шумада К.О.
НТУУ «КПІ», факультет Інформатики та Обчислювальної Техніки
Україна, Київ

Анотація

В роботі запропоновано новий підхід синтезу спостерігаючого пристрою нелінійної системи з використанням методу "backstepping". Виведено аналітичні співвідношення для розрахунку відповідних матриць. Розроблено модель нелінійної системи в середовищі Matlab/Simulink та отримані результати синтезу.

На сьогоднішній день в теорії автоматичного керування базовою є задача управління об'єктами різноманітної природи. Реальна фізична система майже завжди описується нелінійними диференціальними рівняннями. Для розв'язання задачі керування такою системою зазвичай або нехтують нелінійностями, або використовують метод лінеаризації системи. Однак це може привести до погіршення показників якості системи та виникненню похибок. Таким чином, питання розроблення нових та вдосконалення існуючих методів синтезу нелінійних систем є актуальним.

Розглянемо об'єкт керування, що описується нелінійними диференціальними рівняннями:

$$\begin{cases} \dot{x}_1 = k_1 x_1^2 x_2 \\ \dot{x}_2 = k_2 u + x_1 x_2 \end{cases},$$

де x_1, x_2 – змінні стану,

k_1, k_2 – постійні додатні коефіцієнти,

u – управління.

Для знаходження закону керування, що буде забезпечувати асимптотичну стійкість системи, застосуємо метод «backstepping». Перевагою методу є можливість застосування його до нелінійних систем керування без нехтування нелінійностей.

Основою методу «backstepping» є використання функцій Ляпунова для пошуку закону керування. [1]

Розглянемо перше рівняння системи і приймемо x_2 в якості керуючої змінної v_1:

$$\dot{x}_1 = k_1 x_1^2 v_1$$

Вибравши позитивно визначену функцію $V(x_1) = \frac{x_1^2}{2}$, вважаючи, що вона є можливою функцією Ляпунова. Її похідна за часом:

$$\dot{V}(x_1) = k_1 x_1^3 v_1,$$

Для знаковизначеності виразу функція v_1 має мати вигляд :

$$v_1 = -l_1 x_1,$$

Де l_1 – постійний додатній коефіцієнт.

Визначимо помилку між реальним значенням x_2 та бажаним значенням v_1, як:

$$z_2 = x_2 - v_1 = x_2 + l_1 x_1$$

Отримуємо нову систему відносно змінних x_1 та z_2 Розв'язавши її аналогічним чином, отримаємо кінцевий вираз для закону керування:

$$u = \frac{1}{k_2}\left(-l_1 l_2 x_1 - l_2 x_2 - x_1 x_2 - l_1\right)$$

Для можливості управління зазначеним об'єктом за допомогою зворотних зв'язків за станом, потрібно мати повну інформацію про змінні стану об'єкту керування. Як відомо, таку задачу вирішує спостерігаючий пристрій. На теперішній час існує декілька методів синтезу спостерігаючого пристрою.

Пропонуємо для оцінки змінних стану об'єкту керування використовувати еталонну модель, структура і параметри якої співпадають з структурою та параметрами досліджуваної нелінійної системи.

Ідея нового підходу синтезу нелінійного спостерігаючого пристрою полягає в тому, що визначається помилка виходу нелінійної системи та моделі в функції заданих початкових станів.

Для забезпечення відновлення станів такої моделі необхідно ввести деяку матрицю H, що являла б собою матрицю коефіцієнтів зворотного зв'язку спостерігаючого пристрою з елементами $h_1(t)$ та $h_2(t)$.

Для отримання аналітичної залежності коефіцієнтів матриці H від параметрів вихідної системи розглянемо початковий момент часу роботи системи. Визначимо похибку значення початкового стану змінних x_i:

$$\delta_i = x_i - x_i^{'},$$

Де $x_i^{'}$ – початковий стан спостерігаючого пристрою.

Обчислимо значення керування *u'* за вище отриманою формулою за станами та та підставимо отриманий результат у вихідні рівняння станів системи. Доповнимо рівняння системи до еталонної моделі спостерігаючого пристрою та отримаємо:

$$\begin{cases} \dot{x}_1 = k_1 x_1'^2 x_2' + \delta_1 h_1 \\ \dot{x}_2 = k_2 u' + x_1' x_2' + \delta_2 h_2 \end{cases}$$

З наведеної системи рівнянь знаходимо аналітичний вираз для обчислення значень матриці H.

Для порівняння запропонованого підходу з існуючим було розроблено модель спостерігаючого пристрою з використання матриць Якобі та попередньою лінеаризацією об'єкта керування. На Рис.1 та 2 показані результати моделювання роботи спостерігаючого пристрою та відновлення змінних станів з використанням лінеаризації об'єкта керування.[2]

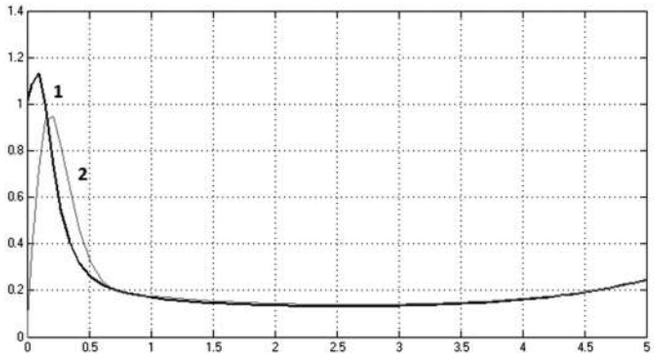


Рис.1 Зміна стану $x_1(t)$
1 – об'єкт керування; 2 – спостерігаючий пристрій

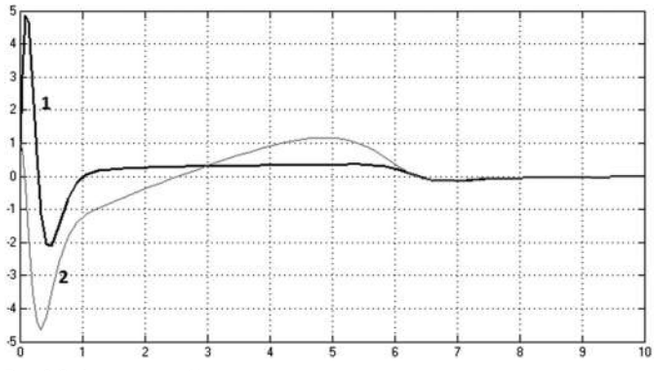


Рис.2 Зміна стану $x_2(t)$
1 – об'єкт керування; 2 – спостерігаючий пристрій

Результати моделювання розробленого нелінійного спостерігаючого пристрою представлені на Рис. 3 і 4:

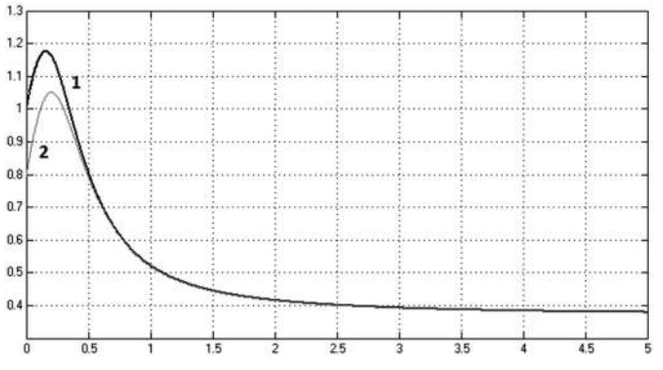


Рис.3 Зміна стану $x_1(t)$
1 – об'єкт керування; 2 – спостерігаючий пристрій

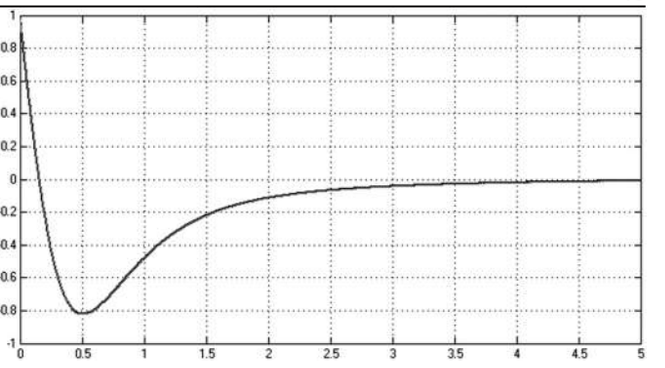


Рис.4 Зміна стану $x_2(t)$

Як видно з графіків (Рис.1, 2), розроблений спостерігаючий пристрій відновлює стани з значною похибкою. Варто також відзначити, що даний метод синтезу є досить громіздким і для реальних систем його реалізувати дуже складно.

Графіки перехідних процесів на Рис. 3, 4 показують, що розроблений спостерігаючий пристрій відновлює змінні станів з достатньо високою точністю.

ВИСНОВКИ

Таким чином, розроблено новий підхід до синтезу нелінійного спостерігаючого пристрою на базі методу "backstepping" з використанням функцій Ляпунова. Даний підхід дозволяє отримувати аналітичні вирази для визначення матриці H, яка зводить похибку відновлення змінних станів нелінійної системи до 0.

СПИСОК ЛІТЕРАТУРИ

1. Zhou J. Adaptive Backstepping Control of Uncertain System / Jing Zhou, Changyun Wen. – Berlin: Springer, 2008. – 243 p.

2. Синицин И. Н. Методы статистической линеаризации (обзор) // АиТ, 1974. №5. С. 82–94.

Bayesian networks in diagnostics and adaptive control problems

Pysarenko A.
National Technical University of Ukraine "KPI"
Ukraine, Kyiv

Tischenko D.
National Technical University of Ukraine "KPI"
Ukraine, Kyiv

Approaches to solving adaptive filtering problem by means of dynamic Bayesian networks in control systems of complex objects under uncertainty conditions were considered. The issues of automatic online diagnosis of the controls system elements state with intelligent diagnostic block based on Bayesian network were developed.

Keywords: Bayesian network, dynamic Bayesian network, adaptive filtering, diagnostics, data mining

Применение байесовских сетей в задачах диагностики и адаптивного управления

Писаренко А.В.
Национальный технический университет Украины «КПИ»
Украина, Киев

Тишенко Д.В.
Национальный технический университет Украины «КПИ»
Украина, Киев

Рассмотрены подходы к решению задач адаптивной фильтрации в системах управления сложными объектами в условиях неопределенности средствами динамических байесовских сетей. Затронуты вопросы автоматической онлайн диагностики состояния элементов системы управления с помощью интеллектуального диагностического блока, включающего байесовскую сеть.

Ключевые слова: байесовская сеть, динамическая байесовская сеть, адаптивная фильтрация, диагностика, интеллектуальный анализ данных

К современным системам управления, повсеместно используемым в различных сферах человеческой деятельности, предъявляются повышенные требования с точки зрения показателей качества и надежности. Обеспечение заданных показателей качества процессов в условиях неопределенности среды их протекания является важнейшей задачей, не утрачивающей своей актуальности на любом этапе развития теории и практики управления. Бурное развитие методов, впоследствии объединенных под общим названием «интеллектуальный анализ данных», позволяет значительно продвинуться в решении указанных задач. Неопределенности, рассматриваемые в теории управления (параметрические, сигнальные, функциональные, структурные, статистические) по-разному учитываются в процессе проектирования, начиная от классических адаптивного и робастного подходов и заканчивая такими методами интеллектуального анализа данных как искусственные нейронные сети, нечеткая логика, нейро-нечеткие сети, байесовские сети, генетические алгоритмы, деревья решений и другие.

Данные в виде последовательностей возникают во многих областях науки и техники. Динамические системы порождаютвременные последовательности данных в ходе своего функционирования, которые могут быть получены с использованием всевозможных датчиков и косвенных инструментов получения данных. В одних задачах имеется необходимость в онлайн анализе, когда данные поступают в режиме реального времени, в других достаточно автономного режима анализа, когда данные уже были собраны.

СЕТИ БАЙЕСА

Как известно, байесовская сеть (БС) кодирует вероятностные взаимосвязи между представляющими интерес переменными, используя ациклический ориентированный граф. Такое представление имеет ряд преимуществ для анализа данных. Поскольку модель описывает зависимости между переменными, она справляется с ситуациями, когда некоторая информация отсутствует. Также БС может быть использована для установления причинно-следственных связей и, следовательно, может быть использована для получения понимания о проблемной области и прогнозирования. К тому же подобное представление позволяет объединить априорную информацию и данные. В результате БС являются очень популярными вероятностными моделями и сегодня интерес к обучению БС с использованием измерительных (статистических) данных значителен.

Примерами задач, в которых БС являются идеальными инструментами для анализа, являются, например: вычисление общей надежности системы с учетом надежности отдельных ее компонентов и их взаимодействия [1]; систем безопасности, где БС используются в качестве инструмента для оценивания факта проникновения в сеть[2]; в судебно-медицинской экспертизе [3]; также в медицинской диагностике, поддержке принятия решений, диагностике состояния датчиков, поиске информации, управлении рисками и робототехнике и других [4]. Таким образом, БС находят все более широкое применение в различных областях человеческой деятельности и на деле доказывают свою перспективность и успешность.

По типу используемых данных различают следующие типы БС: дискретные, непрерывные и гибридные, а по способу описания режимов функционирования объектов исследования – статистические и динамические. Очевидно, что каждый тип БС пригоден для определенного класса задач. Поскольку предметом данной работы является адаптация и диагностика в системах автоматического управления, рассмотрим более подробно применимость БС в указанных задачах.

Скрытые модели Маркова и фильтр Калмана популярны, поскольку они относительно простые и удобные для практического применения. Тем не менее, они ограничены в своих потенциальных возможностях. Динамические байесовские сети (ДБС) обобщают модели Маркова, позволяя представить пространство состояний в разложенном виде, а не в качестве единой дискретной случайной величины. Также ДБС обобщают фильтр Калмана, позволяя произвольно распределять вероятности, а не только в линейной гауссовой постановке задачи.

Исходя из вышеизложенного, представляется обоснованным использование возможностей ДБС для адаптивной фильтрации (наблюдения состояний управляемого процесса), а также для поддержки принятия управленческих решений в автоматическом или полуавтоматическом режимах функционирования систем управления.

В онлайн анализе данных общая задача состоит в том, чтобы прогнозировать будущие наблюдения, учитывая все предыдущие измерения до текущего момента времени, которые мы обозначим как Поскольку, как правило, мы не можем быть уверены в будущем, мы хотели бы получитьпрогноз. Кроме того, мы хотели бы знать, насколько он правдоподобен, чтобы его учитывать. Поэтому необходимо вычислить распределение вероятностей по возможным будущим наблюдениям, обозначив их как $P\left(x_{k+h}\middle|x_{1:k}\right)$, где $h>0$ – горизонт прогнозирования, – на сколько шагов в будущее необходимо оценить прогноз.

В случае управления системой возникает необходимость прогнозирования будущих состояний (выходов) системы в зависимости от значений входных величин. Если обозначить через прошедшие значения входов, а через $u_{k+1:k+h}$ следующие h значений входов, то задача будет состоять в вычислении $P\left(x_{k+h}\middle|u_{1:k+h},x_{1:k}\right)$.

Если рассматривать модели в пространстве состояний, то их использование является более предпочтительным, чем классических регрессионных описаний временных последовательностей по ряду причин: они не подвержены эффекту влияния конечного окна прошлых измерений; они могут легко обрабатывать дискретные и многомерные входы и выходы; а также они могут легко включать исходные предварительные знания. Например, есть переменные, которые мы не можем измерить, но в тоже время необходимо оценить весь вектор состояний.

Есть много способов представления моделей в пространстве состояний, наиболее распространенными из которых являются скрытые марковские модели (СММ), а также модели, используемые в алгоритмах фильтра Калмана (ФК).

Трудности применения СММ. Предположим, что необходимо отслеживать состояний объекта. Пусть каждое состояние может принимать одно из N возможных состояний. Тогда $\mathbf{x}[k]=(x_1[k],...,x_n[k])^T$ может иметь N^n возможных значений. Это означает, что требуется экспоненциальное количество параметров для определения модели наблюдения, то есть потребуется много данных, для построения модели. Кроме того, логический вывод займет экспоненциальное время, например, проход вперед-назад занимает $O(TN^{2n})$, где T – длина выборки.

Трудность применения ФК заключается в том, что он рассматривается для линейных динамических систем, что не всегда приемлемо для решения некоторых задач. Обычной практикой является использование либо расширенного, либо так называемого нечувствительного (unscented) фильтра Калмана в качестве решения в таких случаях.

Использование ДБС, являющихся в некоторой степени обобщением СММ и ФК позволит в значительной степени избежать указанных трудностей. Динамическая байесовская сеть представляет собой способ расширения сетей Байеса для моделирования распределения вероятностей над набором случайных величин $\mathbf{z}[1],\mathbf{z}[2],....$ Мы будем объединять переменные в $\mathbf{z}[k]=(\mathbf{u}[k],\mathbf{x}[k],\mathbf{y}[k])$ для представления входных, скрытых и выходных переменных модели в пространстве состояний.

Последовательность построения модели в форме БС можно представить в виде следующих шагов: (1) – углубленный анализ исследуемого процесса (объекта) с целью установления особенностей его функционирования и выявления родительских и дочерних переменных; (2) – выявление существующих моделей процесса и анализ возможности их дальнейшего использования; (3) – установление существующих связей между переменными процесса с помощью специальных тестов и экспертного оценивания; (4) – сокращение размерности задачи построения модели; (5) – масштабирование и дискретизация переменных; (6) – определение семантических ограничений для модели; (7) – оценивания структур моделей-кандидатов с использованием оптимизационных процедур, то есть поиск альтернативных моделей в форме БС; (8) – параметрическое обучение модели; (9) – анализ качества и выбор лучшей из моделей-кандидатов; (9) – применение выбранной модели для решения поставленной задачи; (10) – формирование вероятностных выводов относительно выбранных переменных с построенной моделью (моделями), анализ

качества полученного результата. Окончательным результатом применения модели в форме БМ (ДБМ) является вычисление вероятности попадания значения в некоторый интервал.

Динамическая байесовская сеть определяется парой $(B_1, B_{\rightarrow})$, где B_1 это байесовская сеть, определяющая предшествующую $P(\mathbf{z}[1])$, а $B_{\rightarrow}$ – временная БС с двумя временными срезами (2ВБС), определяющая $P(\mathbf{z}[k]|\mathbf{z}[k-1])$, как

$$P(\mathbf{z}[k]|\mathbf{z}[k-1]) = \prod_{i=1}^{n} P(\mathbf{z}_i[k]|Pa(\mathbf{z}_i[k])),$$

где $\mathbf{z}_i[k]$ – i-я вершина в момент, $Pa(\mathbf{z}_i[k])$ – предки $\mathbf{z}_i[k]$ на графе. Вершины в первом срезе 2ВБС не имеют каких-либо параметров, связанных с ними, но каждая вершина во втором срезе 2ВБС имеет связанное с ней распределение условной вероятности, определяемой $P(\mathbf{z}_i[k]|Pa(\mathbf{z}_i[k]))$ для всех $k > 1$. Предки вершины, $Pa(\mathbf{z}_i[k])$, могут находиться как в том же временном срезе, так и в предыдущем.

Таким образом, разница между ДБС и СММ заключается в том, что ДБС представляет вектор состояний как набор случайных величин $x_1[k],...,x_n[k]$ т. е., использует распределенное представление состояния. В противоположность этому, в СММ пространство состояний состоит из одной случайной величины $\mathbf{x}[k]$. Разница между ДБС и ФК состоит в том, что распределение условных вероятностей ФК должно быть линейным (одинаковым для всех переменных), в то время как ДБС позволяет произвольное распределение.

Рассмотрим теперь особенности использования БС для задач диагностики. Определим диагностику, как вычисление апостериорного распределения дискретной переменной. Во многих приложениях будущие значения дискретных переменных представляют гипотезы о состоянии оборудования или режима работы, а предыдущие значения переменных представляют наблюдаемые параметры. Модели для таких систем, как правило, заданы наблюдаемыми переменными (назовем X) как потомки переменных гипотез (назовем H), тогда цель состоит в том, чтобы вычислить $P(H|X)$.

Бывает, что гипотеза H имеет несколько возможных значений, которые соответствуют набору взаимоисключающих гипотез, например, нормальную работу и несколько взаимоисключающих режимов отказа. Значение $P(H|X)$ представляет собой список чисел, которые указывают вероятность для каждой из возможных гипотез; как правило, если вероятность одного или нескольких видов отказов велика, то необходимо будет предпринимать какие-либо действия. Однако не существует отдельных вычислений, предшествующих вычислению вероятности каждого вида неисправности; вероятность одного или нескольких сбоев любого рода это лишь $\sum_k \Pr$ («отказ типа k»), так что для того, чтобы вычислить агрегированную вероятность отказа, необходимо провести весь расчет $P(H|X)$.

Таким образом в задачах диагностики наиболее применимы статические БС (дискретные либо непрерывные), позволяющие оценить вероятность возникновения определенного типа отказа из заранее определенного перечня возможных.

СПИСОК ЛИТЕРАТУРЫ

Langseth H., Portinale L., Bayesian networks in reliability // Reliability Engineering and System Safety 92(1), – 2007, 92–108.

Zhang S., Song S., A Novel Attack Graph Posterior Inference Model Based on Bayesian Networks // Journal of Information Security, – 2011,vol 2, pp 8 – 27.

Dawid A., Mortera J., Pascali V, Van Boxel D., Probabilistic expert systems for forensic inference from genetic markers // Scandinavian Journal of Statistics, – 2002,vol 29 № 4, pp 577 – 595.

Pourret O., Bayesian networks: a practical guide to applications // Wiley, – 2008.

Терентьев А.Н. Методы построения байесовских сетей / А.Н. Терентьев, П.И. Бидюк // Межведомственный научно-технический сборник „Адаптивные системы автоматического управления”. -Днепропетровск: Системные технологии, 2005. - № В. - С. 130 - 141.

Information management system of educational process in higher education

Shumeyko N.S.
student of NTUU «KPI»
Faculty of Informatics and Computer Technology
Department of Computer-Aided Management and Data Processing Systems
Kyiv, Ukraine

Abstract: The information society formation substantially modifies the educational system. Modern education is characterized by a high level of technological equipment. System integration of information and communication technologies in the educational process is very important now for current reformation and modernization of the education system. The problems of the software information in educational environment of the university and the characteristics of a learning management system Moodle are studied.

Keywords: learning management system, distance learning, educational technologies.

Інформаційні системи керування навчальним процесом у вищій школі

Шумейко Микола Сергійович

студент 4 курсу НТУУ «КПІ»,
факультет інформатики та обчислюваної техніки,
кафедра автоматизованих систем обробки інформації та управління
Київ, Україна

Анотація: Формування інформаційного суспільства суттєво модифікує освітню систему. Сучасна освіта характеризується високим рівнем технологічної оснащеності. Системна інтеграція інформаційних і телекомунікаційних технологій в освітній процес є ключовим моментом нинішньої реформації і модернізації системи освіти. Розглядаються питання програмного забезпечення інформаційного освітнього середовища вузу, характеристики системи управління навчанням Moodle.

Ключові слова: системи керування навчанням, дистанційне навчання, освітні технології.

Основні характеристики сучасного світу - інформатизація і глобалізація. Соціальний наслідок глобалізації - поява нових форм освіти, які починають набувати риси широкомасштабних систем. Дистанційна освіта відноситься до цих систем. Для позначення методів дистанційного навчання часто використовується термін e-Learning (електронне навчання). У загальному випадку цей термін означає використання нових технологій мультимедіа та Internet для підвищення якості навчання за рахунок поліпшення доступу до ресурсів і сервісів, а також віддаленого обміну знаннями та спільної роботи.

У процесі розвитку технологій e-Learning, в кінці 90-х років минулого століття, виникли системи категорії Learning Management System (LMS) - системи керування навчанням, що включають засоби не тільки для організації та контролю використання комп›ютерних курсів та тренінгів, а й для адміністрування навчального процесу в цілому, в тому числі його традиційних форм. Найбільш поширені на сьогоднішній день системи - Moodle, Blackboard, Sakai. LMS служить фундаментом для побудови всього процесу електронного навчання. Будь-яка LMS передбачає наявність стандартних модулів (кошти розробки курсів, курси, система керування контентом, система керування учнями, система взаємодії з Internet).

Студент отримує від LMS можливості доступу до навчального порталу, який є відправною точкою для доставки всього навчального контенту, вибору відповідних траєкторій навчання на основі попереднього і проміжних тестувань, використання додаткових матеріалів.

Система керування навчанням включає в себе реєстрацію і контроль доступу користувачів до системи і до навчального контенту, організації слухачів в групи, для надання їм загальних курсів і складання звітності, управління аудиторними і викладацькими ресурсами. LMS відповідає також за інтеграцію додаткових елементів навчального процесу (практичні заняття, лабораторні роботи, засоби спільної роботи, посилання на зовнішні матеріали та інші).

Останнім часом активно розвивається новий клас систем, що реалізують керування навчальним контентом - Learning Content Management System (LCMS). На відміну від LMS подібні системи концентруються на завданнях керування змістом навчальних програм, а не процесом навчання і орієнтовані не на менеджерів і студентів, а на розробників контентів, фахівців з методологічної компонуванні курсів і керівників проектів навчання. Яскравим представником систем класу LCMS є розробка компанії IBM - Lotus Workplace Collaborative Learning. Значне число навчальних закладів орієнтується на безкоштовно розповсюджене програмне забезпечення, яке дозволяє організовувати дистанційний навчальний процес. Подібною програмою є Moodle. (Modular Object-Oriented Dynamic Learning Environment - модульне об›єктно-орієнтоване динамічне навчальне середовище), призначена для організації взаємодії між викладачем і учнями, для організації традиційних дистанційних курсів, а також для підтримки очного навчання (використовується вже в 160 країнах [1]). Використовуючи Moodle, викладач може створювати курси, наповнюючи їх вмістом у вигляді текстів, допоміжних файлів, презентацій, опитувальників і т.п. Для використання Moodle досить мати будь-який web-браузер, що робить використання цього навчального середовища зручним як для викладача, так і для учнів.

За результатами виконання учнями завдань викладач може виставляти оцінки і давати коментарі. Таким чином, Moodle є і центром створення навчального матеріалу і забезпечення інтерактивної взаємодії між учасниками навчального процесу. Особливу цінність Moodle являє собою саме тому, що його можна використовувати як з системою Windows, так і з пакетами вільно поширюваного програмного забезпечення: Linux, Ubuntu, OpenOffice.org та інше.

Завдяки своїм функціональним можливостям система набула великої популярності і успішно конкурує з комерційними LMS [2, 3].

Moodle дає можливість проектувати, створювати і далі керувати ресурсами інформаційно-освітнього середовища [4]. Система має зручний, інтуїтивно зрозумілий інтерфейс. Викладач самостійно, вдаючись тільки до допомоги довідкової системи, може створити електронний курс

і управляти його роботою. Практично у всіх ресурсах і елементах курсу в якості полів введення використовується зручний WYSIWYG HTML редактор, крім того, можна вставляти таблиці, схеми, графіка, відео, анімацію та ін.

Використовуючи зручний механізм настройки, укладач курсу може, навіть не володіючи знанням мови HTML, легко вибрати колірну гамму і інші елементи оформлення навчального матеріалу.

Рецензент: ас., к.т.н. Коган Алла Вікторівна, НТУУ «КПІ», Київ, Україна.

ПЕРЕЛІК ПОСИЛАНЬ

1. Анисимов, А. М. Работа в системе дистанционного обучения MOODLE : учебное пособие / А. М. Анисимов. – Харьков, ХНАГХ, 2009. – 292 с.

2. Практика электронного обучения с использованием Moodle / А. В. Андреев, С. В. Андреева, И. Б. Доценко – Таганрог : ТТИ ЮФУ, 2008.

3. Bobrova, L., Marinova, O. Information Educational Environment-The Basis for Work with Remote Audience / L. Bobrova, O. Marinova / World Applied Sciences Journal 27 (Education, Law, Economics, Language and Communication): 5 15–518, 2013.

4. Bobrova, L., Smirnova, N. Management-Probleme von Bildungs-Prozess bei der Arbeit mit dem Remote-Publikum / L. Bobrova, N. Smirnova, European Applied Scientific: modern approaches in scientific researches, 1st International scientific conference. ORT Publishing. Stuttgart. 2012. P. 130–133.

Рецензент: к.т.н. асист. каф. АСОІУ НТУУ «КПІ» А. В. Кочан

ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

Development of automated system of search and granting media content recommendations based on the preferences of other users

Starushyk Artem Mykolajovych
student, FICT, NTUU "KPI"
Ukraine, Kiev

Annotation

This article describes an automated system of search and granting media content recommendations based on individual profiling of users and Pearson correlation criterion as similarity measure.

Keywords: data analysis, recommendation system, Pearson correlation coefficient, collaborative filtering

Розробка автоматизованої системи надання та пошуку рекомендацій медіаконтенту на основі вподобань інших користувачів

Старушик Артем Миколайович
студент, ФІОТ, НТУУ «КПІ»
Україна, Київ

Анотація

В роботі описано автоматизовану систему надання та пошуку рекомендацій медіаконтенту на основі індивідуального профілювання користувачів та критерію кореляції Пірсона, як міри подібності.

ОСНОВНА ЧАСТИНА

Відомо[1], що кількість медіаконтенту різного типу стрімко зростає. Одночасно з цим зростає потреба у фільтруванні даних, відповідно до смаків та вподобань індивідуального користувача, що є основним завданням рекомендаційних систем.

Рекомендаційна система – підклас системи фільтрації даних, яка визначає об'єкти, яким користувач може надати перевагу. Однією із найпоширеніших стратегій створення рекомендаційних систем є колаборативна фільтрація[2] – один із методів визначення рекомендації, прогнозу для користувача в системі. Основне припущення колаборативної фільтрації полягає в тому, що користувачі, які однаково оцінювали будь-які об'єкти в минулому, схильні схожим чином оцінювати інші об'єкти в майбутньому. Існує два типи колаборативної фільтрації: user-based – в основі лежить пошук схожих користувачів та item-based – базується на пошуку схожих об'єктів.

Серед основник проблем рекомендаційних мереж варто виділити:

- проблема «лінивого користувача» – користувач хоче отримати рекомендацію здійснивши якомога меншу кількість дій, оцінивши найменшу достатню кількість об'єктів, тощо;
- проблема швидкодії – система повинна моментально реагувати на запити користувачів, тому алгоритм визначення рекомендацій повинен бути швидким та ефективним;
- проблема мінливості – користувач може змінювати свою точку зору, переоцінювати будь-які об'єкти, що унеможливлює збереження отриманих до цього результатів, оскільки наступні будуть відрізнятися.

Розроблена система використовує гібридний user-based підхід, де користувачу пропонується заповнити спеціально створену анкету, для оцінки різних аспектів його вподобань. Користувач може як надавати, так і отримувати рекомендації. На основі подібності анкет користувач отримує рекомендації, які надали інші, найбільш подібні до нього користувачі.

Система розроблена з використанням мова програмування C# та технології для створення веб-застосунків ASP.Net 5 MVC 6[3]. За замовчуванням дана технологія використовує шаблон проектування MVC (Model - View - Controller)[4], що полегшує розробку та тестування функціоналу. Доступ до бази даних здійснюється за допомогою Entity Framework – це об›єктно - реляційний модуль співставлення, що дозволяє розробникам .NET працювати з реляційними даними за допомогою об›єктів, спеціалізованих для доменів. Сама ж база даних створена з використанням підходу Code First – один із підходів в Entity Framework, де за допомогою програмного коду описується модель та анотація даних, а фреймворк автоматично створює базу даних на основі цього. За замовчуванням для авторизації використовується технологія ASP.Net Identity.

Основні архітектурні рішення в системі:

- шаблон проектування MVC (Model - View - Controller) технологія ASP.Net використовує за замовчуванням. Model – опис структури і логіки даних в системі. Controller – центральна ланка архітектури, яка обробляє запит та надає відповідь. View – зовнішнє представлення даних.
- Шаблон проектування Repository[5] для роботи з базою даних. Створено узагальнений CRUD[6] інтерфейс *IRepositry<T, TKey>*, де T – клас моделі з якою працює репозиторій, TKey – тип первинного ключа. CRUD – (англ. create, read, update, delete) – чотири базові операції для роботи із персистентними сховищами даних[7] створення, зчитування, редагування, видалення. Класи-репозиторії реалізують даний інтерфейс відповідно до особливостей моделі, з якою вони працюють.
- ASP.Net 5 MVC 6 має вбудований контейнер інверсії управління (Inversion of Control, IoC), що значно полегшує реалізацію принципу інверсії залежностей, оскільки не потрібно підключати сторонні бібліотеки для впровадження залежностей (Dependency injection, DI). Наприклад в програмному коді системи в конструктори контролерів передаються потрібні інтерфейси, а вибір відповідної реалізації покладається на IoC контейнер. Це дозволяє збільшити гнучкість системи, забезпечити слабку зв'язність об'єктів, полегшити підтримку, розширення та заміну елементів в системі;
- використовується асинхронна модель методів контролерів, для раціонального використання потоків при роботі з базою даних. Методи дій контролера повертають значення *Task<IActionResult>* та позначені ключовим словом *async*;
- для формування анкети, визначення рекомендацій та обчислення міри подібності створено узагальнені інтерфейси IQuestionary<T>, IRecommendationManager<T>, IMeasure<T> та їх реалізації. Залежності впроваджені в IoC контейнер.

Для запобігання підробки запитів використовуються токени, за допомогою функції *ValidateAntiForgeryToken()* та атрибута *[ValidateAntiForgeryToken]*.

Алгоритм надання рекомендацій користувачу наступний:

1) перевірка на наявність анкети:
- якщо анкета є – перейти до пункту 2;
- якщо анкета немає – перейти до процедури заповнювання анкети;

2) дані з анкети представити у вигляді одновимірного масиву, де кожен елемент – це числове значення відповіді;

3) порівняти дані поточного користувача з даними інших користувачів з анкетами, оцінивши їх подібність за допомогою коефіцієнта кореляції Пірсона[8];

4) виділити найбільш подібних користувачів та відобразити рекомендації, які вони надали.

Для оцінки подібності профілів користувачів використовується коефіцієнт кореляції Пірсона, який якісно оцінюється за шкалою Чеддока[9]. Подібність масивів виведених даних визначається за наступним алгоритмом:

1) розрахунок середніх значень:

$$\overline{x} = \frac{\sum x_i}{n}, \overline{y} = \frac{\sum y_i}{n}$$

де, x_i та y_i – i-ті елементи масивів оцінок користувачів, n – потужність масивів;

2) розрахунок дисперсій:

$$S^2(x) = \frac{\sum x_i^2}{n} - \overline{x}^2, S^2(y) = \frac{\sum y_i^2}{n} - \overline{y}^2$$

де, x_i та y_i – i-ті елементи масивів оцінок користува-

чів, $\overline{x}$ та $\overline{y}$ – середні значення масивів, n – потужність масивів;

3) розрахунок середньоквадратичних відхилень:

$$\sigma_x = \sqrt{S^2(x)}, \sigma_y = \sqrt{S^2(y)}$$

де, $S^2(x)$ та $S^2(y)$ – дисперсії для масивів оцінок;

4) розрахунок коефіцієнта кореляції Пірсона:

$$r_{xy} = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{n\sigma_x\sigma_y}$$

де, x_i та y_i – i-ті елементи масивів оцінок користувачів, $\overline{x}$ та $\overline{y}$ – середні значення масивів, σ_x та σ_y – середньоквадратичні відхилення, n – потужність масивів;

5) оцінка отриманого коефіцієнту за шкалою Чеддока (таблиця 1);

Таблиця 1 – Шкала Чеддока

Кількісна міра кореляції	*Якісна міра кореляції*
0,1 - 0,3	Слабка
0,3 - 0,5	Помірна
0,5 - 0,7	Помітна
0,7 - 0,9	Висока
0,9 - 1	Дуже висока

6) на основі отриманих даних відбираються користувачі для яких якісна міра кореляції «помітна», «висока» та «дуже висока», тобто коефіцієнт кореляції Пірсона має значення від 0,5 і вище.

Таким чином поточний користувач може знайти інших користувачів з подібними вподобаннями та отримати рекомендації, які вони надали.

Перевага алгоритму:

- простота реалізації та розуміння;
- не потребує великих затрат обчислювальної потужності та додаткової пам'яті;

Недоліки алгоритму:

- час роботи прямо пропорційний кількості зареєстрованих користувачів із наявними анкетами.

Особливістю реалізованої системи є те, що анкета користувача містить конкретний перелік питань, що дозволяє в загальному оцінити його вподобання та контролювати проблему «лінивого користувача». Система надає рекомендації, створені іншими користувачами, тобто результат базується не лише на «холодному розрахунку», а також і на людському факторі. З технологічної точки зору систем кросплатформенна, що дозволяє розробляти та розгортати її на платформах Windows, Mac, Linux.

ВИСНОВКИ

Відомо напевне, що кількість даних медіаконтенту з часом лише збільшуватиметься, а тому тема рекомендаційних мереж залишатиметься актуальною, як для окремих користувачів, так і для медіа гігантів, таких як Netflix, Last.fm, IMDb тощо. Тому нові підходи, методи, проміжні ланки, критерії з'являтимуться з високою частотою, що сприятиме розвитку даної теми.

В роботі було описано розробку автоматизованої системи надання та пошуку рекомендацій медіаконтенту з використанням user-based підходу на основі індивідуальних профілів користувачів з використанням критерію кореляції Пірсона, як міри подібності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Основы медиарекламного бизнеса [Електронний ресурс] – Режим доступу: http://vi-minsk.com/business/publications/86/

2. Н.С. Лєсна. Методи пошуку та фільтрації інформації з використанням колаборативної фільтрації - / Н. С. Лєсна, С. М. Гайдамака // Системи обробки інформації. - 2013. - № 5. - с. 80-82.

3. The ASP.Net Site [Електронний ресурс] – Режим доступу: http://www.asp.net/

4. Справочник «Паттерны проектирования» [Електронний ресурс] – Режим доступу: http://design-pattern.ru/patterns/mvc.html

5. Справочник «Паттерны проектирования» [Електронний ресурс] – Режим доступу: http://design-pattern.ru/patterns/repository.html

6. Techopedia [Електронний ресурс] – Режим доступу: https://www.techopedia.com/definition/25949/create-retrieve-update-and-delete-crud

7. Techopedia [Електронний ресурс] – Режим доступу: https://www.techopedia.com/definition/8842/persistence-computing

8. J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient - №1 Feb. 1988. - pp. 59-66

9. R. E. Chaddock. Principles and methods of statistics - №7 Apr. 2014. – pp. 39-40

Рецензент: к.т.н, доц. каф. АУТС НТУУ «КПІ» Катін П.Ю.

Application of modern opengl tools in Qt

Tumanov Vladyslav Valeriyovych
NTUU "KPI"
Ukraine, Kyiv

Overviewing the way of combining functionality of OpenGL with Qt–framework to achieve the highest performance when processing large amounts of graphic objects.

Keywords: Qt-framework, OpenGL, shader, Vertex Buffer Object, performance

Застосування сучасних засобів opengl в Qt

Туманов Владислав Валерійович
НТУУ “КПІ”
Україна, Київ

Розглянуто спосіб поєднання функціоналу OpenGL з засобами Qt–фреймворка для досягнення найвищої швидкодії при обробці великої кількості графічних об’єктів.

ВСТУП

Qt–фреймворк[1] широко відомий завдяки зручним та легким у використанні інструментам для розробки інтерфейсу користувача. Також він надає різноманітні засоби для роботи з графікою та растровими зображеннями. Головним інструментом для малювання в робочій області вікна є клас QPainter, який забезпечує оптимізовані функції для виконання більшої частини графічних операцій. Він може намалювати все: від простих ліній до складних форм, таких як сектори і криві Безьє. Він також може відображати вирівняний текст і растрові зображення. QPainter може працювати на будь-якому об’єкті, який успадковує клас QPaintDevice.

Проте такий спосіб відображення має один суттєвий недолік – низьку швидкодію. Справа в тому, що QPainter взаємодіє з API операційної системи, який в свою чергу використовує процесор для виконання графічних операцій на найнижчому рівні без застосування апаратного прискорення, яке може надати GPU (відео карта). В результаті, маємо значне сповільнення при малюванні вже порядку 10^5 таких примітивів як прості лінії. Така проблема не буде фігурувати при статичному відображенні об’єктів, проте гостро стане коли потрібно буде постійно перемальовувати вид для, наприклад, обертання всієї графіки навколо довільної вісі.

Підхід представлений у даній роботі базується на використанні можливості Qt суміщати в собі найпотужніші засоби OpenGL[2] для обробки графіки на апаратному рівні що в результаті приносить майже миттєву швидкодію, яка є критичним показником для програм, призначення яких полягає в обробці великої кількості графіки. Особливо це стосується різних САПР, таких як AutoCAD, OrCAD, Graphite та інші.

Програмна основа

Для роботи з OpenGL, Qt пропонує спеціальний набір класів, головним з яких є QGLWidget, що надає функціональні можливості для відображення OpenGL графіки, вбудовані в додаток Qt. Все що вам необхідно, це відкрито (public) успадкувати від нього свій власний клас, і надалі використовувати його як і будь-який інший об’єкт класу QWidget з урахуванням того, що тепер ви можете виконувати графічні операції як за допомогою QPainter, так і використовуючи засоби OpenGL.

QGLWidget надає три зручні віртуальні функції, які користувач бібліотеки може перевизначити в підкласі для виконання типових завдань OpenGL:

paintGL() – призначений для опрацювання графіки засобами OpenGL. Викликається щоразу, коли віджет повинен бути оновлений;

resizeGL() - встановлює параметри виду екрана, проекції на екран і т.д. Викликається щоразу, коли віджет змінює розміри (а також, коли він показаний в перший раз, тому що всі новостворені віджети автоматично отримують подію зміни розміру);

initializeGL() - встановлює контекст відображення OpenGL, визначає списки відображення і т.д. Викликається один раз, під час ініціалізації вікна.

Іншим важливим класом є QGLFunctions, який надає крос-платформенний доступ до OpenGL API. Він пропонує функціонал, який доступний для більшості настільних або вбудованих реалізацій OpenGL. Найдоцільніше використовувати цей клас в формі захищеного (protected) наслідування.

ПРОГРАМНА РЕАЛІЗАЦІЯ

Розроблюваний клас має бути наслідуваний від QGLWidget відкрито та від QGLFunctions захищено. Використання апаратного прискорення означає малювання всієї графіки за допомогою шейдерів[3] (програми, які компілюються в інструкції безпосередньо для GPU), та представлення великих кількостей об’єктів у формі Vertex Buffer Object (VBO). Отже, клас повинен містити наступні групи полів:

- об’єкти QGLShaderProgram (програм шейдерів) для всіх примітивів та текстур з якими працюватиме програма;
- поля атрибутів для кожної програми шейдерів (тип GLuint);
- поля для збереження матриць виду та проекції, які будуть використовуватись при малюванні примітивів шейдерами (тип QMatrix4x4);
- поля для збереження uniform–індексів, що зв’язують матрицями програми та матрицями шедерів (тип GLint);
- поля для збереження ідентифікаторів VBO для кожного буфера(тип GLuint);
- інші поля, такі як масиви для зберігання буферів, тексту програм шейдерів, координат текстур і т. ін.

Перевизначений метод initializeGL() повинен починатись з виклику initializeGLFunctions(), який ініціює спеціальні функції OpenGL, в контексті поточного вікна. Після цього можна створювати буфери та об’єкти програм шейдерів, супроводжуючи їх перевіркою результату зв’язування, яка покаже чи підтримуються такі шейдери конкрет-

ною системою, та чи нема синтаксичних помилок в коді шейдерів. З вже скомпільованих шейдерів отримуються атрибути та юніформи (глобальні змінні в шейдерах) для їх подальшого використання в якості прив'язки до буферів.

Перевизначення методу resizeGL() полягає в отриманні відповідної до поточних розмірів вікна матриці проекції за допомогою стандартних засобів OpenGL, а також, надає можливість легко створити фонове зображення відповідних розмірів:

- спершу створюється порожня картинка QPixmap з розмірами вікна;
- за допомогою QPainter на ній малюється те що має бути фоном фікна;
- за допомогою методу bindTexture() на основі об'єкта Qpixmap створюється текстура, яку тепер можна намалювати відповідним шейдером як статичне фонове зображення.

Так можливості стандартних класів Qt органічно доповнюють можливості OpenGL.

І, врешті-решт, в методі paintGL() має бути реалізовано малювання буферів об'єктів на клієнтській області вікна за допомогою відповідних програм шейдерів.

ПОРІВНЯННЯ ШВИДКОДІЇ

Для демонстрації переваг засобів OpenGL над стандартними методами Qt мною було розроблено експериментальну програму, в якій реалізовано два вікна, які малюють задану кількість ліній на робочій області. Одне вікно використовує для цього об'єкти QPainter та QPainterPath. Інше – використовує метод, описаний в даній статі. Програма заміряє час, витрачений на малювання змісту вікна. Експеримент проведений з використанням процесора AMD E–450 APU with Radeon HD Graphics. Заміри проводились для 10^5, 10^6, та 10^7 ліній. Усереднені результати наведені в таблиці 1.

Табл. 1. Результати замірів

	QPainer	***OpenGL***
10^4	0.031 с	0.002 с
10^5	0.125 с	0.016 с
10^6	2.43 с	0.031 с

ВИСНОВКИ

Отримані в результаті експерименту результати цілковито підтверджують переваги засобів OpenGL над стандартними класами Qt–феймворка, які б оптимізовані їх методи не були. Єдиною значною перевагою готових класів є їх зручність та простота, адже використання сучасних засобів OpenGL потребує значного досвіду та знань. Написання складних шейдерів або, так званого, «ренедерера» графіки (модуля, який відповідає за правильну та впорядковану роботу засобів OpenGL в межах програми) не є тривіальними завданнями, тож якщо швидкодія не є приорітетом, то краще і простіше використати стандартні методи Qt.

ПЕРЕЛІК ПОСИЛАНЬ

1. Qt Documentation [Електронний ресурс] – Режим доступу: https://doc.qt.io

2. OpenGL introduction [Електронний ресурс] – Режим доступу: https://open.gl

3. OpenGL ES Shading Language Reference [Електронний ресурс] – Режим доступу: http://www.shaderific.com/glsl/

Рецензент: к.т.н, доц. каф. АУТС НТУУ «КПІ» Катін П.Ю.

Plagiarism search engine for a given subject area

Mahdych Bohdan Valentinovich
Ukraine, Kiev**,** Faculty of Information and Computer Science, NTUU "KPI"
«IT-Enterprise», Ukraine, 02140, Kiev, pr. Bagana 14a, 4th floor

Summary
This work is a brief overview of the basic algorithm " Shingle" and its modifications based on the particular area of application.

Keywords: plagiarism, algorithm "Shingle", duplicate, copy, copyright protection

Система пошуку плагіату для заданої предметної області

Магдич Богдан Валентинович
Україна, Київ, Факультет Інформатики та Обчислювальної техніки, НТУУ «КПІ»
«IT-Enterprise», Україна, 02140, Київ , пр. Бажана 14а, 4й поверх

Анотація
В роботі надається на короткий огляд базового алгоритму «Шинглів» та його модифікації, що базується на конкретній області його застосування.

ВСТУП

Стрімкий розвиток мережі Інтернет поряд з збільшення комп'ютерної грамотності широкої аудиторії користувачів стало причиною широкого використання плагіату в різних сферах людської діяльності, зокрема в галузі освіти та науки. В освіті проблема плагіату є доволі гострою та активно обговорюється. На жаль, більшість студентів не проводять багато часу в бібліотеках та архівах через те, що більшість інформації можна «скачати» з джерел мережі Інтернет, або запозичити роботи студентів минулих років навчання і при здачі матеріалу викладачеві банально замінити прізвище на своє. Таким чином, в теперішній час студенти досить часто порушують авторське право на інтелектуальну власність. Тому для викладачів досить актуальною є проблема виявлення плагіату в роботах студентів. В статті запропонована модифікація відомого алгоритму «шинглів» для поліпшення результативності пошуку плагіату.

АЛГОРИТМ «ШИНГЛІВ»

Уди Манбер в 1994 році першим в світі висловив ідею пошуку дублікатів, а в 1997 році Андрій Бродер оптимізував і довів її до логічного завершення, давши ім'я даній системі - «алгоритм шинглів».

На сьогоднішній день він є найпопулярнішим алгоритмом для пошуку плагіату в довільних текстах. Він розроблений для пошуку копій (дублікатів) розглянутого тексту в документі та є потужним інструментом, що може боротися з проявами плагіату.

Метод заснований на представленні текстів у вигляді множини послідовностей фіксованої довжини, що складаються із сусідніх слів. При значному перетині таких множин документи будуть схожі один на одного.

Розберемо через які етапи проходить текст, що буде порівнюватись:

- Канонізація тексту і видалення «стоп-символів» і «стоп-слів»;
- Розбиття на шингли;
- Обчислення хешів шинглів;
- Порівняння та визначення результату.

КАНОНІЗАЦІЯ ТЕКСТУ

Канонізація тексту приводить оригінальний текст до єдиної нормальної форми. Текст очищається від «стоп-символів» і «стоп-слів» (прийменників, сполучників, знаків пунктуації тощо), які не повинненні брати участь у порівнянні. У деяких випадках також пропонується видаляти з тексту прикметники, так як вони не несуть смислового навантаження. Також на етапі канонізації тексту можна приводити іменники до називного відмінку, єдиного числа або залишати від них тільки корінь. На виході цього етапу ми маємо текст, очищений від «сміття» і готовий до порівняння (рис. 1).

Рисунок 1. Послідовність робіт етапу канонізації в алгоритмі «шинглів»

РОЗБИТТЯ НА ШИНГЛИ

Шингли - виділені підпослідовності слів. Необхідно з порівнюваних текстів виділити підпослідовності слів, що йдуть один за одним по N штук (довжина шингла). Таким чином, розбиваючи текст на підпослідовності, ми отримаємо набір шинглів. Дії по кожному із пунктів виконуються для кожного з порівнюваних текстів (рис. 2).

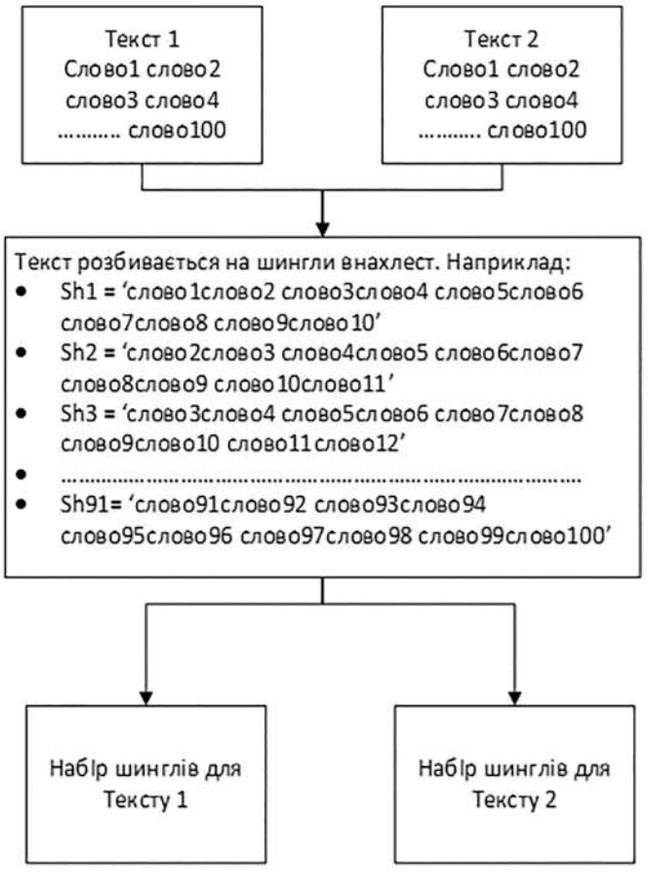


Рисунок 2. Послідовність робіт етапу розбиття на шингли

ОБЧИСЛЕННЯ ХЕШІВ ШИНГЛІВ

Принцип алгоритму шинглів полягає в порівнянні випадкової вибірки контрольних сум шинглів (підпослідовностей) двох текстів між собою. Тепер у кожного з текстів є свої набори шинглів. Слід розрахувати контрольну суму кожного з шинглів. Для розрахунку можна використовувати відомий алгоритм CRC3 2 (рис. 3).

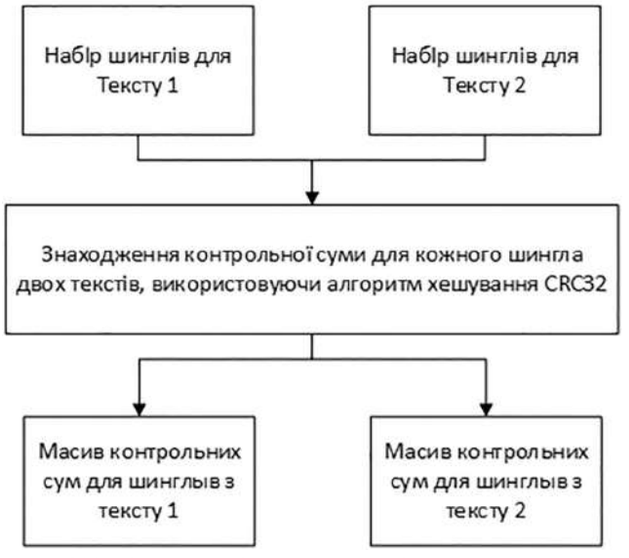


Рисунок 3.Послідовність робіт при знаходженні контрольних сум шинглів

ПОРІВНЯННЯ ТА ВИЗНАЧЕННЯ РЕЗУЛЬТАТУ

На цьому етапі порівнюємо між собою всі елементи першого масиву з відповідними елементами другого масиву, зчитуємо відношення однакових значень та з цього отримуємо кінцевий результат (рис. 4).

Рисунок 4.Послідовність при порівнянні елементів масиву

МОДИФІКАЦІЯ АЛГОРИТМУ ДЛЯ ЗАДАНОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

Заданий алгоритм є досить дієвим, проте в кожній області є свої сталі вирази (фразеологізми) та свої найбільш часто вживані слова, які необхідно також видаляти з тексту. Тому заданий алгоритм потрібно підлаштовувати під кожну предметну область. Для покращення роботи алгоритму були введені деякі модифікації.

Даний алгоритм не враховує слів синонімів. Досить часто здійснюється перефразовування речення для обходження системи анти-плагіату. Тому доцільним буде введення множини синонімів слів, що часто вживаються в заданій предметній області. Дане введення є досить великою модифікацією, що покращить заданий алгоритм.

Також необхідно ввести множину синонімів словосполучень, тому що досить часто автоматизовані системи надають новим словосполученням достатньо великий відсоток унікальності. Тому при знаходженні слів, що потрапляють в задані множини (множина синонімів часто вживаних слів, множина синонімів словосполучень), їх також потрібно враховувати в відсоток подібності текстів.

Задані модифікації будуть ще більш краще знаходити подібні тексти, та виявляти подібні тексти в заданій предметній області.

ПЕРЕЛІК ПОСИЛАНЬ

1. W-SHINGING — URL : https://en.wikipedia.org/wiki/W-shingling

2. MOSS. A System for Detecting Software Plagiarism. — URL: https://theory.stanford.edu/~aiken/moss/

3. Miller C., ‘‘Detecting duplicates: a searcher’s dream come true’’ — URL: https://www.questia.com/magazine/1G1-9065495/detecting-duplicates-a-searcher-s-dream-come-true

Generative and Multi-staged Programming. Lightweight Modular Staging

Veres D. S.
student at National Technical University of Ukraine "Kyiv Polytechnic Institute", FICT
Ukraine, Kyiv

This article presents a study of methods and tools that can allow to use high-level functional programming languages in a field of systems-level programming, where every drop of performance matters.

One of the ways to achieve our goal is to use the approach of generative programming and modern technique called Light-weight Modular Staging.

Keywords: generative programming, multi-staged programming, Scala, Light-weight Modular Staging, meta-programming

Генеративне та багатоетапне програмування. Lightweight Modular Staging

Верес Д. С.
студент Національного технічного університету України "Київський політехнічний інститут", ФІОТ
Україна, Київ

Дана стаття пропонує дослідження методів та інструментів, що дозволять використовувати функціональні мови програмування високого рівня у сфері системного програмування, де ефективність роботи програми є дуже важливою.

Одним з шляхів досягнення даної мети є використання підходу генеративного програмування і сучасної техніки під назвою *Light-weight Modular Staging.*

Ключові слова: генеративне програмування, мультиетапне програмування, Scala, Light-weight Modular Staging, мета-програмування.

ГЕНЕРАТИВНЕ ТА БАГАТОЕТАПНЕ ПРОГРАМУВАННЯ

Одною з найбільших проблем в розробці програмного забезпечення на сьогоднішній день є те, що продуктивність програмного забезпечення все більше і більше розходиться з продуктивністю програміста. Ця проблема посилюється цілою низкою тенденцій:

- Тактова частота процесора більше не подвоюється кожних 18 місяців. Замість цього, апаратні частини сучасних комп'ютерів стають все більш паралельними та розподіленими. Програмісти повинні використовувати різноманітні моделі низькорівневого програмування, щоб найкращим чином використовувати наявні апаратні ресурси.
- Мови програмування і методології розробки приділяють все більше уваги абстракції та узагальненню, що дозволяє програмістам створювати великі системи з простих, але універсальних частин. Через це, компілювання програм на мовах високого рівня у ефективний код є принципово важким завданням, оскільки компілятори не мають можливості оптимізувати предметно-специфічні операції.
- З рухом в сторону "великих даних" та високих навантажень, поширення мобільних пристроїв та вбудованих систем, зростає попит на високоефективне програмне забезпечення.

Ці тенденції змушують команди розробників програмного забезпечення, вкладати великі ресурси в оптимізацію свого коду. З точки зору розробки програмного забезпечення, це бентежить, тому що ручна оптимізація вимагає, щоб програмісти, по суті, відмовились від усіх найкращих практик і переваг високорівневого програмування, таких як використання абстракцій, спільності і модульності. Програми, оптимізовані вручну важко читати, важко підтримувати, важко тестувати і, таким чином, зростає ймовірність, виникнення помилок і вразливостей в системі.

Альтернативою до даного підходу є створення генераторів програм – програм, які після запуску повертають програмний код як вихідну інформацію. Генератор програм може виконувати будь-які обчислення під час створення коду цільової програми, по цій причині, код такої програми може бути дуже ефективним.

Генеративне програмування, як підрозділ мета-програмування, описує практичні аспекти створення програм, які генерують інші програми під час свого виконання.

Підхід етапного (staged) або мульти-етапного (multi-staged) програмування описує ідею поділу обчислювального процесу на різні явні етапи, тобто програма поточного рівня генерує код, який буде оброблений на наступному етапі. Дана концепція описана у [1], її автори помітили,

що виконання багатьох програм може бути розділено на етапи за частотою виконання, або за наявністю потрібної для виконання інформації. Підхід мультиетапного програмування (Multi-stage Programming), запропонований Taha і Sheard [2], робить дані етапи явними та дозволяє програмістам відкласти виконання визначених виразів мови програмування до певного етапу. Для цього вони запропонували мову програмування MetaML, яка використовує оператори цитування, що є дуже схожими на макроси мов Lisp та Scheme.

Етапний підхід зазвичай використовується як метод генерування програм. Виконання багатоетапної програми дозволить отримати об'єктну програму, яка може бути скомпільована та виконана на потрібному наборі даних. Даний підхід використовується для спрощення як процесу генерації програми, так і для спрощення трансформацій.

Генеративне програмування дозволяє користуватись рядом цікавих шаблонів програмування. До них відносять mixed-stage типи даних, що містять як статичні, так і динамічні частини. Іншим шаблоном є спеціалізація згенерованого коду засновуючись на статично доступній інформації.

LIGHTWEIGHT MODULAR STAGING

Lightweight modular staging (LMS) [3] – це сучасна техніка мульти-етапного програмування, що базується на типах: замість використання синтаксичного цитування (як у MetaML), дана техніка використовує потужну систему типів мови Scala для визначення виразів на наступних етапах. В той час, як будь-який звичайний Scala вираз з типом Int, String чи в загальному T виконується нормально, LMS використовує спеціальний конструктор типів Rep[T], особливістю якого є те, що всі операції над об'єктами типів Rep[Int], Rep[String] чи Rep[T] згенерують код, який виконає дані операції пізніше.

Ось простий приклад використання LMS:

```
val test = new LMS_Driver[Int,Int] {
    def power(b: Rep[Int], x: Int): Rep[Int] =
        if (x == 0) 1 else b * power(b, x - 1)

    def snippet(x: Rep[Int]): Rep[Int] = {
        power(x,5)
    }
}
test(4)
=> 1024
```

У наведеному прикладі створюється об'єкт LMS_Driver. В середині даного об'єкта можна використовувати Rep типи та відповідні їм специфічні операції. Метод snippet є головним методом даного об'єкта (аналог метода/функції main у більшості C-подібних мовах програмування). Функція test виконає функцію snippet з заданим символьним вхідним параметром. Цей виклик дасть можливість провести повне рекурсивне обчислення функції power (оскільки вона є функцією поточного етапу) та запише індивідуальні вирази у вирази у вигляді проміжного представлення (intermediate representation) по мірі знаходження їх у коді. По виході з функції snippet, драйвер зкомпілює згенерований код і завантажить його у програму. Вихідний згенерований код буде мати наступний вигляд:

```
class Anon11 extends ((Int) => (Int)) {
    def apply(x0:Int): Int = {
        val x1 = x0 * x0
        val x2 = x0 * x1
        val x3 = x0 * x2
        x3
    }
}
```

Виконані перетворення автоматично очищені від типізації: в описі функції power, лише змінна b є динамічною (тип Rep[Int]), всі інші частини програми обробляються статично на на етапі генерації коду. Далі, вираз test(4) виконує згенерований код та повертає результат 1024.

ВНУТРІШНЯ БУДОВА **LMS.**

Техніку LMS називають легкою (lightweight), оскільки вона реалізовується у вигляді бібліотеки та не є невід'ємною частиною мови програмування. Її називають модульною (modular) тому, що вона надає свободу у визначенні доступних операцій для роботи з Rep[T] значеннями. Для клієнтського коду, LMS надає абстрактні інтерфейси, які розширяють вибрану функціональність типу T до роботи з типом Rep[T]:

```
trait Base {
    type Rep[T]
}
trait IntOps extends Base {
    implicit def unit(x: Int): Rep[Int]
    def infix_+(x: Rep[Int], y: Rep[Int]): Rep[Int]
    def infix_*(x: Rep[Int], y: Rep[Int]): Rep[Int]
}
```

Всередині, даний інтерфейс створює проміжне представлення (intermediate representation - IR) над яким можна проводити перетворення та з рештою з його допомогою згенерувати фінальний код.

Дану структуру також можна розглядати як методи поверхневого або глибокого влаштовування об'єктів мови проміжного представлення [3]. Наприклад, такі методи як infix_+ можуть слугувати розумними конструкторами, які виконують певні оптимізації коду на льоту під час генерування проміжного представлення [4]. Використання даного підходу та певні налаштування компілятора мови Scala, дає можливість перенести стандартні для мови умовні оператори чи присвоєння змінних у проміжне представлення, за допомогою їх перевизначення у виклики методів [5].

ВИСНОВКИ

В даній роботі описані методи досягнення "абстрації без нарікань" ("abstraction without regret"): отримання високої ефективності роботи програм використовуючи код високого рівня.

Загалом, описані техніки дають можливість використовувати мови програмування високого рівня для створення

швидкого та ефективного коду, хоча це і здавалось неможливим ще кілька років тому.

Дана стаття демонструє що LMS та Scala є хорошим вибором для досягнення такої мети, проте близькі результати можна отримати використовуючи й інші мови.

В той час, як уже існує багато шаблонів, які можуть використовуватись у генеративному програмуванні (використання функцій вищого порядку для композиції фрагментів коду, об'єкти та класи для опису багатоетапних структур даних та модульності під час генерації коду), дана галузь знаходиться лише у періоді становлення та відчуває брак нових технік програмування.

ДЖЕРЕЛА

1. U. Jørring and W. L. Scherlis. Compilers and staging transformations. In POPL, 1986.

2. W. Taha and T. Sheard. Metaml and multi-stage programming with explicit annotations. Theor. Comput. Sci., 248(1-2):211–242, 2000

3. J. Svenningsson and E. Axelsson. Combining deep and shallow embedding for EDSL. In TFP, 2012

4. Z. DeVito, J. Hegarty, A. Aiken, P. Hanrahan, and J. Vitek. Terra: a multi-stage language for high-performance computing. In PLDI, 2013.

5. T. Rompf, N. Amin, A. Moors, P. Haller, and M. Odersky. Scalavirtualized: Linguistic reuse for deep embeddings. Higher-Order and Symbolic Computation (Special issue for PEPM'12).

Реуензент: к.т.н., доц. каф. ТК НТУУ «КПІ» Остапченка К. Б.

УДК 004.451.622

Comparative analysis of the digital watermarking robustness to the modifications of container

Valchuk K.I.
Student of the National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, Kiev

Dorogyy Y.Y.
Cand. Sc. (Eng.), Assoc. Prof, National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, Kiev

Abstract

The goal of this work is to compare methods of hiding text in an image file using algorithms Least Significant Bit (LSB), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) based steganography. An estimation of the particular technique is evaluated based on the parameters BRE, PSNR, robustness and capacity of the cover image.

Keywords: steganography, LSB, DCT, DWT, stego-image, BRE, PSNR

Порівняльний аналіз стікості водяних знаків до модифікацій контейнера

Вальчук Х.І.
Студентка Національного Технічного Університету України «Київський Політехнічний Інститут»
Україна, Київ

Дорогий Я.Ю.
Кандидат технічних наук, доцент, Національний Технічний Університет України «Київський Політехнічний Інститут»
Україна, Київ

Анотація

Дана стаття проводить аналіз методів кодування тексту в контексті зображення із використанням алгоритмів найменш значимого біту (LSB), дискретного косинусного перетворення (DCT) і дискретного вейвлет-перетворення (DWT) на основі стеганографії. Оцінка продуктивності окремих методів та якості зображення цих трьох методів заснована на оцінці параметрів BER, PSNR, ємності та стійкості зображення.

Ключові слова: стеганографія, LSB, DCT, DWT, стего-зображення, BRE, PSNR

Стеганографія трансформувалася у цифрову стратегію приховування даних у різноманітних мультимедійних формах, таких як графічні зображення, аудіо чи навіть відео-файли. Така технологія отримала назву Цифрові Водяні Знаки (ЦВЗ). Для забезпечення високих захисних властивостей, ЦВЗ мають бути максимально стійкими до спотворення контейнеру (зовнішній мультимедійний об'єкт). В даній роботі проводиться порівняльний аналіз стійкості

групи методів вбудовування ЦВЗ в нерухомі зображення до ряду спотворюючих впливів.

У даній роботі ми будемо розглядати систему, в яку вбудовується ЦВЗ, представлене цифровим зображенням M в інше цифрове зображення C, яке називається контейнером. Заповнений контейнер S може піддаватися всіляким перетворенням із подальшим виокремленням ЦВЗ. Введемо позначення E, T і D для описання процесів вбудовування ЦВЗ в контейнер, перетворення заповненого контейнера та виокремлення ЦВЗ, тоді схему стенографічної системи можна представити у вигляді:

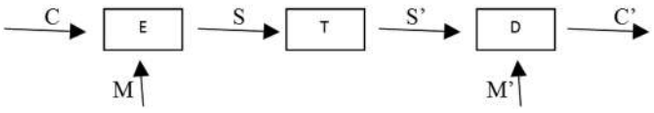


Рисунок 1 – Схема стенографічної системи

ОБ'ЄКТИ ТЕСТУВАННЯ

До просторових методів стеганографії можна віднести метод найменш значущого біту LSB (Least Significant Bit) та метод Куттера-Джордана-Боссена (надалі метод Куттера), а до частотних – методи дискретного косинусного перетворення DCT (Discrete Cosine Tranform) та техніки дискретного вейвлет-перетворення DWT (Discrete Wavelet Transform).

Об'єктами тестування стали наступні методи вбудовування ЦВЗ: стандартний метод LSB, метод Куттера, двокоефіцієнтний алгоритм Коха-Жао на основі технології косинусного перетворення та алгоритм вейвлет-перетворення Джин-Пенг. Останній алгоритм є представником класу вбудовування ЦВЗ у низькочастотну площину вейвлет-розкладу і має вищу стійкість ніж при використанні високочастотних піддіапазонів.

МЕТОДИКА ОЦІНКИ СТІЙКОСТІ ТА УМОВИ ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

У якості тестового зображення використовувались 10 напівтонових та 10 кольорових зображень із роздільною здатність 640x640 пікселів.

При порівнянні різних методів необхідно для кожного з них підібрати оптимальну силу вбудовування: ЦВЗ повинен бути максимально стійкий до впливів, але візуально зміна зображення не повинна відтворюватись.

Оцінка рівня спотворення зображення здійснювалася за допомогою параметра PSNR (пікове співвідношення сигнал / шум), який розраховується за формулою (1):

$$PSNR = 10 \lg \frac{255^2 * M * N}{\sum_{x,y}^{M,N} \left(f(x,y) - \hat{f}(x,y) \right)^2} (1),$$

Де $f(x,y)$ – контейнер, $\hat{f}(x,y)$ – стегоконтейнер, x, y – координати пікселей зображення, M, N – висота та ширина зображення. Спотворення вважаються помітними, якщо $PSNR \geq 43$ дБ. Алгоритми були перевірені на стійкість вбудованого ЦВЗ до найбільш поширених шкідливих впливів: компресії зображення із втратами (JPEG), зашумлення, розпливчатої фільтрації і масштабування

Співвідношення вбудованої та виокремленої інформації після зовнішнього впливу на контейнер оцінювався за допомогою коефіцієнта помилкових біт BER (Bit Error Rate), що приймає значення в діапазоні від 0 до 1:

$$BER(S, S'') = \frac{\sum p_i}{N} (2),$$

де S_i – j-й біт оригіналу вбудованої строки; S''_i – біт виокремленої строки; N –загальна кількість біт.

Рівень вейвлет-розкладання зображення-контейнера для стеганографічних алгоритмів обирався від 1 до 3. Вибір коефіцієнтів сили вбудовування ЦВЗ вибирався виходячи з умови вибору максимального рівня спотворень зображення-контейнера, що не приводить до візуалізації артефактів. ЦВЗ уявлялося у вигляді псевдовипадкової бітової послідовності. Довжина бітової послідовності відповідала максимальної місткості зображення-контейнера.

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ

У тесті на стійкість до JPEG-стиску, зображення стискалося за алгоритмом JPEG в повному діапазоні значень параметра K_{JPEG} (від 0 до 100), відповідального за якість стиснення. Результат представлений на рисунках 2 та 3. Найбільшу стійкість для напівтонових зображень до цього типу спотворювань показав метод Джин-Пенга і найменш стійким виявився просторовий метод LSB.

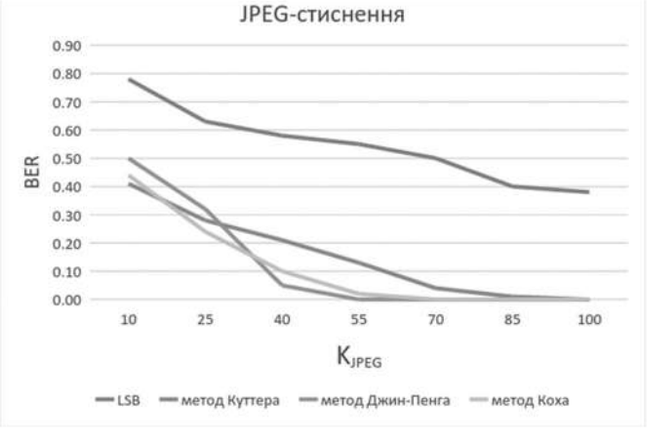


Рисунок 2 – Залежність BER напівтонового зображення від коефіцієнта стиснення JPEG

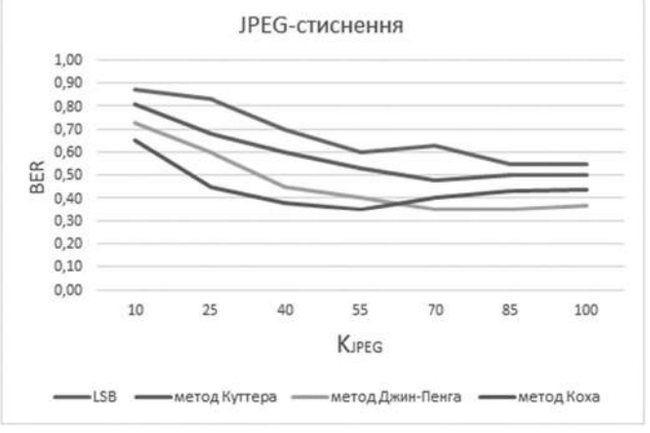


Рисунок 3 – Залежність BER кольорового зображення від коефіцієнта стиснення JPEG

Для тестування був обраний медіанний фільтр, що підвищує різкість зображення, з розміром вікна 3x3 пікселя. Результати експерименту представлені на рисунках 4 та 5. Найвищу стійкість показав метод Коха, і найгірше проявив себе LSB для обох типів зображення, проте діапазон значень BER для LSD при тестуванні напівтонового зображення ширший, аніж для кольорових.

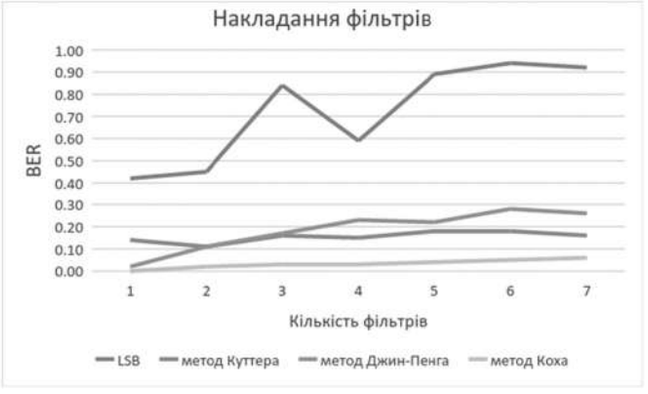


Рисунок 4 – Залежність BER напівтонового зображення від кількості накладених фільтрів

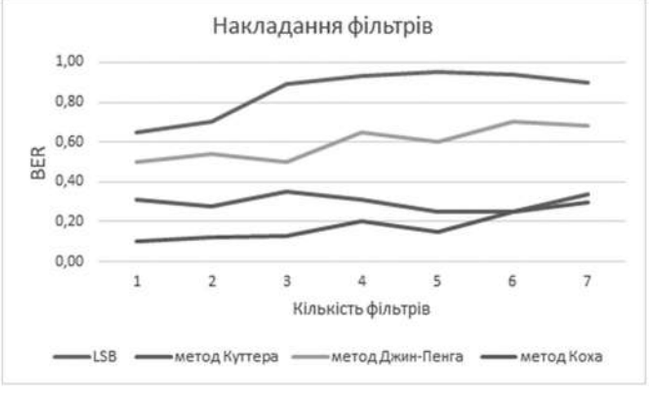


Рисунок 5 – Залежність BER кольорового зображення від кількості накладених фільтрів

Для перевірки стійкості ЦВЗ до зашумлення, в контейнер вносився білий гаусівський шум з нульовим середнім значенням і різними значеннями середньоквадратичного відхилення, що змінюється від 0 у бік зростання до величини, що приводить до такого рівня деградації зображення, при якому його подальше використання в комерційних цілях неможливо. Результат представлений на рисунках 6 та 7. Для кольорових зображень метод Коха виявився менш ефективним при великих значеннях рівня шуму.

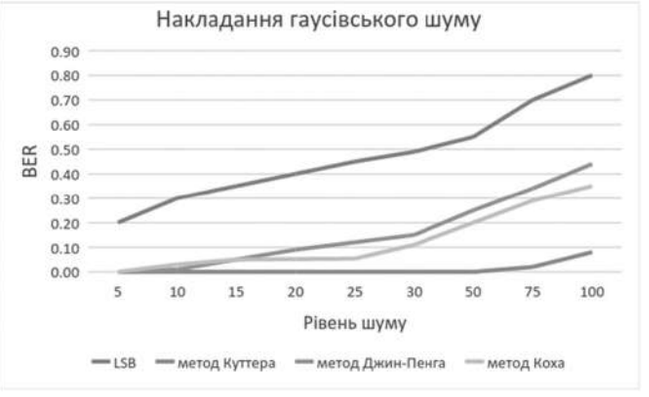


Рисунок 6 – Залежність BER напівтонового зображення від ступеня шуму

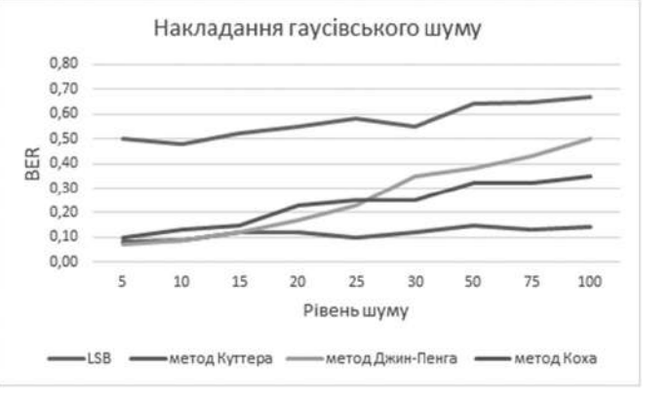


Рисунок 7 – Залежність BER кольорового зображення від ступеня шуму

Через те, що шум представлений одиничними пікселями, він, в першу чергу, впливає на коефіцієнти високочастотних компонентів, проте і низькочастотні компоненти також піддаються сильному впливу, особливо при значному зашумленні.

В ході експериментів зображення-контейнер стискалося в лінійних розмірах до 50% від оригіналу, тобто в 4 рази за кількістю пікселів зображення. Перед зчитуванням ЦВЗ розмір зображення-контейнера відновлювався до розміру оригіналу. Результати експерименту представлені в таблиці 1.

Таблиця 1 – Залежність коефіцієнту BER від масштабування контейнеру

	LSB	***Метод Куттера***	***Метод Коха***	***Метод Джин-Пенга***
Напівтонове зображення	0.72	0.29	0.21	0.46
Кольорове зображення	0.57	0.13	0.17	0.21

ВИСНОВКИ

Велику небезпеку для ЦВЗ, вбудованого в область DWT, являє масштабування зображення-контейнера. При даному виді зовнішнього впливу спостерігається зменшення лінійних розмірів зображення, що може привести до зменшення втрати ЦВЗ в порівнянні з менш масштабованим зображенням, що явно спостерігається для високих рівнів розкладання в низькочастотних площинах.

Таким чином, можна рекомендувати стеганографічні алгоритмів, що використовують DCT та DWT, на основі для використання в стеганосистемах, що забезпечують підвищену стійкість ЦВЗ до впливів частотної області на зображення-контейнер. Перевагу слід віддавати алгоритмам, що використовують низькочастотну площину вейвлет-розкладання і якомога більшого рівня розкладання. Отримані результати стійкості ЦВЗ до зовнішніх впливів на зображення-контейнер підтверджують теоретичну перевагу використання технології косинусного перетворення в стеганосистемах з підвищеними вимогами до стійкості ЦВЗ.

Просторові алгоритми LSB та метод Куттера не слід використовувати при таких деформаціях як JPEG-стиснення або застосування фільтрації, так як отримані показники незадовільні. Проте, метод Куттера показав високі результати при тестуванні на зашумлення та масштабування зображення. При JPEG-перетворенні метод Коха є ненадійним при великих коефіцієнтах стискання.

Стеганографічні алгоритми, що використовують DWT, забезпечують високу стійкість ЦВЗ до JPEG стиснення з втратами. Застосування всього 3-х рівневого розкладання може гарантувати практичну невразливість ЦВЗ до даного виду впливу.

При низькочастотній фільтрації, стійкість ЦВЗ підвищується при збільшенні рівня розкладання, що призводить до перевищення розміру проекції області вбудовування над розміром вікна фільтрів. Фільтри, що підвищують контрастність зображення, призводять до значних втрат ЦВЗ, що вимагає розробки додаткового захисту від даного виду впливів.

СПИСОК ЛІТЕРАТУРИ

1. Krenn. steganography and steganalysis [Електронний ресурс] / Krenn – Режим доступу до ресурсу: http://www.krenn.nl/univ/cry/steg/article.pdf.

2. Neeta. Implementation of LSB Steganography and its [Електронний ресурс] – Режим доступу до ресурсу: http://www.ijana.in/papers/v2i5-11.pdf.

3. Goel. A Review of Comparison Techniques of Image Steganography [Електронний ресурс] – Режим доступу до ресурсу: http://www.iosrjournals.org/iosr-jeee/Papers/Vol6-issue1/G0614148.pdf.

4. Digital Image Steganography Based on Assignment Algorithm and Combination of DCT-IWT [Електронний ресурс] – Режим доступу до ресурсу: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6274358&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F6273236%2F6274306%2F06274358.pdf%3Farnumber%3D6274358.

.NET Core as a powerful instrument for software optimization and simplification of a development process

Prokhorova Keteryna Serhiivna
Full-time student, The National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, Kyiv

ANNOTATION

This article is dedicated to consideration of new features, which are provided by .NET Core software platform. Was revealed that .NET Core isn't just cross-platform framework, but it is also a powerful tool for optimizing and simplifying a development process. A comparison of certain aspects of .NET Core. and .NET Framework software platforms led to revealing of such .NET Core advantages: modularity , manageability, extensibility.

.NET Core, .NET Framework, cross-platform, runtime, standard library, dynamic-link library

.NET Core як потужний інструмент оптимізації програмного забезпечення і спрощення процесу розробки

Прохорова Катерина Сергіївна
студент стаціонарної форми навчання, Національний технічний університет України «Київський політехнічний інститут»
Україна, Київ

АНОТАЦІЯ

Дана стаття присвячена розгляду нових можливостей, що надає програмна платформа .NET Core. Виявлено, що разом з перевагою кросплатформності .NET Core має ряд модернізацій, що роблять її потужним інструментом оптимізації програмного забезпечення і спрощення процесу розробки. Проводиться порівняння певних аспектів програмної платформи .NET Core. і .NET Framework в результаті якого виявлено такі переваги .NET Core: модульність, керованість, розширюваність.

.NET Core, .NET Framework, кросплатформність, середовище виконання, стандартна бібліотека, динамічно приєднувана бібліотека

Історія .NET Framework налічує більше 15 років. За цей період програмна платформа обростала новою функціональністю, додавались нові компоненти і розширювався діапазон мов, що підтримувались. Той факт, що платформа .NET Framework розрахована на роботу під операційними системами сімейства Microsoft Windows зумовив виникнення проектів Potable .Net і Mono метою яких надання кросплатформної втілення .NET Framework. Багатокомпонентність і мультифункціональність платформи .NET Framework є причиною виникнення труднощів при досягненні поставленої мети. Тому й виникають численні проблем з стандартизацією, а користувачі стикаються з обмеженнями у використанні різноманітних аспектів .NET Framework. З огляду на це, можна стверджувати, що є необхідність у представленні цілісного і ефективного рішення, що забезпечить .NET Framework перевагою кросплатформності і відповідність ESMA стандартам.

Рішення було представлено Microsoft Corporation. Наразі розробляється нова платформа, що використовує здобутки .NET Framework, названа .NET Core і вже доступна версія .NET Core 1.0. .NET Core – це кросплатформна реалізація .NET, що має низку суттєвих відмінностей і модернізацій відносно свого попередника.

.NET Core включає середовище виконання CoreCLR, бібліотеку класів CoreFX, що прийшли на заміну Common Language Runtime(CLR) і Base Class Library (BCL). Бібліотека BCL встановлюється централізовано і всі .NET програми сумісно її використовують. .NET Core надає більш гнучкий підхід. По-перше, розробник може самостійно обрати яку версію бібліотеки використовувати. По-друге, завдяки специфічній структурі CoreFX, є можливість підключити окремі бібліотеки у вигляді пакетів за допомогою NuGet. CoreFX побудована як набір бібліотек-компонентів, що розроблювались таким чином, щоб мінімізувати залежності від інших бібліотек. Наприклад, System. Collections тепер зележить лише від System.Runtime, а в .NET Framework ця бібліотека залежить також і від System. Xml. Тобто розробник підключає лише ті бібліотеки, що необхідні. Вони будуть включені як частина додатку і поставлятимуться разом з ним з метою усунення конфліктів з централізовано встановленою версією і труднощів при розгортанні додатку.

Так само як і з бібліотеками, можливість обирати необхідне середовище виконання також присутня. Можна використовувати CoreCLR чи .NET Native для додатків Universal Windows Platform (UWP). Крім того, наразі у процесі розробка нове середовище виконання для .NET Core - CoreRt, що оптимізоване для Ahead-of-Time (AOT) компіляції. Вже випущено бета-реліз - CoreRT for CLI Beta 01-15-2016. Ще одною модернізацією є те, що .NET Core дозволяє розробнику налаштовувати і контролювати середовище виконання. Це спрощує процес написання коду, гарантує безпечне виконання та служить інструментом оптимізації і забезпечення сумісності з машинним кодом.

Разом з середовищем виконання і бібліотекою класів .NET Core включає компілятор, який також можна обирати на розсуд програміста і підключати за допомогою NuGet. Для компіляції в байткод може використовуватись NET Compiler Platform (кодова назва Roslyn). Roslyn написаний на керованому коді, надається у вигляді динамічно приєднуваних бібліотек. .NET Core дозволяє повністю використовувати всі переваги використаного у Roslyn підходу «компілятор як сервіс». Розробнику надається графічний інтерфейс в якому можна переглянути дані проходження процесу компіляції. Це надає додаткові можливості для відлагодження коду. Якщо говорити про компіляцію в машинний код, то наразі .NET Core містить вбудований модернізований just-in-time(JIT) компілятор RyuJIT для компіляція «на льоту». Його задача підвищити швидкодію без погіршення, а може, й покращення якості, отриманогов коду. Крім стратегію JIT в .NET Core доступна також інша стратегія - Ahead of Time(AOT), в цьому напрямку йде робота над середовищем виконання CoreRt і компілятором LLILC. Стратегія AOT означає, що компіляція виконується перед виконанням.

Необхідно зазначити, що є аспекти в яких .NET Core поступається останній версії .NET Framework - .NET Framework 4.6.1. Оскільки .NET Core зовсім молода платформа, не всі можливості, що реалізовані в останній версії .NET Framework доступні в .NET Core. Але розробка ведеться дуже динамічно завдяки наступним обставинам: .NET Core є пріорітетним напрямок для Microsoft, проект має відкритий програмний код і доступній на GitHub, спільнота .NET Foundation бере активну участь у процесі розробки. Варто зауважити таку важливу річ: Microsoft анонсує, що нові версії .NET Core випускатимуться значно частіше ніж релізи .NET Framework. Це означає, що настане момент, коли в .NET Core з`являться елементи не доступні в .NET Framework.

Як перевагу .NET Framework можна розцінювати те, що належність платформи до Windows компонентів забезпечує їй обслуговування і оновлення засобами Windows Updates. Підхід, що застосовує .NET Core, вбачає включення певних компонентів платформи безпосередньо в розроблювані додатки. Таким чином, кожен додаток має свій власний набір різноманітних компонентів. Тому, вони повинні обслуговуватись через NuGet, а не через оновлення операційної системи.

Наприкінці наведемо конкретні рекомендації щодо випадків, коли доцільно використання .NET Core,

а коли .NET Framework. Для створення десктопних додатків для Windows альтернативи відсутні – зараз і надалі використовуватиметься .NET Framework. .NET Core варто використовувати для написання ASP.NET, консольних та UWP додатків, а також бібліотек та фреймворків. У всіх інших випадках використовується на сьогоднішній день .NET Framework. Наділі спектр використання .NET Core буде розширюватись, а щодо програмного забезпечення, що не може бути написане на платформі .NET Core сьогодні, можна застосовувати наступний принцип: використовувати сервіси написані на .NET Core в сценаріях програм. Корисно знати, що у деяких випадках код, написаний використовуючи .NET Framework, може бути без змін перенесений на .NET Core. Для перевірки цього можна використати .NET Portability Analyzer.

Виходячи з вищенаписаного можна підсумувати, що разом з кросплатформністю .NET Core надає ряд інших переваг, а саме: модульність, керованість та розширюваність. Такі ґрунтовні якісні зміни, що привносить .NET Core порівняно з .NET Framework, надають широкий спектр можливостей для написання ефективних і кросплатформних програмних продуктів, але разом з цим, від розробника вимагається вищий рівень професіоналізму і обізнаності, з метою отримання ним можливості використання повного спектру нововведень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Overview of .NET Ecosystem in 2015 [Електронний ресурс] // Режим доступу: https://dotnet.github.io/about/overview.html

2. .NET Framework [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/.NET_Framework

3. Mono [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/Mono

4. Portable .NET Framework [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/Portable.NET

5. .NET Core and Open-Source [Електронний ресурс] // Режим доступу: https://msdn.microsoft.com/en-us/library/dn878908(v=vs.110).aspx

6. Introducing .NET Core [Електронний ресурс] // Режим доступу: http://docs.asp.net/en/latest/conceptual-overview/dotnetcore.html

7. Introducing .NET Core [Електронний ресурс] // Режим доступу: https://blogs.msdn.microsoft.com/dotnet/2014/12/04/introducing-net-core/

8. RyuJIT .NET JIT compiler CTP1 FAQ [Електронний ресурс] // Режим доступу: https://blogs.msdn.microsoft.com/dotnet/2013/11/18/ryujit-net-jit-compiler-ctp1-faq/

9. .NET Compiler Platform [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/.NET_Compiler_Platform

Рецензент: к.т.н., доц. каф. АСОІУ НТУУ «КПІ» Жданова О. Г.

Lookup tables for increasing computing speed in computer graphics and their prospects in the future

Goncharenko O. R.
student of the NTUU «KPI»
Kyiv, Ukraine

This article describes look-up tables, whose use of is one of the many methods of optimization, increasing productivity and performance in actual computing. Examples of the usage of the most common lookup tables are demonstrated in this article. Discussed prospects of the lookup tables in the future.

Key words: lookup table, LUT, memristor.

Таблиці пошуку для підвищення швидкості обчислень у комп'ютерній графіці та їх перспективи у майбутньому

Гончаренко О. Р.
студент НТУУ «КПІ»
Київ, Україна

В докладі розглянуто таблиці пошуку, використання яких є одним з багатьох методів оптимізації, підвищення продуктивності та швидкодії в сучасних обчисленнях. Демонструються приклади використання деяких таблиць пошуку. Розглянуто перспективи таблиць пошуку у майбутньому.

Таблиці пошуку (англ. lookup table, LUT) – це структура даних, яка використовується для заміни обчислень на операції пошуку. Якщо отримання даних з таблиці пошуку буде швидшим ніж обчислення результатів з нуля, то використання таблиці дасть значний приріст в продуктивності [1]. Для таблиці обчислюються найбільш поширені вхідні данні. Для запитів, які потрапляють між прикладами з таблиці, алгоритм інтерполяції може генерувати прийнятні наближенні значення шляхом усереднення найближчих значень.

	1	2	3	4
1	*f (1.1)*	*f (1.2)*	*f (1.3)*	*f (1.4)*
2	*f (2.1)*	*f (2.2)*	*f (2.3)*	*f (2.4)*
3	*f (3.1)*	*f (3.2)*	*f (3.2)*	*f (3.4)*

Рис. 1. Приклад таблиці пошуку

На рис. 1 зображено приклад таблиці пошуку. Кожна точка набору даних є показником вхідних значень конкретної величини таблиці пошуку. Масив даних таблиці служить в якості дискретного представлення функції, обчисленої в певній точці, яка є індексом даних [2].

Проте, використання таблиць пошуку у простих обчисленнях може призвести до погіршення роботи. Час запиту ресурсів з пам›яті і складність пам›яті можуть підвищити час виконання програми і складність системи. Можливість росту кеш-пам›яті також може стати проблемою. Запит даних у великих таблицях може призвести до промахів кешу. У деяких мовах програмування (наприклад, Java), звернення до таблиці пошуку може бути навіть більш «дорогим» через обов›язкові перевірки кордонів, що включає в себе додаткові порівняння та розгалуження для кожної операції пошуку.

Розглянемо приклад використання таблиць пошуку у алгоритмі пошуку схожих зображень. Маємо систему з великою кількістю зображень і на вхід надходить команда «Знайти схожі зображення». Для цього потрібно проаналізувати кожне зображення, зменшити його (зазвичай роблять сітку 8x8), прибрати колір (привести зображення до градацій сірого), створити ланцюг біт, беручи 0 або 1 в залежності від середнього значення кольору в певній точці, перевести отримане значення до простого хешу, отримати таким самим чином хеш вхідного зображення, та на кінець порівняти кількість різних бітів. Час отримання результату пошуку буде помітно зростати з кожним новим зображенням. Але, якщо, мати таблицю пошуку з попередньо підрахованими значеннями хешу для кожного зображення, то швидкість формування масиву відповіді значно збільшитися. Окрім швидкого пошуку схожих зображень, можливо пришвидшити трансформацію зображення, а саме за рахунок таблиць пошуку оптимізувати приведення зображення до чорно-білого спектру. Потім розглянемо це більш детально.

Таблиці пошуку також широко використовуються в перетворенні (трансформації) зображень. В залежності від складності перетворення збільшення швидкості обчислень може бути здобуте саме за рахунок таблиць пошуку, аж до 15 разів або навіть більше для логарифмічних чи експоненціальних перетворень [3].

Розглянемо ефект найбільш поширених таблиць пошуку. Таблиця пошуку квадратної степені зменшує загальну яскравість зображення, але збільшує контраст, роблячи темні зони зображення більш темними, світлі – яскравішими. З іншого боку таблиця пошуку квадратного кореня, збільшує загальну яскравість, але як тільки ефект стає менш вираженим у яскравих зонах, тоді у темних – зменшується контраст. S-подібна Гауссова або сигмоїдна таблиця пошуку надає світлим і темних зонам гомогенності але збільшує контраст у зонах з середнім освітленням. В додаток до прискорення, набуте за рахунок використання таблиць пошуку, вони також дозволяють виконувати перетворення відтінків сірого, які не можуть бути реалізовані легким шляхом.

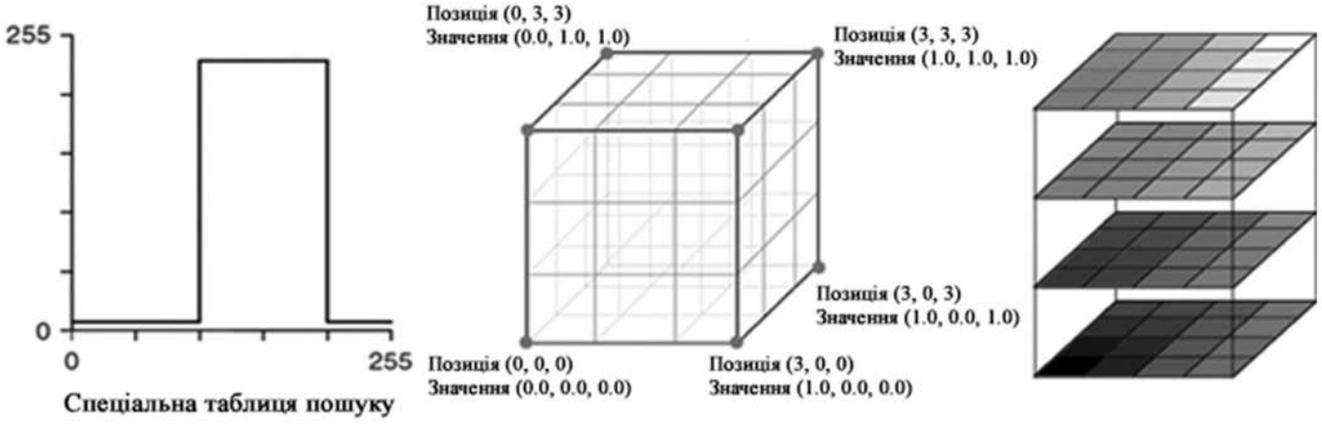


Рис. *2.* Спеціально визначена таблиця пошуку та тривимірна таблиця пошуку

Рис. 2 показує спеціально визначену таблицю пошуку. Ця таблиця (рис. 2) повертає чорний колір для дуже яскравих та дуже темних зон, а для зон з середнім освітленням – білий. Таким чином палітру можна зменшити до двох рівнів сірого.

Якщо подивитися на це з математичної точки зору, то використання таблиць-пошуку для перетворень у границях сірого, означає, що деяка функція трансформування *f(x)* надана у вигляді таблиці, яка містить одну допоміжну точку функції для кожного можливого значення сірого кольору. Іншими словами таблиця містить певне значення для кожного значення відтінку сірого. Кольорові зображення також можуть бути трансформовані за допомогою таблиць пошуку, але для цього треба буде мати таблицю пошуку для кожного кольорового каналу і така таблиця пошуку називається тривимірною (3D). Прискорення відбувається за рахунок того, що не потрібно кожен раз обчислювати значення функції для певного параметру, це робиться один раз при першому запуску додатку.

Тривимірні таблиці пошуку індексовані за трьома незалежними параметрами, як показано на рис. 2 [4].

Тоді як одновимірна таблиця містить тільки 4 елементи в 4-х однакових місцях по кожній осі, відповідна трьох вимірна таблиця пошуку містить 4^3, що дорівнює 64-м елементам. Тому розмірність 3D таблиці пошуку зростає з лінійною частотою дискретизації. Для того щоб помістити в пам'ять 32x32x32 таблицю треба 393 КБ, 256x256x256 таблиця вимагає 200 МБ. Навіть якщо GPU має стільки доступних ресурсів, великі трьох-вимірні таблиці пошуку можуть швидко заповнити весь кеш текстурування, що може погіршити продуктивність. 3D таблиці пошуку є незамінною складовою у системах реального часу, задача яких - відтворювати потокове відео. Саме за рахунок таблиць пошуку відео може бути оптимізоване для перегляду без значного зниження fps шляхом використання певних фільтрів які корегують контраст, яскравість, тон та інші параметри зображення [5].

Розглянемо приклад лінійної кольорокорекції, зміни контрасту за допомогою лінійного розтягнення (як авто-контраст у Photoshop). Корекція – до зображення застосовується перетворення яскравості, компенсуючи небажаний ефект:

$$f^{-1}(y) = x$$

де y – яскравість пікселя на вхідному зображенні, x – яскравість пікселя після корекції.

Компенсація вузького діапазону яскравості – лінійне розтягнення:

$$f^{-1}(y) = (y - y_{\min}) \cdot \frac{255}{y_{\max} - y_{\min}}$$

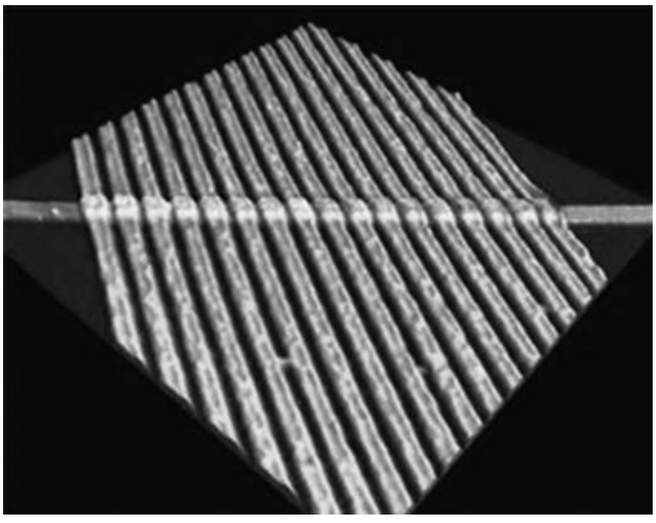


Рис. 3. 17 мемристорів

Для потокового відео обчислення данної формули для кожного пікселя буде деградувати fps. Щоб уникнути цього треба обчислювати нові таблиці пошуку для всього кольорового діапазону одразу після зостосування нових налаштувань контрасту, що також може деградувати продуктивність за рахунок переповнення кеш-пам'яті, бо окрім цієї тривімірної таблиці потрібно зберігати й інші.

За підрахунками NVidia [4] таблиці пошуку підвищують продуктивність більше ніж у 100 разів і це не є межею. Недоліки, пов'язані з розміром таблиць пошуку, можуть бути усунені в майбутньому з використанням комп'ютерів нового покоління на базі мемристорів. Мемристор (рис. 3) – це пасивний елемент в мікроелектроніці, здатний змінювати свій опір в залежності від заряду, що протікав в ньому. Основна перевага цього елементу в його енергонезалежності та надшвидкій передачі даних [6].

У даний час компанія Hewlett-Packard розробляє перший у світі суперкомп'ютер на базі мемристорів. Проекту дали кодову назву «The Machine». Комерціалізація технології очікується у 2020-х [7].

LUT (0,0)	LUT (0,1)	LUT (0,2)
LUT (1,0)	LUT (1,1)	LUT (1,2)
LUT (2,0)	LUT (2,1)	LUT (2,2)

Рис. 4. Можливий варіант таблиці пошуку у системі на базі мемристорів

Саме збільшення швидкості запису та зчитування відкриє таблицям пошуку нові можливості. Відтоді якщо у комп'ютера не буде вистачати ресурсів на зберігання таблиці пошуку у кеш-пам'яті, стане актуальним зберігати її частини у базах даних, так-як проблема зі швидкістю зчитування інформації з БД або файлу зникне. Таким чином буде можливо оптимізувати такі задачі як факторизація для великих діапазонів, що не актуально зараз через великий об'єм таблиць пошуку з факторами чисел, поліномів або матриць.

Наприклад, на рис. 4 ви можете побачити приблизну структуру такої таблиці пошуку для комп'ютерів на базі мемристорів. Структура таблиці пошуку проста: масив 3x3, кожен елемент якого містить в собі посилання на інші масиви (таблиці пошуку), які занадто великі щоб зберігати їх усі в кеш-пам'яті. Тому під час запиту до певного елементу буде завантажуватись до кешу певна область, яка необхідна для обчислень в даний момент (як LUT (0,2) замінює LUT (1,1) на рис. 4).

ЗАКЛЮЧЕННЯ

В даній роботі було розглянуто таблиці пошуку, що використовуються для підвищення ефективності обчислень та приклади їх використання, навелено приклади поширених таблиць пошуку та їх ефект на зображення, описано формулу лінійної кольорокорекції та формулу зміни контрнасту за допомогою лінійного розгортання. Запропоноване можливе рішення для усунення деяких недоліків таблиць пошуку за допомогою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Automated Memorization in C++ [Електронний ресурс] – Режим доступу до ресурсу: http://pmcnamee.net/c++-memoization.html

2. Anatomy of a Lookup Table [Електронний ресурс] – Режим доступу до ресурсу: http://www.mathworks.com/help/simulink/ug/anatomy-of-a-lookup-table.html

3. Christian Demant. Industrial Image Processing: Visual Quality Control in Manufacturing / Christian Demant, Bernd Streicher-Abel, Carsten Garnica, 1999. – 353 c.

4. Matt Pharr. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation / Matt Pharr, 2005. – 814 c.

5. Benny Bing. Next-Generation Video Coding and Streaming / Benny Bing, 2015. – 344 c.

6. Гончаренко О. Р. Підходи до побудови пам'яті обчислювальних систем на основі мемристорів / О. Р. Гончаренко, А. М. Волокита // Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем / Дніпропетровськ, 2015. – С. 129-131

7. The Machine: A new kind of computer [Електронний ресурс] – Режим доступу до ресурсу: http://www.labs.hpe.com/research/themachine/

Рецензент: к.т.н., доц. каф. ОТ НТУУ "КПІ" Волокита А. М.

Authentication system increased reliability

Valery P.Simonenko
professor, Sc.D
Ukraine, Kiev

T.Dregalo
postgraduate student
Ukraine, Kiev

The paper deals with modern authentication systems. Author has paid much attention to method of authentication with a use of login and password. He has also explained all laws and paradigms that are related to authentication systems.

The author has proposed a new authentication system, in which he proposes to use an account. The main idea is to use as account login and password, but not real. Users do not know real login, but they know only password. The proposed system has a 3 level architecture and each level has special functions. Result of any action of system saves in database.

Password, user, authentication system, operation system, account.

Система аутентифікації підвищеної надійності

Сімоненко В.П.
д. т. н., професор
Україна, Київ

Дрегало Т.В.
аспірант
Україна, Київ

В даній статі розглядаються сучасні системи аутентифікації. Автор вибрав метод аутентифікації за допомогою паролю та логіну. Автор також поясним всі принципи роботи та парадигми котрі пов'язані з системами аутентифікації.

Автор запропонував нову систему аутентифікації, в котрій він використовує поняття акаунту. Головна ідея використовувати логін і пароль від акаунту, але не використовувати реальний. Користувачі системи не будуть знати реальні свої логіни, але

вони будуть знати лише паролі. Архітектура система складається з 3х рівнів, а кожний рівень має свій функціонал. Результати кожної дії в системі зберігаються в базі даних.

Пароль, логін, системи аутентифікації, операційні системи, аккаунт.

ВСТУП

Питання безпеки було досить актуально завжди. Раніше до даного терміну відносили лише поняття безпеки людей, але в час інформаційних технологій даний термін використовується для захисту інформації. Безпека інформаційних технологій включає в себе набір для захисту та атак. В поняттях безпеки виділяють три основні поняття[1]:

- конфіденційність;
- цілість;
- надійність системи.

Конфенційність є один із найважливіших понять безпеки, основною задачею котрої є надання доступу користувачам до ресурсів до котрих вони мають доступ. В сучасних інформаційних системах конфіденційність розділена на дві підсистеми, а саме на систему ідентифікація та його авторизації.

ПОСТАНОВКА ЗАДАЧІ

Задача полягає в аналізі існуючих способів аутентифікації, вибрати найбільш поширену систему аутентифікації. Визначити основні недоліки вибраної системи аутентифікації та запропонувати свою систему аутентифікації.

ТЕРМІНОЛОГІЯ

Аккаунт – це об’єкт, котрий використовується в запропонованій системі, який складається з логіну, паролю та системних даних, котрі необхідні для роботи системи.

User Level – це перший рівень архітектури, в котрому представлений функціонал користувачів.

SBSL – це другий рівень архітектури, даний рівень представлений системними функціями, підсистемою аудитом та базами даних.

Admin Level – це третій рівень архітектури, за допомогою заданого рівня відбувається управління системою.

Db – це реляційна база даних.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Найголовнішою підсистемою в інформаційній системі є система перевірки наявність користувача в ній, а після надання йому необхідних прав. В якості аутентифікації можна використовувати цифрові ключі, комбінації логіну та паролю, біометричні данні такі, як голос або відбитки пальців або апаратні способі.

Проте не дивлячись на велику кількість способів аутентифікації в 95% [2] випадків використовують комбінацію логіну та пароля, так як інші способи потребують додаткових апаратних пристроїв. Основною проблемою використання даного способу аутентифікації [4] є те, ще користувачі зазвичай використовують прості паролі. Наприклад це можуть бути паролі, типу ім’я та дата народження, ім’я місто рік народження та інші. Проте є ряд інших проблем до яких можна віднести не досконаліть існуючих механізмів захисту [1] . Існуючі механізми захисту не досить досконалі, наприклад механізм блокування та ігнорування запитів від одного користувача з певного IP адресу на певний час, в такому блокування проходить лише на IP адрес і на певний час, коли час блокування пройде користувач знову зможе надсилати запити.

Для прикладу простий пароль, котрий складається лише з 5 цифр, можна підібрати за 100 тисячі спроб. Отже, щоб підібрати такий пароль потрібно необхідно 100 000 / 3 =33,001 спроб (для прикладу було вибрано 3 спроби для аутентифікації після система блокує запити з заданого адресу, в якості було вибрано програму login [3], яка використовується в операційних системах сімействах Linux). Отже нам необхідно 33 тисяч спроб, якщо дані запити робити з різних IP адресів, наприклад 100, то за 333 такти ми підберемо пароль. Проте даний приклад досить простий та вибраний для демонстрації, але він наглядно демонструє недосконаліть системи аутентифікації, питання постає лише в складності пароля та кількості IP адресів з котрих будуть приходити запити, отже виходить:

$N = \frac{Nv}{Nip} n$, де N - кількість спроб за котрі можна підібрати пароль, Nv – максимальна можлива кількість паролів залежить від довжини пароля та кількості можливих символів, котрі використовуються в паролях, Nip кількість IP адресів з котрих будуть надходити запити, а Nt кількість спроб на аутентифікації. Отже, можна отримати час за котрий можна підібрати пароль отримаємо:

$\tau = \frac{N * \tau d}{Nt}$, де τ час в хв. за котрий буде підібраний пароль, τd час в хв. блокування після невдалої аутентифікації.

За даними формулами ми зможемо порахувати, час котрий необхідний для підбору [2] 6 значного цифрового пароля, кількість спроб на аутентифікації, а час блокування 5 хвилин. Результати в таблиці 1 та на рисунок 1.

Таблиця 1. Демонстрація необхідний часу для підбору простого 6 значного пароля.

№	Nv	Nt	Nip	N	Delay (хв)	T (хв)
1	1000000	3	1	333334	5	555555,6
2	1000000	3	50	6667	5	11111,11
3	1000000	3	100	3333	5	5555,556
4	1000000	3	150	2222	5	3703,704
5	1000000	3	200	1667	5	2777,778
6	1000000	3	250	1334	5	2222,222
7	1000000	3	300	1111	5	1851,852
8	1000000	3	350	953	5	1587,302

9	1000000	3	400	834	5	1388,889
10	1000000	3	450	741	5	1234,568
11	1000000	3	500	667	5	1111,111
12	1000000	3	550	607	5	1010,101
56	1000000	3	2750	121	5	202,0202
57	1000000	3	2800	119	5	198,4127
58	1000000	3	2850	116	5	194,9318
58	1000000	3	2900	114	5	191,5709

Маючи 2900 IP адресів пароль можливо підібрати пароль за 191 хвилини, отже отримаємо графік зображений на рисунку 1. Дивлячись на даний графік з певністю можна сказати, зо даний спосіб аутентифікації на пряму залежить від складності пароля та кількості IP адресів з котрих приходять запити. В цілому графік на рисунку 1 не зміниться для пароля довільної складності. Дивлячись на графік можна сказати, що пароль довільної складності не дасть повного захисту, все опирається лише в час необхідний для підбору.

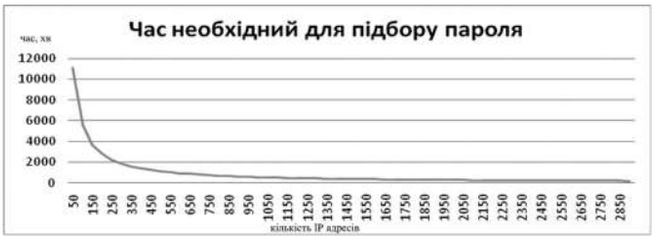


Рисунок 1. Показує час необхідний для підбору пароля в залежності від кількості IP адресів.

Системи аутентифікації мають ряд проблем, для вдосконалення її необхідно розбити на декілька автономних рівнів. В першу чергу з управляючого та системного, а також окремо виділити рівень користувачів та аудит. Більш детально зображено на рисунку 2.

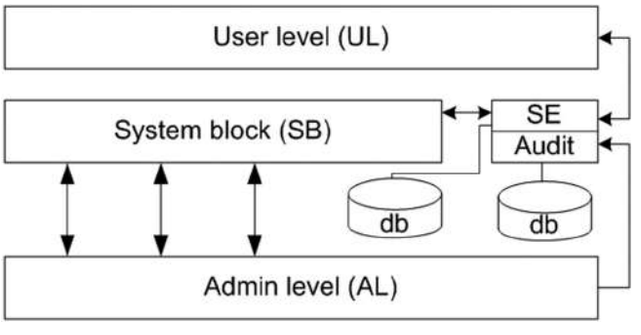
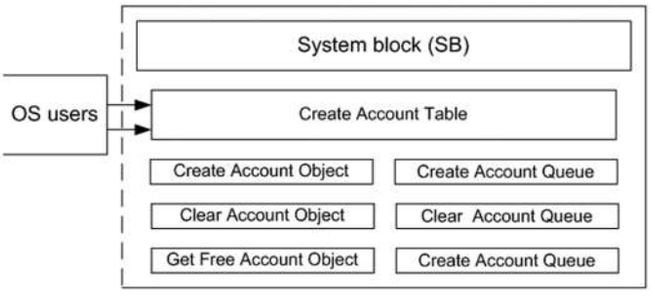


Рисунок 2. Схема удосконаленої системи аутентифікації.

Рівень користувача представляє собою об'єкт котрий складається з системних інформації та комбінації логіна та пароля. Рівень SBSL складається з двох блоків, а саме з SE та SB. Далі даний об'єкт передається в SE блок, котрий в першу чергу записує запис в базу через підсистему Audit, далі SE блок приймає рішення чи обробляти запити чи ні. Якщо SB прийняв рішення обробляти він передає об'єкт до SB блоку, в іншому випадку система створює Thread та робить йому sleep на певний час, а після закінчення додає запис через блок Audit та повертає значення UL.

На SB блок випадає найбільша завантаженість, даний блок використовує для заміни реального логіну та пароля. Основні функції даного блока зображені на рисунку 3.

Коли користувач робить спробу аутентифікації в систему, спочатку перевіряється чи існує аккаунт з його логіком чи ні, якщо не існує – тоді створюється аккаунт та встановляються початкові параметри, не має різниці чи існує такий логін в системі чи ні. Після створення аккаунту створюється черга в котру додається даний запит. Якщо аккаунт уже існує, тоді запит додається в чергу та чекає свого опрацювання. У кожного аккаунту є поле час відклику, початковий час відклику 0 секунд, з кожною невдалої аутентифікації час відклику буде збільшуватись за певним законом. Кожний наступний запит буде оброблятись за час, який рівний часу відклику аккаунту. Також можна додати функцію автоматичної генерації нового логіну зі збереженням старого пароля, після декількох невдалих спроб аутентифікації, та автоматичної відправки його за допомогою повідомлення користувача. Результат запиту передаються в блок SE, де підсистема Audit зберігає результати.



Рисунку 3. Основні функції SB блока.

У SB блока є багато функцій котрі він сам не може використовувати, до них відносять очистка аккаунту, очистка черги аккаунту, отримання вільного аккаунту та інші, дані функції необхідні для AL. Данні функції будуть доступні для управління системи зовні, наприклад, якщо необхідно очисти чергу чи змінити логін від аккаунту. В свою чергу AL рівень необхідний для правління SBSL. До основних функцій даного рівня відносять заміну логіна чи пароля, довільні маніпуляції над аккаунтами або зміна налаштування інших рівнів. Виклик довільної функції з AL одночасно проходить по двом блокам, виклик самої функції в SB та запис дії підсистемою Audit.

ВИСНОВКИ

В даній роботі розглянуто сучасні системи аутентифікації, способи та механізми аутентифікації. Зроблено аналіз та знайдено недоліки в існуючих системах, а на основі чого було запропоновану систему аутентифікації на основі поняття аккаунту. Основою котрої є пере направлення реальних паролів та логінів на аккаунти, а аккаунти в свою чергу мають свої логіни та паролі. Основною відміною від сучасних систем є те, що запропонована система захищає в основному логіни користувачів, а не паролі. В випадку,

якщо аккаунт заблокований користувач отримує новий логін, а пароль залишається старим, що в свою чергу в рази збільшує конфіденційність та надійність системи. Ще одною головною перевагою є наявність функції аудиту, результат кожної дії в системі зберігається в базі даних.

ЛІТЕРАТУРА

1. Authentication in an Internet Banking Environment [Електронний ресурс] // Federal Financial Institutions Examination Council. – 2010. – Режим доступу до ресурсу: https://www.ffiec.gov/pdf/authentication_guidance.pdf.

2. LOGIN(1) [Електронний ресурс]. – 2012. – Режим доступу до ресурсу: http://man7.org/linux/man-pages/man1/login.1.html.

3. Таненбаум Э. С. Современные операционные системы. 4-е изд / Э. С. Таненбаум, Х. Бос. – Санкт-Петербург: Питер, 2016. – 1120 с.

4. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – Москва: Триумф, 2002. – 816 с. – (3000).

ОБРОБЛЕННЯ ІНФОРМАЦІЇ В СКЛАДНИХ СИСТЕМАХ

УДК 004.622:004.822

Formation of subject area and the co-authors network by sounding of Google Scholar Citations service

D.V. Lande
Institution for Information Recording of National Academy of Science of Ukraine
Ukraine, Kyiv
dwlande@gmail.com,

V. B. Andrushchenko
Institution for Information Recording of National Academy of Science of Ukraine
Ukraine, Kyiv
valentyna.andrushchenko@gmail.com

The suggested methodic is the way of formatting the subject areas models and co-authors networks by sounding the content networks. The paper represents the notion networks which match tags and authors of Google Scholar Citations service. Models depicted in the work were built for the physical optics area, and it can be applied for other domains. The proposed ways of defining connections between science areas and authors depicts the collaborations opportunities and versatility of interdisciplinary.

Keywords: subject domain, co-authorship network, legal science, sensing of a network, information network, physical optics, text mining

Today science and technique progress depends heavily on the way scientists can establish the right cooperation process and create the successful collaboration. And activity in every science field assumes not only experiments and their depiction in scholar papers, but the repercussion of scientific results. And first of all - the way the paper is cited, when, whom and in what way. The information scientometric databases allow researchers trace their activity and fix their achievements, and this data is actual not only for the scientific image but for the way of promoting their research all over the world and even for career.

New information resources give new opportunities for describing the subject areas and studying the consistent pattern of the scientific intercommunications.

One of the main instruments of investigating the regularity of scientific cooperation - is the co-authorship network forming by scientometrics services [2].

Co-authorship network permits to:

- obtain scientometric indexes;
- find experts to solve complex problems;
- define the participants for the expertise procedure;
- search out colleagues for the collaboration formation;
- discover interconnections between scholars for further scientometric analysis;
- provide the estimations for the grant effectiveness and the performance of research institution;
- etc.

There are some major services of scientific information, which give an opportunity to get the scientometric data, create users profiles and can also contain bibliographic information, these databases are Web of Science and Scopus. But the access to the resources of these services is paid and it complicates the work with them.

One of the main services with the free access is Google Scholar, which allows creation of profile containing information about all the published materials of the researcher, and also has the powerful search engine.

Authors experienced the approach to form the subject domain model and create the co-authors network (‘text mining’ [3] ‘legal science’ [5], ‘physical optics’) by sounding big information network and creating the notion network, matching with tags of sceintometric service Google Scholar Citations (http://scholar.google.com/citations) [3], [4].

The interface of the Google Scholar Citations gives the authors lists and the complemented tags (notions and concepts), which are corresponded with the preassigned tag.

The work also presents the algorithm of co-authors network formation - the model of researchers cooperation by sounding mentioned scientometric network.

The proposed sequence of operations according to the algorithm [1]::

- The short list of the base tags, defined in an expert way.
- One tag is chosen from the list.
- The performed search represents the web-page corresponding with the chosen tag.
- The neighboring tags, comprised in the page, are added to the forming network.
- From the neighboring tags are choses the ones, pages of which are planned to be proceed for the further analysis. This tag is the one with the highest rate, which responds the theme of the chosen subject area, and the transition to which hasn’t been executed.
- If such a tag been chosen the jump to the step 3 is to be provided.
- If there is no such a tag, and the list of base tags is not fulfilled, then the segue to the next base tag from the initial tag list is provided (Step2). Otherwise the network is considered to be built.

The described algorithm was complemented by the rule, which reads that the Steps 3-6 were provided so many times as expert set. In example with the ‘physical optics’ tag it makes up 5.

Figure 1 shows the example of the subject area notion network, built according to the algorithm on base tags. Figure 2 - the enlarged part of the network.

Figure 1. The structure of the network according to the base tag

By scanning the Google Scholar Citations there was obtained the network with such parameters:

- the nods number - 401;
- the edges number - 670;
- the average node rate - 1,67;
- the diameter - 6;
- the average shortest way - 2, 48.

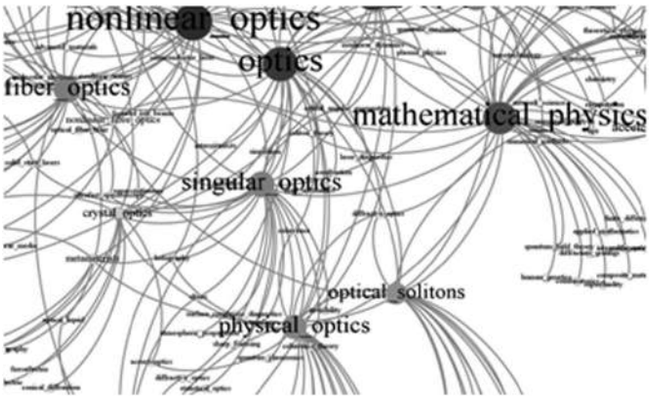


Figure 2. The part of the notion network

The given principle was adopted to the formation of co-authorship network [3]:

- The base tag is expertly defined (as an example - ‘physical optics’) and there are several further steps:
- The page, corresponded to this page opens.
- The most cited author, represented on the page is chosen.
- All the authors from the chosen author’s page are added to the framing network. The edges-connections to these nodes (co-authors) form the initial nod (author).
- From the list of nods the one is chosen, transition on which page is planned for the further analysis. This is the most weighty nod responded the theme of chosen subject domain (its tags contain the parts of words, singled out by experts, ‘physic’ and ‘optics’ in particular) and it is not the one from the nods, which pages were observed.
- If there is no such an author the network is considered to be completed.

In appliance with the algorithm the co-authors network was built taking into the consideration the restrictions for the number of scanning nodes. The route of the algorithm, the list of researchers and their tags are depicted on the figure 3.

As the result of the scientometric network sounding there was obtained the co-authors network with the next parameters:

- the nods number - 207;
- the edges number - 384;
- the average node rate - 2,34;
- the diameter - 7;
- the average shortest way - 4,23.

Tiberiu Tudor physical_optics polarization coherence lasers quantum_optics
Sabino Chavez-Cerda optics mathematical_physics physical_optics diffractve_optics optical_solitons
David Snchez-de-la-Llave optics physical_optics fourier_optics_and_signal_processing holography
Miguel A. Bandres physics optics photonics
Johannes Courtial physics optics ray_optics holography
Mark R Dennis mathematical_physics optics singular_optics topology
Franco Nori condensed_matter_physics quantum_optics quantum_information physics superconductivity
Gran Johansson quantum_physics quantum_computing microwave_quantum_optics the_dynamical_casimir_effect mesoscopic_superconductivity
Abraham G. Kofman quantum_physics quantum_information quantum_optics laser_physics solid_state_qubits
Skab Ihor physical_optics singular_optics crystal_optics piezo_and_electrooptics acoustooptics
Eduard Carcol'' physical_optics seismology computers
Neill Lambert physics quantum_optics quantum_computing nano_mechanics quantum_mechanics
Arend G. Dijkstra theoretical_chemical_physics nonlinear_optics open_quantum_systems
B. M. Rodrguez-Lara quantum_optics optical_physics
Suren A. Chilingaryan quantum_optics_and_quantum_information quantum_physics quantum_mechanics
Myun-Sik Kim metrology interferometry physical_optics phase_anomaly microlens
G. Rodriguez Zurita physical_optics interferometry fourier_optics
Vlokh Rostyslav physical_optics
Karol Bartkiewicz quantum_physics quantum_optics quantum_information
Anirban Pathak physics quantum_information quantum_optics
Swapan Mandal quantum_optics laser_spectroscopy quantum_information_theory mathematical_physics
Ioannis Besieris stochastic_linear_and_nonlinear_wave_propagation phase_space_techniques wave_localization

Figure 3. The route of forming the co-authors network

Using the Gephi application the visualization of the network became available (Figure 4). Application of the clustering analysis makes the detection of the closest researchers-co-authors groups, scientific schools, and experts groups approachable.

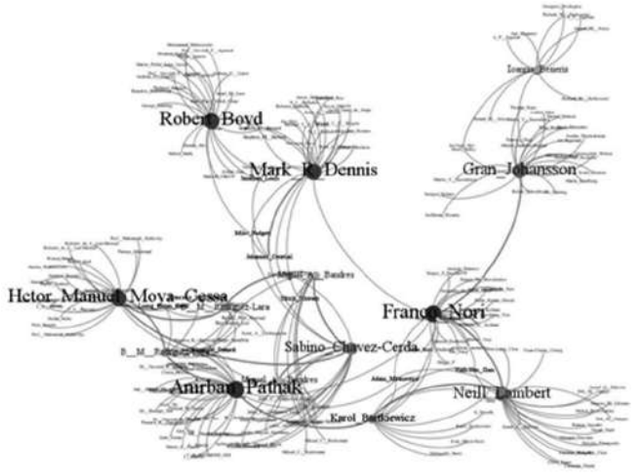


FIgure 4. The structure of the co-authors network.

If we leave only structurally weighty nodes and edges, using Gephi, the clustering of the initial network can be reached, and also the most connected subgroups of researchers (Figure 5).

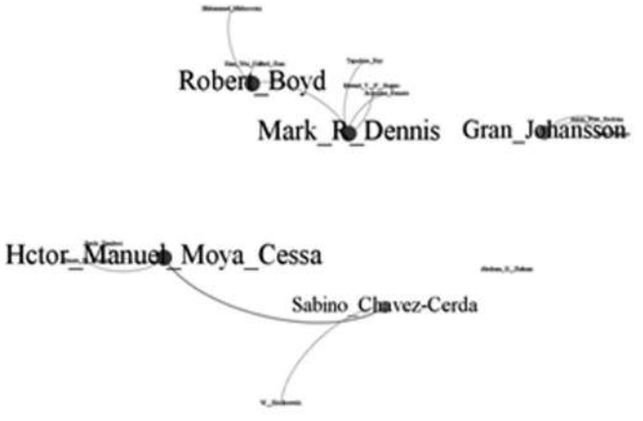


Figure 5. The biggest clusters of the represented network.

Thus the suggested attempt to form the model of subject domain and the co-authors network in frames of the defined subject area, the measured elements of which are knowledge markers (tags), preassigned by researchers-participants of Google Scholar Citations project. It's necessary to notice the fundamental difference of the represented model of automatic subject domain model forming from the existed ones, based on the text corpuses analysis or the direct participation of experts in process of electing nodes and edges. This way the expert-user inputs only the grains of the knowledge as tags and small dictionaries (up to 10 words). Hereinafter program uses information provided by the authors of publications and tags noted as the main ones.

The work is the actual application for the information scientometric databases. It clearly widens the existed facilities and represents the great amount of analytical information vital not only for researchers but for the research institutions as a way of monitoring the dynamics of scholars activity and cooperation, and also can appear the recommended instrument in founding science policy of the country.

REFERENCES

1. Lande D.V. A Domain Model Created on the Basis of Google Scholar Citations // CEUR Workshop Proceedings (ceur-ws.org). Vol-1536 urn:nbn:de:0074-1536-8. Selected Papers of the XVII International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2015) Obninsk, Russia, October 13-16, 2015. – pp. 57-61.

2. Liu J., Li Y., Ruan Z., Fu G., Chen X., Sadiq, Deng Y. A new method to construct co-author networks // Phisica A. – 2015. – 419. – pp. 29-39.

3. Lande D.V., Balagura I.V., Andrushchenko V. B. Building a network of co-authorship according to Google Scholar Citations service: materials VI intern. scientific-technical. conf. ["Open semantic technologies of design of intellectual systems" (OSTIS-2016)], (Minsk, 18-20 feb. 2016). – Minsk: BSUIR, 2016. – pp. 233-237. (Russian)

4. Brezina V. Use of Google Scholar in corpus-driven EAP research // Journal of English for Academic Purposes.– 2012. – 11. – P. 319-331.

5. Lande D.V., Andrushchenko V. B. Building of co-authorship networks of the law according to Google Scholar Citations service // Information and Law. - 2016(3). – pp. 146-150. (Ukrainian)

Research of classification problem of large volumes of medical and statistical data based on the importance sampling

Oleksandr A. Galkin
National University of Kyiv, the Faculty of Cybernetics
Ukraine, Kyiv

Taras Shevchenko
National University of Kyiv, the Faculty of Cybernetics
Ukraine, Kyiv

This paper is devoted to the analysis and processing of real data related to the professional functioning of medical personnel in the performance of duties in patient care. The aim of our study of large amounts of medical and statistical data is to determine the optimal functioning of health professionals using the supervised learning algorithms. We use the method that makes it possible to modify the set of time features of sectors into prior probability of functioning sectors by using the concept of importance sampling.

Key Words: big data, Bayesian network, time window

Дослідження задачі класифікації великих масивів медико-статистичних даних на основі вибірки за значимістю

Галкін Олександр Анатолійович
Кандидат фізико-математичних наук, асистент кафедри математичної інформатики факультету кібернетики Київського національного університету імені Тараса Шевченка
Україна, Київ

Стаття присвячена аналізу та обробці реальних даних, що пов'язані з професійною діяльністю медичних працівників при виконанні обов'язків по догляду за пацієнтами. Запропоновано метод визначення оптимального функціонування медичних працівників протягом робочого дня з використанням апріорних знань про відповідні сектори функціонування. Метод дозволяє модифікувати множину часових ознак секторів в апріорну ймовірність секторів функціонування за рахунок використання концепції вибірки за значимістю.

Ключові слова: великі дані, байєсівська мережа, часове вікно

Останнім часом, науковцями було вивчено можливість визначення оптимального функціонування медичних працівників за допомогою спеціальних датчиків: наприклад, акселерометри, гіроскопи, низькочастотні аудіо пристрої, тощо [1]. Однак, незважаючи на стрімкий розвиток вказаних підходів, певні питання залишаються відкритими. При визначенні показників оптимального функціонування медичних працівників, класи функцій описуються предметно-специфічним чином. Визначення функцій є досить складним процесом, оскільки в ньому враховуються ознаки, значення яких варіюються навіть для окремих класів. Крім того, такі дії мають певні дисбалани, наприклад число входжень серед класів, число починань в тиждень, а також тривалість.

Для проведення даного дослідження було відібрано реальну множину великих даних, що характеризують різні види функцій медичних працівників. Також, було відібрано великі за обсягом дані зі спеціальних датчиків для їх використання в машинному навчанні з учителем. Дані зі спеціальних датчиків були отримані від медичних працівників лікарняного закладу, що спеціалізується на дослідженні серцево-судинних захворювань. Зазначимо, що експериментальні дослідження проводилися виключно з тими медичними працівниками та пацієнтами, які погодилися на використання спеціальних датчиків. Множина великих даних містила розмічені дані, що були відібрані протягом трьох тижнів та нерозмічені дані від 134 пацієнтів, відібрані протягом року.

У якості множини розмічених даних було використано дані щодо виконання професійних обов›язків 27 медичних працівників за період протягом трьох тижнів в грудні 2015 року. Зауважимо, що зазначені дані були розмічені

за допомогою спеціального пристрою іншими медичними працівниками, які виступали в якості спостерігачів. Перед початком дослідження ми визначили 37 класів функцій, інформацію про яких було записано спостерігачами.

Інтерпретація ознак для реальних функцій медичних працівників вимагає ретельного опрацювання. У професійній діяльності медичних працівників догляд за пацієнтами має найвищий пріоритет, однак, у даному випадку має місце проблема відсутності багатьох ознак або невірних ознак часу. Тому, у якості спостерігача було залучено медичного працівника, який керував іншим пристроєм Cowon Z2 для запису функцій медичних працівників. На даному пристрої було встановлено програмне забезпечення, за допомогою якого спостерігач вибирав клас функцій, на якому медичний працівник збирається почати процедуру та активував значок закінчення, коли процедура закінчувалася.

Зазначимо, що у випадку, коли спостерігач очікував на початок виконання процедури медичним працівником, початкова часова ознака мала відповідну затримку. Таким чином, спостерігач та медичні працівники взаємодіяли одне з одним для отримання коректних початкових часових ознак [2]. Тому, перед тим, як почати певну процедуру, медичний працівник оголошував спостерігачеві про початок свого функціонування.

У тому ж відділенні лікарняного закладу було відібрано множину нерозмічених даних, отриманих за допомогою датчика протягом року.

Функції медичних працівників мають різні властивості, що залежать від часу доби. При наявності у навчальних даних ознак та часових ознак, ми можемо модифікувати множину часових ознак в апріорну ймовірність виконуваних функцій. Крім того, якщо ми будемо використовувати початковий і кінцевий час функцій, ми можемо отримати інформацію про тривалість їх виконання. Отримання інформації про те, коли і як довго виконуються функції по догляду за хворими має важливе значення для аналізу даних [3].

У процесі дослідження був запропонований метод, що використовує інформацію про мітки часу для того, щоб побудувати розподіл апріорної ймовірності функцій протягом робочого дня, а також їх реалізацію на основі вибірки за значимістю. В результаті, отримана інформація використовується для байєсівської оцінки функцій.

Будемо вважати, що конкретний час доби виражається як ціле число між 1 та T, де послідовність $(1,2,...,T)$ позначимо як $1:T$. Ми припускаємо, що вектор ознак виділяє декілька статистичних значень з часового вікна входу датчика в околі значення t. Послідовність векторів ознак $(z_1,z_2,...,z_T)$ позначимо як $z_{1:T}$.

Введемо величину $\tilde{N}$, що вказує на множину класів функцій, які необхідно визначити. Припустимо, що у будь-який момент часу t можуть бути використані числові функції в результаті того, що медичний працівник виконує декілька функцій одночасно, або тому, що алгоритм визначення функцій проводить нечіткі оцінки. Таким чином, нашою метою є визначення того, чи при бінарному значенні t функція в момент часу t відповідає $\tilde{n} \in \tilde{N}$.

Запропонований метод може бути застосований для кожного класу функцій $\tilde{n} \in \tilde{N}$, де використовується або найбільш ймовірний клас $\arg_c \max \mathrm{P}(a_t^c)$ або всі класи, оцінені за час t. Однак, ми визначали лише одну функцію $\tilde{n} \in \tilde{N}$ та оцінювали її точність.

Ми використовуємо термін «сектор» як неперервний діапазон часу, де виконується функція $\tilde{n}$ і представляємо його у вигляді пари початкового часу та часу закінчення виконання функції. Припускаючи, що сектори L^c повторюються для функції $\tilde{n}$ на протязі робочого дня медичного працівника, сектор l з моменту часу $b(l)$ до моменту часу $e(l)$ визначається, як

$$s_l^c = (b(l), e(l)), \tag{1}$$

де $1 \le b(l) \le e(l) \le T$.

Процедура визначення функцій може бути змодельована як проблема отримання максимального аргументу $\tilde{n} \in \tilde{N}$ від $\mathrm{P}(a_t^c | z_t)$ лише для локального часового вікна t. Зауважимо, що обчислення $\mathrm{P}(z_t | a_t^c)$ може бути виконано, використовуючи теорему Байєса. Далі ми будемо називати величину $\mathrm{P}(z_t | a_t^c)$ ймовірністю локального часу. Однак, в якості нашому внеску ми будемо вирішувати проблему отримання ймовірності функцій медичних працівників впродовж повного робочого дня $\mathrm{P}(a_{1:T}^c | z_{1:T})$.

Ми припускаємо, що ймовірності між будь-якими секторами s_l^c та $s_{l'}^c$ $(l \ne l')$ є незалежними. Зазначимо, що гранична ймовірність може бути виражена у такому вигляді:

$$\mathrm{P}(z_{b(l):e(l)}, a_{b(l):e(l)}^c, s_l^c) = \mathrm{P}(s_l^c) \prod_{t \in b(l):e(l)} \mathrm{P}(z_t | a_t^c) \mathrm{P}(a_t^c | s_l^c) \ . \tag{2}$$

У випадку, коли s_l^c є фіксованим, a_t^c можна легко отримати для $b(l) \le t \le e(l)$, тому ми можемо видалити $\mathrm{P}(a_t^c | s_l^c)$. Відповідо,

$$\mathrm{P}(z_{b(l):e(l)}, a_{b(l):e(l)}^c, s_l^c) = \mathrm{P}(s_l^c) \prod_{t \in b(l):e(l)} \mathrm{P}(z_t | a_t^c) \ . \tag{3}$$

Для отримання умовної ймовірності між $a_{b(l):e(l)}^c$ та $z_{b(l):e(l)}$, ми зменшуємо s_l^c, в результаті чого має місце така рівність:

$$\mathrm{P}(a_{b(l):e(l)}^c, z_{b(l):e(l)}) = \sum_{s_l^c} \mathrm{P}(s_l^c) \prod_{t \in b(l):e(l)} \mathrm{P}(z_t | a_t^c) \ . \tag{4}$$

Далі, ми розділяємо часову послідовність $1:T$ на сектори

$$\{b(1):e(1)\}, \{b(2):e(2)\}, ..., \{b(L^c):e(L^c)\} \tag{5}$$

та розглядаємо граничну ймовірність для кожного часу $1:T$, як

$$\mathrm{P}(a_{1:T}^c, z_{1:T}) = \mathrm{P}(a_{b(1):e(1)}^c, z_{b(1):e(1)}, a_{b(2):e(2)}^c, z_{b(2):e(2)}, ..., a_{b(L^c):e(L^c)}^c, z_{b(L^c):e(L^c)}) \tag{6}$$

Якщо припустити, що пари секторів незалежні один від одного, то формула (6) записується як добуток секторних граничних ймовірностей, а саме:

$$\mathrm{P}(a_{1:T}^c, z_{1:T}) = \prod_{l \in 1:L^c} \mathrm{P}(a_{b(l):e(l)}^c, z_{b(l):e(l)}) \ . \tag{7}$$

Підставляючи

$$\mathrm{P}(a_{b(l):e(l)}^c, z_{b(l):e(l)}) = \sum_{s_l^c} \mathrm{P}(s_l^c) \prod_{t \in b(l):e(l)} \mathrm{P}(z_t | a_t^c) \ , \tag{8}$$

ми отримуємо таку рівність:

$$P(a^c_{LT}, z_{LT}) = \prod_{l \in L} \left\{ \sum_{s^c_l} P(s^c_l) \prod_{m \in N(l)} P(z_m | a^c_l) \right\}. \quad (9)$$

Отже, з огляду на вхід z_{LT}, ми маємо, що

$$P(a^c_{LT} | z_{LT}) \propto \prod_{l \in L} \left\{ \sum_{s^c_l} P(s^c_l) \prod_{m \in N(l)} P(z_m | a^c_l) \right\}. \quad (10)$$

Формула (10) використовує не лише ймовірність локального часу $P(z_l|a^c_l)$, а також апріорну ймовірність секторів $P(s^c_l)$. Тому, було використано ймовірність локального часу $P(z_l|a^c_l)$, отриману з результатів наївного методу, а також апріорну ймовірність $P(s^c_l)$, використовуючи відповідні вибірки з навчальних даних. В результаті можна зробити висновок, що оскільки ймовірність $P(s^c_l)$ може бути досить інформативною, запропонований метод може призвести до підвищення точності в процесі визначення функцій протягом всього робочого дня.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. He H. Learning from imbalanced data / H. He, E.A. Garcia // IEEE Transactions on Knowledge and Data Engineering. – 2009. – 21. – P. 1263-1284.

2. McQueen, J.M. Segmentation of continuous speech using phonotactics / J.M. McQueen // Journal of Memory and Language. – 1998. – 39. – P. 21-46.

3. Guyon I. An introduction to variable and feature selection / I. Guyon, A. Elisseeff // Journal of Machine Learning Research. – 2003. – 3. – P. 1157-1182.

Evaluation of correctness determination of the logical document structure

Andrii Osidach
Aspirant
National University “Lviv Polytechnic”
Ukraine, Lviv

Abstract

The report proposes a methodology and multi-criteria assessment of the effectiveness of information storage and retrieval of structured information, which allows for a more adequate assessment of the functioning of the real system. The report proposed as a criterion for assessing the quality of recognition of the logical structure of the documents used δ-tree which size is directly related to the time-consuming manual editing of the document structure.

Key words: ***corporate document management, electronic document, retrieval of structured information, δ-tree.***

Оцінка ефективності правильності визначення логічної структури документа

Осідач Андрій Олегович
Аспірант
Національний університет “Львівська політехніка”
Україна, Львів

Анотація

У доповіді пропонується методологія і багатокритеріальна оцінка ефективності інформаційної системи збереження та пошуку структурованої інформації, яка дозволяє більш адекватно оцінювати функціонування реальної системи. Також в доповіді запропоновано в якості критерію для оцінки якості розпізнавання логічної структури документів використовувати δ-дерево, розмір якого напряму пов›язаний з затратністю ручного редагування структури документа.

Ключові слова: корпоративний документообіг, пошук структурованої інформації, δ-дерево.

Розглянемо деякий документ $D \in \mathfrak{D}$. Припустимо, що ми маємо два впорядковані дерева T_1 і T_2, що описують логічну структуру документа D. При цьому нехай дерево T_1 описує логічну структуру, отриману в результаті автоматизованого розбору, а дерево T_2 представляє ідеальну логічну структуру. Визначимо певним чином відмінності між деревами T_1 і T_2.

Поставимо у відповідність вузлам обох дерев деякий набір унікальних ідентифікаторів.

Визначення 1. Два вузли $t_1 \in T_1$ і $t_2 \in T_2$ називаються ізоморфними, якщо вони розрізняються тільки ідентифікаторами, а їх значення і контекст рівні.

Визначення 2. Два впорядковані дерева T_1 і T_2 називаються ізоморфними, якщо для кожного вузла в одному дереві знайдеться єдиний ізоморфний вузол в іншому дереві, тобто $\forall\, t_1 \in T_1\ \exists$ єдиний ізоморфний $t_2 \in T_2$ і $\forall\, t_2 \in T_2\ \exists$ єдиний ізоморфний $t_1 \in T_1$.

Визначення 3. Назвемо відображенням повної відповідності $M_F: T_1 \rightarrow T_2$ таке бієктивне відображення, яке кожному вузлу $t_1 \in T_1$ однозначно ставить у відповідність ізоморфний йому вузол $t_2 \in T_2$.

Очевидно, що повна відповідність можлива тільки в тому випадку, коли дерева T_1 і T_2 ізоморфні.

Припустимо тепер, що дерева T_1 і T_2 не є ізоморфними. Тоді виділимо в дереві T_1 підмножину вузлів $T'_1 \subseteq T_1$, що мають ізоморфні їм вузли в дереві T_2, аналогічно, вузли дерева T_2, що мають ізоморфні їм вузли в дереві T_1 можуть бути об›єднані в підмножину $T'_2 \subseteq T_2$. Тоді, якщо підмножина T'_1 і T'_2 не порожні, можна дати наступне визначення.

Визначення 4. Назвемо відображенням часткової відповідності $M_p: T'_1 \rightarrow T'_2$ таке бієктивне відображення, яке кожному вузлу $t_1 \in T\rangle_1$ ставить в однозначну відповідність ізоморфний йому вузол $t_2 \in T\rangle_2$.

Очевидно, що для оцінки відмінностей між двома деревами T_1 і T_2 необхідно спочатку визначити деяку часткову відповідність між деревами, а потім знайти таку послідовність елементарних операцій, яка дозволяє перетворити дерево T_1 в дерево T_2.

Визначимо чотири операції, які використовуватимуться для редагування дерев:

a. Вставка. Вставка нового вузла *t* в дерево T_1.

b. Видалення. Видалення вузла *t* з дерева T_1. Виконання цієї операції можливе тільки у тому випадку, якщо вузол *t* не має нащадків.

c. Модифікація. Модифікація значення вузла *t* в дереві T_1.

d. Переміщення. Переміщення піддерева з коренем вузла *t* в дереві T_1.

Усі вищезгадані операції є стандартними операціями редагування дерев, за винятком операції переміщення. Ця операція виконується не лише стосовно окремо взятих вузлів, але і до цілих піддерев, які утворюють нащадки переміщуваного вузла.

Розглянемо тепер послідовність операцій редагування, яка перетворює одне дерево в інше.

Визначення 5. Сценарієм редагування *S* дерева T_1 відносно дерева T_2 називається така кінцева послідовність елементарних операцій (1) - (4), яка переводить дерево T_1 в деяке дерево $T\rangle_1$ таке, що дерево $T\rangle_1$ ізоморфно T_2.

Наприклад, для дерев, представлених на рис. 1, сценарій редагування *S* для дерева T_1 відносно T_2 має наступний вигляд:

S = {*вставка* вузла 23 в піддерево з коренем 3 перед вузлом 4;
видалення вузла 6;
переміщення піддерева з коренем 3}. (1)

В результаті застосування сценарію редагування (1) до початкового дерева T_1 отримуємо дерево $T\rangle_1$, ізоморфне кінцевому дереву T_2.

Очевидно, що для двох дерев T_1 і T_2 може існувати більш за один сценарій редагування. Тому необхідно ввести поняття вартості редагування. Вартість операцій редагування залежить від типу операції і вузлів, залучених в операцію. В цілях простоти тут усі операції (a)-(d) вважаються операціями одиничної вартості. Проте, вартості можуть бути відкориговані згідно з вагою різної ваги змін в ланцюгах уточнення відмінностей між ними. Загальна вартість сценарію редагування є сумою вартостей його окремих операцій.

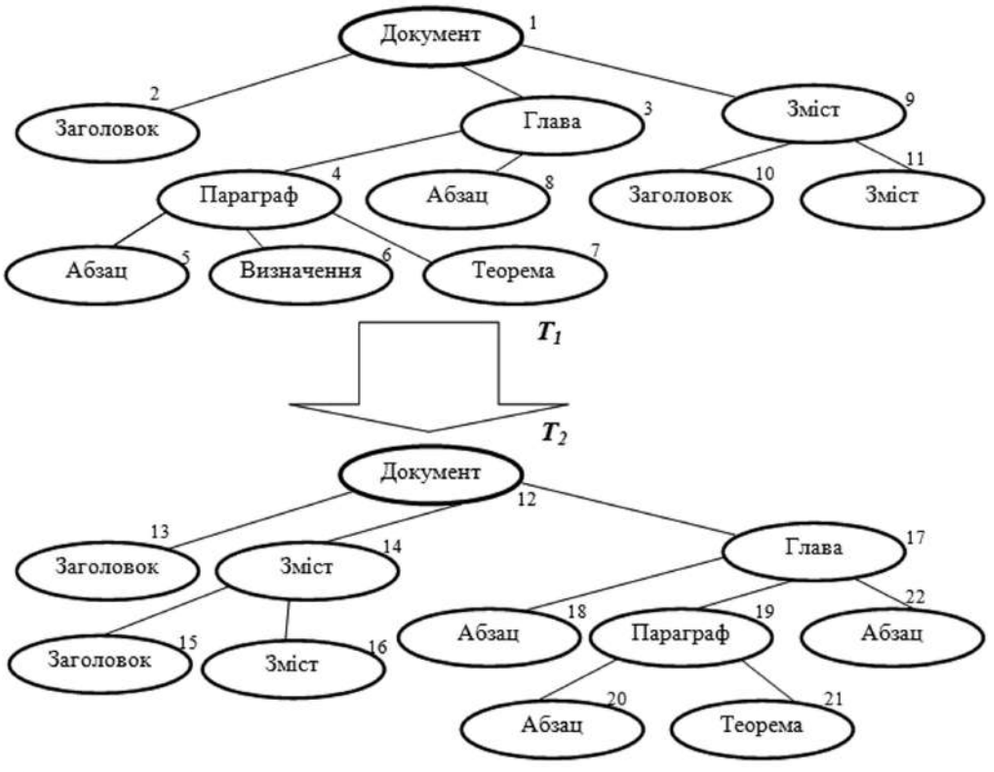


Рисунок 1. Дерева логічної структури T_1 і T_2 документа *D*

Припустимо тепер, що задано два впорядковані дерева T_1 і T_2 і безліч сценаріїв редагування S_1, S_2, ...,S_k, дерева T_1 відносно дерева T_2. Припустимо також, що кожен сценаріїв редагування S_i має вартість редагуванні C_i, i = 1, 2, …, *k*.

Визначення 6. Сценарій редагування S_{opt}, що має мінімальну вартість $C = \min_{1 \le i \le k} C_i$ називається оптимальним сценарієм редагування, а мінімальна вартість C називається відстань *d* (T_1, T_2) між впорядкованими деревами T_1 і T_2.

Таким чином, метою порівняння впорядкованих дерев є отримання оптимального сценарію редагування і відстані між деревами в сенсі визначення (1). У [5] запропонований алгоритм *EditScript* для знаходження оптимального сценарію редагування S_{opt} за час *O(n-r)*, де *n* – загальна кількість вузлів і *r* - кількість невирівняних вузлів у редагованому дереві (зазвичай *n* набагато перевищує *r*).

У [5] під невирівняними вузлами розуміються вузли, для яких вірно твердження. Нехай для дерев T_1 і T_2 задане відображення часткової відповідності M_p. Тоді якщо вузол $t_1 \in T_1$, то $t_2 = M_p(t_1)$ відповідний йому вузол в дереві T_2. Припустимо, що вузли *g* і *h* - діти вузла t_1 в дереві T_1 і вузол *g* - лівий брат вузла *h*. Згідно [5], вузли *g* і *h* є невирівняними, якщо вузли $M_p(g)$ і $M_p(h)$ - діти вузла t_2 в дереві T_2, і $M_p(g)$ є правим братом $M_p(h)$.

Відстань між деревами в даному випадку обчислюється просто як число операцій в оптимальному сценарії редагування:

$$d(T_1, T_2) = |S_{opt}|. \quad (2)$$

Іншим наочним представленням відстані між впорядкованими деревами є δ-дерева. Використання δ-дерев дозволяє відмовитися від ідентифікаторів вузлів, які є обов›язковими при побудові сценаріїв редагування і можуть тільки ускладнювати перегляд і пошук. Окрім того, δ-дерева є більш наглядним представленням сценаріїв редагування.

δ-дерева є узагальненням на ієрархічні структури δ-відношень, що використовуються в реляційних СУБД для визначення змін в кортежах, що входять в базу даних відношень. Для кожного відношення *R* тоді обчислюється набір δ-відношень, що містять відповідно кортежі, які були вставлені і видалені з відношення *R*, а також старі і нові значення тих кортежів, які були змінені [5].

Аналогічно δ-відношенням, δ-дерева повинні представляти ієрархічні структури обох дерев, а також описувати набір елементарних операцій редагування, необхідних для переведення одного впорядкованого дерева в інше, що складають сценарій редагування. Припустимо, що T_1 і T_2 - два впорядкованих дерева, δ-дерево для T_1 по відношенню до T_2 - це дерево, в якому вузли окрім власних значень, мають також відмітку про визначену до цього вузла елементарну операцію редагування.

Визначення 7. δ-дерево для дерева T_1 відносно дерева T_2 називається правильним, якщо воно має такий відповідний сценарній редагування *S*, що *S* трансформує дерево T_1 до дерева T_2.

Визначення 8. δ-дерево для дерева T_1 відносно дерева T_2 називається оптимальним, якщо воно є правильним і відповідний йому сценарій редагування *S* є оптимальним, тобто $S = S_{opt}$.

На рис.2 показане δ –дерево, яке є правильним відносно сценарію *S*, представленого формулою (1). Слід зазначити, що представлене δ-дерево є також оптимальним, а відстань між деревами *d* (T_1, T_2) = 3.

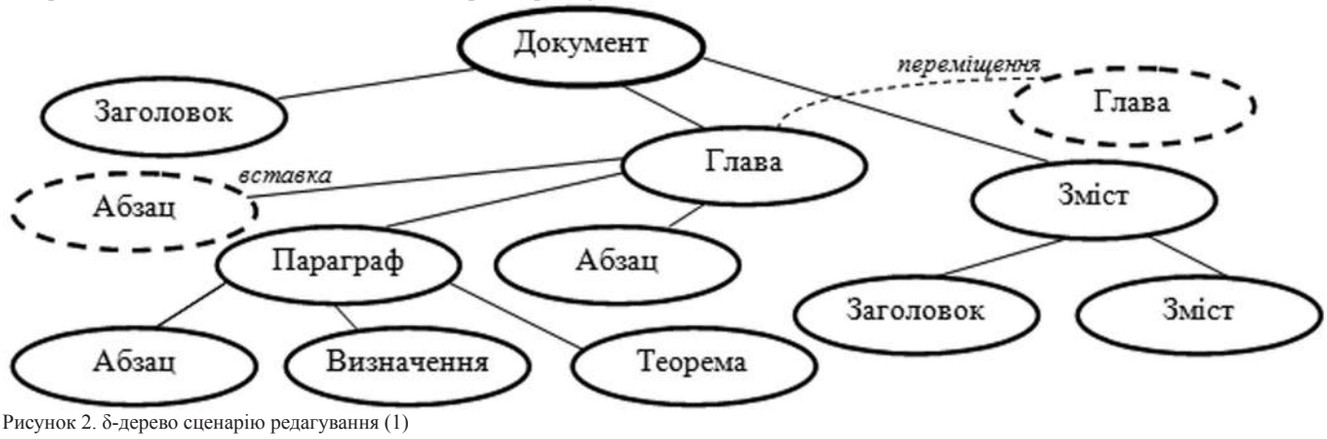


Рисунок 2. δ-дерево сценарію редагування (1)

Для побудови оптимального δ-дерева може бути використаний згаданий алгоритм *EditScript*. Для цього необхідно тільки додати операцію, яка при кожному додаванні операції редагування до оптимального сценарію редагування додаватиме відповідну вершину і мітку в δ -дерево.

Існує безліч критеріїв оцінки інформаційних систем збереження та пошуку структурованої інформації, проте чотири наступні критерії прийнято вважати основними [6]:

a. Зусилля, що витрачаються користувачем при отриманні відповідей на запити.

b. Часовий інтервал, тобто середній інтервал часу між заданням запиту і отриманням відповіді.

c. Повнота системи, тобто відсоток релевантних документів, знайдених у відповідь на пошуковий запит.

d. Точність системи, тобто відсоток релевантних документів у видачі.

СПИСОК ЛІТЕРАТУРИ.

1. Осідач А.О. Математична модель електронного документа / А.О. Осідач. – Технічні науки і технології. – №1 (1). – Чернігів, 2015. – С. 146-152.

2. Осідач А.О. Опис елементів електронного документообігу та зв'язків між ними / А.О. Осідач. – East European Scientific Journal. – 2016. – № 3. – Vol. 4 – P. 69-72.

3. Осідач А.О. Методи подання різнотипної інформації в системі електронного документообігу / А.О. Осідач. – East European Scientific Journal. – 2016. – № 5. – Vol. 5 – P. 96-101.

4. Осідач А.О. Опис моделі класу документів за допомогою граматик / А.О. Осідач – Збірник матеріалів науково-практичної конференції "Найновіші досягнення європейської науки – 2015". – Софія: "Бял ГРАД-БГ", 2015. – Т.13 – С. 65-69.

5. Chawalhe S. Managing change in heterogeneous autonomous databases. Phd thesis, Stanford University, Stanford, USA, 2009. – 308 p.

6 .Солтон Дж. Динамические библиотечно-информационные системы: Пер. с англ. – М.: Мир, 2009. – 558 с.

Рецензент: д.т.н. проф., Інститут комп`ютерних наук та інформаційних технологій Шаховська Н. Б.

Comparison of the efficiency of the automatic optimization based on the polyhedral model

Sushko Sergey
doctoral student, Pukhov Institute for Modelling in Energy Engineering
Kiev, Ukraine

Chemeris Alexander
PhD. of CS, Pukhov Institute for Modelling in Energy Engineering
Kiev, Ukraine

Small parts of the source code that consume more resources are computational loops. Polyhedral model is one of the most efficient mathematical models that allows to represent arbitrary loop as some abstract description. Polyhedral model is obtained from source code and then can be altered by using different approaches and finally reverted to the new optimized source code. This paper contains optimization results of the software that uses polyhedral model.

Keywords: automatic software optimization, polyhedral model, pluto software

Сравнение эффективности автоматической оптимизации на основе полиэдральной модели

Сушко Сергей
аспирант, Институт проблем моделирования в энергетике им. Г. Е. Пухова НАН Украины
Киев, Украина.

Чемерис Александр
канд. техн. наук, Институт проблем моделирования в энергетике им. Г. Е. Пухова НАН Украины
Киев, Украина.

Вычислительные циклы являются малыми частями программного кода, которые потребляют большинство вычислительных ресурсов. Полиэдральная модель это математическая модель, которая позволяет представить произвольный цикл в виде абстрактного описания. Полиэдральная модель получается из исходного кода, затем может быть изменена каким-либо способом и затем преобразована обратно в исходный код. Эта статья содержит результаты автоматической оптимизации приложения, использующего полиэдральную модель.

Актуальной задачей разработки программно-аппаратных комплексов является оптимизация программного обеспечения. Оптимизированное программное обеспечение выполняется быстрее, потребляет меньше ресурсов для выполнения задачи.

Несмотря на очевидную полезность оптимизации, очень много программного обеспечения не оптимизировано в должной мере или оптимизировано поверхностно. Это связано с тем, что подходы к оптимизации программного обеспечения могут значительно отличаться на различных аппаратных платформах, а также тем, что ручная оптимизация сильно зависит от квалификации разработчика.

Исследования различных алгоритмов показывают, что для широкого класса задач большее время выполнения занимают относительно небольшие участки кода. Этими участками кода являются вычислительные циклы. Основная часть времени алгоритмов тратится именно в них. При этом чем больше вложенностей циклов, тем чаще выполняется самый вложенный цикл, тем больший вклад во время выполнения вносится этим участком программы. Таким образом, основные усилия по оптимизации целесообразно прилагать именно к циклам. Такой подход позволяет значительно ограничить область применения оптимизации и, в то же время, добиться существенных результатов оптимизации.

Чтобы представить циклы в качестве модели, пригодной для дальнейшего анализа и обработки используют полиэдральную модель или метод многогранника (polyhedral model и polytope method). Полиэдральная модель является математической основой для представления циклов про-

извольной вложенности и может быть использована для оптимизации программ. Полиэдральная модель трактует каждую итерацию цикла внутри вложенных циклов как точки решетки внутри геометрических объектов, называемых многогранники, и использует аффинные преобразования, а затем преобразовывает вновь полученные многогранники обратно в эквивалентный оптимизированный исходный код, проходя по всем узлам многогранника.

Полиэдральная модель обеспечивает абстракцию на уровне описания вложенных вычислительных циклов и позволяет не только изменять представление циклов на уровне узлов внутри многогранника, но и изменять саму топологию многогранника, добавлять или сокращать, при необходимости, размерности многогранников, добиваясь определенных целей, например повышения локальности данных.

На данном этапе полиэдральная модель получила значительный исследовательский интерес и используется как мощное решение для представления, модификации и обратной генерации кода. Совсем недавно появились автоматические средства оптимизации, использующие полиэдральную модель как математическую основу.

На данный момент существуют несколько пакетов программного обеспечения, предоставляющего функционал, обеспечивающий автоматическую оптимизацию циклов. Рассматривая существующие пакеты с точки зрения охвата возможных подходов, полноты описания и научной базы, которая лежит в их основе, объектом исследования был выбран свободно распространяемый пакет программного обеспечения Pluto версии 0.11.4. Помимо опций, непосредственно влияющих на метод оптимизации, пакет имеет опцию включения распараллеливания через библиотеку OpenMP. Это также может оказывать значительный эффект на общую производительность оптимизированного алгоритма.

Для проверки эффективности опций оптимизаций данного пакета был выбран набор тестовых программ PolyBench/C 4.1. Он содержит в себе 29 коротких программ на C, каждая из которых содержит некий прикладной алгоритм**.** Кроме того, каждая программа включает в себе модуль замера времени своего выполнения, которое выводится по окончанию расчета. Все указанные особенности удовлетворяют задаче проверки эффективности оптимизации различных исходных кодов и измерения времени их выполнения. В качестве аппаратной тестовой платформы применялся настольный ПК на базе четырехядерного процессора Intel Core i5-4670K.

Тестирование представляет собой ряд последовательных действий. На первом этапе были скомпилированы и запущены на выполнение изначальные исходные коды тестового пакета с целью получения исходного времени выполнения каждого теста. Затем для каждого тестового примера выполнялся запуск пакета оптимизации Pluto с разными опциями. Полученный оптимизированный исходный код так же компилировался и затем замерялось время его выполнения. Полученные данные представлены на таблице ниже.

ТАБЛИЦА 1. Таблица времени выполнения тестовых программ при различных опциях оптимизации

Тестовый пример	Опция tile, ускорение	Опции tile + parallel, ускорение	Опции tile + L2Tile + parallel, ускорение	Опция innerpar, ускорение	Опции innerpar + parallel, ускорение	Опции inner-par + tile + parallel, ускорение	Опции tile + multipar + parallel, ускорение	Опция di-amond-tile, ускорение	Опции diamond-tile + parallel, ускорение
correlation	4,91	11,18	3,93	4,63	7,78	**11,43**	6,32	5,01	10,85
covariance	4,99	10,86	3,85	4,70	9,63	**10,87**	6,47	4,92	10,75
gemm	2,31	9,70	**11,62**	1,44	3,88	9,51	4,97	2,29	9,39
gemver	5,17	16,10	16,08	14,39	**22,07**	15,90	9,12	5,03	14,37
gesummv	0,89	2,83	2,40	1,68	4,63	**2,98**	1,64	0,89	2,78
symm	1,05	1,03	1,05	1,02	1,02	1,05	1,05	1,05	1,05
syr2k	1,41	5,26	1,44	1,42	**5,63**	5,22	2,64	1,32	5,20
syrk	0,99	2,85	1,08	0,99	2,24	2,14	1,43	1,00	2,85
trmm	3,43	**17,52**	3,62	5,23	1,22	17,38	9,06	3,46	16,43
2mm	1,94	5,61	1,86	1,61	**6,14**	5,56	3,75	1,95	5,54
3mm	1,63	4,71	1,57	1,31	**5,03**	4,75	3,14	1,63	4,73
atax	0,40	1,26	1,27	0,99	**1,45**	1,24	0,69	0,39	1,25
bicg	0,82	2,64	2,52	2,12	**2,85**	2,56	1,46	0,77	2,00
doitgen	5,02	4,00	3,43	**6,00**	0,25	4,03	5,22	5,03	3,90
mvt	3,78	11,32	**12,84**	1,16	3,68	11,48	7,16	3,58	9,99
cholesky	1,26	2,98	1,16	1,00	1,80	**3,09**	1,62	1,27	3,02
durbin	1,00	1,00	0,99	0,99	1,00	1,00	1,00	1,00	1,00
gramschmidt	1,77	3,13	1,40	1,04	2,85	3,13	2,58	1,77	3,06
lu	1,49	**3,80**	1,74	1,56	1,40	4,45	2,53	1,50	3,73
ludcmp	1,00	1,00	1,00	0,99	1,00	1,00	0,98	0,96	0,99
trisolv	0,71	1,79	1,57	1,22	1,54	**1,95**	1,02	0,74	1,31
deriche	1,28	1,98	1,58	1,00	1,91	**2,22**	1,25	1,19	1,40
floyd-warshall	0,83	1,62	1,02	0,99	0,42	1,63	1,12	0,83	1,61
nussinov	1,05	3,07	1,20	1,01	2,21	2,62	1,93	1,06	**3,09**
fdtd-2d	0,91	1,52	0,39	0,94	**2,87**	2,27	1,33	-	-
heat-3d	2,07	2,11	-	2,55	**7,98**	3,28	2,22	-	-
jacobi-2d	0,97	1,78	-	0,96	**3,59**	2,19	1,43	-	-
seidel-2d	1,19	2,80	-	1,01	**3,89**	3,75	1,97	-	-

Исследование показало, что в большинстве случаев пакет автоматической оптимизации циклов Pluto дает ускорение времени работы тестовых примеров. В то же время не существует какого-то одного наилучшего подхода для всех тестовых примеров. Для практических задач требуются исследования, какой же способ оптимизации будет наилучшим. Стоит отметить, что наилучшие показатели эффективности оптимизации для всех примеров кроме одного получены при включенной опции parallel, то есть с включенной поддержкой многоядерности.

Отдельно стоит упомянуть о невозможности работы Pluto для некоторых из тестовых примеров при некоторых опциях оптимизации. Данный факт свидетельствует о том, что несмотря на общую эффективность рассматриваемого приложения, еще существуют условия, при которых возможна его некорректная работа.

Большие разбросы результатов по алгоритмам подтверждают различную внутреннюю их структуру и, как следствие, различную способность быть эффективно преобразованными в оптимизированный код.

BIBLIOGRAPHY:

1. *Uday Bondhugula.* Effective Automatic Parallelization and Locality Optimization Using The Polyhedral model : дис. канд. / Uday Bondhugula. - The Ohio State University, 2010. - 193 с.

2. *C´edric Bastoul*. Code generation in the polyhedral model is easier than you think. In IEEE International Conference on Parallel Architectures and Compilation Techniques, pages 7–16, September 2004.

3. *Uday Bondhugula, M. Baskaran, S. Krishnamoorthy, J. Ramanujam, A. Rountev, and P. Sadayappan.* Automatic Transformations for Communication-Minimized Parallelization and Locality Optimization in the Polyhedral Model. International Conference on Compiler Construction (ETAPS CC), April 2008, Budapest, Hungary.

4. *Pouchet L.* PolyBench/C the Polyhedral Benchmark suite [Електронний ресурс] / Louis-Noël Pouchet – Режим доступу до ресурсу: http://web.cse.ohio-state.edu/~pouchet/software/polybench/#description.

Research issues of mining big data streams

Gagarin O.O.
NTUU KPI, PhD student
Kiev, Ukraine

Toporivskyi B. P.
NTUU KPI, PhD student
Kiev, Ukraine

Abstract

The current stage of information technology has shown that the use of concepts of big data is effective for a wide range of problems. To maintain a competitive decision-making processed and analyzed huge amounts previously available for analysis of data types with new intelligent processing methods data mining. Stream data mining is one of the important directions because evolving data streams methods are becoming most efficient way for real time prediction and analysis.

Keywords: big data, stream data, data mining, stream processing

INTRODUCTION

Traditional databases store are all relatively static records with no pre- defined notion of time, you can insert, update, delete, or select any record at any time if you have the authorization. Traditional databases have been used in applications that require persistent data storage and complex querying. Usually a database consists of a set of objects, with insertions, updates, and deletions occurring less frequently than queries. However during past few years have witnessed an emergence of applications that do not fit this data model and querying paradigm. Instead, information naturally occurs in the form of a sequence of data values [1].

Every day, huge volumes of sensory, transactional, and web data are continuously generated as streams, which need to be analyzed online as they arrive. Streaming data can be considered as one of the main sources of what is called big data. While predictive modeling for data streams and big data have received a lot of attention over the last decade, many research approaches are typically designed for well-behaved controlled problem settings, overlooking important challenges imposed by real-world applications [2].

Streaming data have gained considerable attention in database and data mining communities because of the emergence of a class of applications that produce these data. Data streams have some unique characteristics that are not exhibited by traditional data: unbounded, fast-arriving, and time-changing. Traditional data mining techniques that make multiple passes over data or that ignore distribution changes are not applicable to dynamic data streams. Mining data streams has been an active research area to address requirements of the streaming applications. [3]

BIG DATA MINING

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying, updating and information privacy. The term often refers simply to the use of predictive analytics or certain other advanced methods to extract value from data, and seldom to a particular size of data set. Accuracy in big data may lead to more confident decision making, and better decisions can result in greater operational

efficiency, cost reduction and reduced risk [4].

In general terms, as a common denominator of the various definitions available, 'big data' 4 refers to the practice of combining huge volumes of diversely sourced information and analysing them, using more sophisticated algorithms to inform decisions. Big data relies not only on the increasing ability of technology to support the collection and storage of large amounts of data, but also on its ability to analyse, understand and take advantage of the full value of data [5].

Doug Laney was the first one in talking about 3 V's in Big Data management:

- volume: there is more data than ever before, its size continues increasing, but not the percent of data that our tools can process;
- variety: there are many different types of data, as text, sensor data, audio, video, graph, and more ;
- velocity: data is arriving continuously as streams of data, and we are interested in obtaining useful information from it in real time
- Nowadays, there are two more V's:
- variability: there are changes in the structure of the data and how users want to interpret that data
- value: business value that gives organization a compelling advantage, due to the ability of making decisions based in answering questions that were previously considered beyond reach

Big Data mining is the capability of extracting useful information from these large datasets or streams of data, that due to its volume, variability, and velocity, it was not possible before to do it [6].

Data mining involves exploring and analyzing large amounts of data to find patterns for big data. The techniques came out of the fields of statistics and artificial intelligence (AI), with a bit of database management thrown into the mix.

Generally, the goal of the data mining is either classification or prediction. In classification, the idea is to sort data into groups. For example, a marketer might be interested in the characteristics of those who responded versus who didn't respond to a promotion.

These are two classes. In prediction, the idea is to predict the value of a continuous variable. For example, a marketer might be interested in predicting those who will respond to a promotion.

Typical algorithms used in data mining include the following:

Classification trees: A popular data-mining technique that is used to classify a dependent categorical variable based on measurements of one or more predictor variables. The result is a tree with nodes and links between the nodes that can be read to form if-then rules.

Logistic regression: A statistical technique that is a variant of standard regression but extends the concept to deal with classification. It produces a formula that predicts the probability of the occurrence as a function of the independent variables.

Neural networks: A software algorithm that is modeled after the parallel architecture of animal brains. The network consists of input nodes, hidden layers, and output nodes. Each unit is assigned a weight. Data is given to the input node, and by a system of trial and error, the algorithm adjusts the weights until it meets a certain stopping criteria. Some people have likened this to a black–box approach.

Clustering techniques like K-nearest neighbors: A technique that identifies groups of similar records. The K-nearest neighbor technique calculates the distances between the record and points in the historical (training) data. It then assigns this record to the class of its nearest neighbor in a data set [7].

DATA STREAM MINING

The developments of information and communication technologies dramatically change the data collection and processing methods. What distinguish current datasets from earlier ones are automatic data feeds. We do not just have people entering information into a computer. We have computers entering data into each other [14]. Moreover, advances in miniaturization and sensor technology lead to sensor networks, collecting high detailed spatiotemporal data about the environment.

Data mining in this context requires continuous processing of the incoming data monitoring trends, and detecting changes. Traditional one-shot systems—memory based, trained from fixed training sets and generating static models are not prepared to process the high detailed data available—are also not able to continuously maintain a predictive model consistent with the actual state of the nature, or to quickly react to changes [8].

Mining big data streams faces three principal challenges: volume, velocity, and volatility. Volume and velocity require a high volume of data to be processed in limited time. Starting from the first arriving instance, the amount of available data constantly increases from zero to potentially infinity. This requires incremental approaches that incorporate information as it becomes available, and online processing if not all data can be kept [9].

The widespread dissemination and rapid increase of data stream generators coupled with high demand to utilize these streams of data in critical real-time data analysis tasks have led to the emerging focus on stream processing. Data stream processing is broadly classified into two main categories according to the type of processing namely

data stream management: this represents querying and summarization of data streams for further processing

data stream mining: performing traditional data mining techniques with linear/sublinear time and space complexity

The next table shows the major differences between data stream processing and traditional data processing. The objective of this table is to clearly differentiate between traditional stored data processing and stream processing as a step towards focusing on the data mining aspects of data stream processing systems [10].

Stream Processing	*Traditional Processing*
Real-time processing	Offline processing
Rapid data generation relative to the available computational resources.	Normal or slow data generation relative to the available computational resources.
Storage of data is not feasible	Storage of data is feasible
Approximate results are acceptable	Accurate results are required
Processing of samples of data is the usual task	Processing of every data item/record is the usual task
Storage of aggregated and summarized data only	Storage of the raw data
Spatial and temporal contexts are particularly important	Spatial and temporal contexts are considered for certain classes of applications
Linear and sublinear computational techniques are widely used	Techniques with high space and time complexity are used if necessary

RESEARCH ISSUES OF BIG DATA STREAM MINING

Most of the traditional data mining processing methods are originated from the statistical area with progressive development and evolution, which tend to be more focused on the correctness and availability of the algorithm and lack in-depth study and attention on processing large-scale data sets, high-dimensional data processing capabilities and the execution efficiency of algorithms. In addition, there are no high standards on the space and time complexity of the algorithm.

With the development of information technology, big data problems appear gradually. It is necessary to process data with the grade of TB or even PB. Furthermore, the growth trend of big data will surpass the growth rate of corresponding data processing capacity [11].

Data stream mining is a stimulating field of study that has raised many challenges and research issues. The following is a brief discussion of some crucial open research issues:

Memory management: The first fundamental issue we need to consider is how to optimize the memory space consumed by the mining algorithm. Memory management is a particular challenge when processing streams because many real data streams are irregular in their rate of arrival, exhibiting burstiness and variation of data arrival rate over time. Fully addressing this issue in the mining algorithm can greatly improve its performance [12].

Data pre-processing: data pre-processing is an important and time consuming phase in the knowledge discovery process and must be taken into consideration when mining data streams. The challenge here is to automate such a process and integrate it with the mining techniques.

Compact data structure: Due to bounded memory size and the huge amount of data streams coming continuously, efficient and compact data structure is needed to store, update and retrieve the collected information.

Resource aware: This is a fundamental issue that considers the problem of how the limited resources, e.g., memory space and computation power, can be well utilized to produce accurate estimates. Data will be lost when the memory is used up and this would lead to inaccuracy of the mining results, thus degrade the performance of the mining algorithm [13].

CONCLUSION

Data stream mining applications address the same tasks as traditional data mining but over unbounded, continuous, fast-arriving, and time-changing data streams. These characteristics impose many new challenges for even the simplest task in traditional data mining. Most of the existing techniques cannot be adopted for the data stream environment.

In this regard, there is a need to investigate and improve data mining real-time algorithms to adapted for possible use in a wide range of industries. Streaming data analysis in real time is becoming the fastest and most efficient way to obtain useful knowledge from what is happening now, allowing organizations to react quickly when problems appear or to detect new trends helping to improve their performance.

REFERENCES

1. Issues in Data Stream Management, https://tianyesite.com/2016/01/25/issues-in-data-stream-management/,[access: 14.05.2016].

2. Krempl, G., Zliobaite, I., Brzezinski, D., Hullermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., Stefanowski, J.: Open challenges for data stream mining research. 16(1), 1–10 (2014). June

3. P. Boedihardjo: Efficient Algorithms for Mining Data Streams (2010), PhD thesis

4. Big data - Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Big_data ,[access: 14.05.2016].

5. European Data Protection Supervisor, Opinion 7/2015: Meeting the challenges of big data, 19 November 2015.

6. Wei Fan, Albert Bifet, "Mining Big Data: Current Status and Forecast to the Future", SIGKDD Explorations, 14 (2), pp1-5

7. Data Mining for Big Data, http://www.dummies.com/how-to/content/data-mining-for-big-data.html, [access: 14.05.2016].

8. Lalit S. Agrawal and Dattatraya S. Adane Models and Issues in Data Stream Mining

9. T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In 38th Symposium on the Interface of Statistics, Computing Science, and Applications. Citeseer, 2005

10. O. Maimon, L. Rokach (eds.), Data Mining and Knowledge Discovery Handbook, 2nd ed.

11. WANG, Yuan-Zhuo, JIN, Xiao-Long CHENG, Xue-Qi. Network Big Data: Present and Futur[J]. Chinese Journal Of Computers,2013.

12. L. Golab and M. T. Ozsu. Issues in Data Stream Management. In SIGMOD Record, Volume 32, Number 2, June 2003.

13. Elena I, Suzana L, Dejan G (2007) A survey of stream data mining. In: Proceedings of 8th national conference with international participation, ETAI, Ohrid

14. Muthukrishnan, S. (2005). Data Streams: Algorithms and Applications. Now Publishers.

Usage of Fuzzy Sets in Technical Research

O.M.Morgal
Dept. of Automation and Control in Technical Systems
NTUU "KPI"
Ukraine, Kyiv
m_olegm@ukr.net

O.V.Savchuk
Dept. of Automation and Control in Technical Systems
NTUU "KPI"
Ukraine, Kyiv
savchuk_l1@ukr.net

I.O.Latash
Dept. of Automation and Control in Technical Systems
NTUU "KPI"
Ukraine, Kyiv

Abstracts - The paper investigates the possibility of using the neural network technologies for improving electric radio components (ERC) diagnosing. Classification and presorting of observed objects according to their physical and technical states is proposed to operate with the RBF- neural networks in the MATLAB environment. The paper describes use of fuzzy logic theory with the training of fuzzy inference on real data.

Keywords: neural network technologies; technical diagnostics; diagnosing; electric radio components.

Використання нечітких множин в технічній діагностиці

О.М. Моргаль
Національний технічний університет «Київський політехнічний інститут»
Україна, Київ

О.В. Савчук
Національний технічний університет «Київський політехнічний інститут»
Україна, Київ

І.О. Латаш
Національний технічний університет «Київський політехнічний інститут»
Україна, Київ

Анотація - У статті досліджується можливість використання нейромережевих технологій для поліпшення діагностування електро-радіокомпонентів (ЕРК). Пропонується класифікація і попереднє сортування спостережуваних об'єктів відповідно до їх фізико-технічних станів за допомогою RBF- нейронних мереж в середовищі MATLAB. У статті описується використання теорії нечіткої логіки з підготовкою нечіткого виведення на реальних даних.

В умовах невизначеності технічного чи фізичного стану складних інфраструктур неможливо забезпечити їх необхідну якість та надійність без інтелектуального аналізу діагностичної інформації.

Постановкою проблеми є використання нечітких множин для удосконалення методів діагностування в техніці.

Завдання, що вирішувалися:

застосування електрофізичних методів діагностування для отримання первинної діагностичної інформації інтегральних мікросхем (ІМС) по інтегральним ефектам для підвищення надійності складних інфраструктур;

стиснення та інтелектуальний аналіз апостеріорної інформації із застосуванням нейромереж для розбраковки досліджуваних ІМС за їх фізичним станом у середовищі програмного пакету MATLAB та його бібліотеки Neural Network Toolbox.

Інформаційна можливість електрофізичних методів діагностування за інтегральним ефектом нелінійності та загальний підхід до апаратурного забезпечення методів технічного діагностування достатньо повно надається у [1].

Стиснення первинної діагностичної інформації про стан ЕРК виконано за допомогою дискретного розкладання Карунена-Лоева (ДРКЛ), що є розкладанням ансамбля початкових векторів за власними векторами коваріаційної матриці. Для мікросхем цей простір складається з трьох координат, та кількість матриць за типами дефектів зросте до п'яти.

Дослідження принципів обробки багатомірної інформації дало змогу обрати та обґрунтувати доцільність використання розкладу Карунена-Лоева у якості математичного апарату для опрацювання діагностичної інформації ІМС. Для практичної реалізації розбраковки ІМС по інтегральним фізичним ефектам запропоновано застосування сучасних нейромережевих технологій (багатошаровий перцептрон, карти Кохонена, радіально-базисні мережі).

Для поглибленого розвитку даного напряму пропонований апарат нечіткої логіки [2], в якому загальним підходом щодо усунення суб'єктивізму формування правил і функцій приналежності є навчання системи нечіткого логічного висновку на реальних даних. Рішення задач включає наступні традиційні етапи: 1) введення нечіткості (фазифікації); 2) логічний висновок; 3) композицію; 4) приведення

до чіткості (дефаззіфікації). Для другого та третього етапів добре опрацьовані алгоритми Мамдані, Сугено, Ларсена [3]. Зосередимося на фазифікації і побудові правил для логічного виводу.

У нечіткій логіці з n входами і одним виходом операції здійснюються за правилами, які мають такий вигляд:

$$R_j: \text{ якщо } K_1 \in A_{1,j_2} \cap K_2 \in A_{2,j_2} \cap ... \cap K_n \in A_{n,j_2}, \text{ то } Q \in S_j, \quad (1)$$

де K_i, $i = 1,\ldots, n$ — вхідні лінгвістичні змінні; Q — вихідна лінгвістична змінна; $A_{i,j}$ — нечітка множина вхідної лінгвістичної змінної K_i; R_j — нечітке правило; S_j — вихідна нечітка множина. Далі термін «лінгвістична» опускаємо

Отже, розглядаємо нечітку систему з n входами і одним виходом. Для побудови функцій належності будемо використовувати безліч числових навчальних вибірок P з n входами і одним виходом $P = \{(k_{1,j},...,k_{n,j}) \mid j = 1,...,m\}$, яка утворена m парами входів і виходів даних, де $k_{i,j}$ - значення i-й вхідної змінної K_i, що складає безліч $(k_{1,j},\ldots,k_{n,j}, q_i)$; q_i - значення відповідної вихідної змінної Q, $1 \le i \le n$ и $1 \le j \le m$.

Перед тим, як будувати нечітке відношення еквівалентності безлічі навчальних виборок даних P, необхідно впорядкувати безліч значень змінної Q в порядку зростання:

$$P' = \{(k'_{1,p},...,k'_{n,p},q'_p) \mid q'_{p_1} \le q'_{p_2}, 1,2,...,m, \text{і} 1 \le p_1 \le p_2 \le m\}, \quad (2)$$

де $(k'_{1,p},k'_{2,p},...,k'_{n,p},q'_p) \in P$.

Нечітке відношення сумісності $R(q'_{p_1},q'_{p_2})$ між змінними Q в упорядкованій безлічі навчальної вибірки даних P' можна визначити за допомогою Евклідової відстані

$$R(q'_{p_1},q'_{p_2}) = 1 - |q'_{p_1} - q'_{p_2}| / \delta, \ \delta = (\sum_{i=1}^{m-1} |q_i - q_m|)/(m-1) \quad (3)$$

де q'_{p_1} і q'_{p_2} — значення змінної Q в упорядкованій множині навчальної вибірки даних P', q_m – максимальне значення змінної Q у множини P'. Нечітке відношення еквівалентності $R^T(q'_{p_1},q'_{p_2})$ між значеннями q'_{p_1} і q'_{p_2} змінної Q множини P' можна отримати за допомогою max-min транзитивного замикання відношення сумісності $R(q'_{p_1},q'_{p_2})$. Поділимо множину даних упорядкованої навчаючої вибірки P' на основі α – перетинів відношення еквівалентності $R^T(q'_{p_1},q'_{p_2})$ на r різних підмножин G_j, $j = 1, \ldots, r$, де j-у підмножину G_j множини P' можна представити у вигляді:

$$G_j = \{(k'_{1,p},k'_{2,p},...,k'_{n,p},q'_p) \mid R^T(q'_{p_1},q'_{p_2}) \ge \alpha, \ \alpha \in [0,1] \ 1 \le p \le m, \ 1 \le p_1, p_2 \le m \quad (4)$$

де α — порогове значення, яке вибирається для розбиття множини P'; $1 \le j \le r$ і r — кількість підмножин, що отримані з множини P'.

Допустимо, що j-а множина значень Q_j змінної Q і j-а множина значень $I_{i,j}$ змінної K_i отримані з підмножини G_j множини P':

$$O_j = \{q_p \mid \forall (k_{1,p},k_{2,p},...,k_{n,p},y_p) \in G_j, 1 \le p \le m\}, 1 \le j \le r \quad \text{і} \quad (5)$$

$$I_{i,j} = \{k_{i,p} \mid \forall (k_{1,p},k_{2,p},...,k_{i,p},...,k_{n,p},q_p) \in G_j, 1 \le p \le m\} \ 1 \le i \le n \text{ і } 1 \le j \le r.$$

Таким чином, функцію належності нечітких множин для змінної Q можна отримати, використовуючи множину значень O_j, де $j = 1,\ldots,r$. Оскільки на основі α-перетинів відносини еквівалентності значення множини P' розбиті на r різних множин O_j, $j = 1,\ldots,r$, то кожну множину Q_j змінної Q можна вважати α-перетином $A_{j,\alpha}$ вихідної нечіткої множини A_j, тобто $A_{j,\alpha} = \{q \mid q \in O_j \text{ і } \mu_{A_j}(q) \ge \alpha\}$, $j = 1,...,r$, де $\mu_{A_j}(q)$ – функція належності вихідної нечіткої множини A_j змінної Q; O_j - j-а множина вихідних значень змінної Q.

На основі методу [2], розроблений алгоритм навчання, що дозволяє будувати функції належності вхідних нечітких змінних. Навчальна вибірка даних P складається з m пар входів і виходів $(k_{1,p}, \ldots, k_{n,p}, q_p)$ вхідних змінних $K_1, \ldots, K_n$ і вихідної змінної Q_p, $1 \le p \le m$. На етапі розпізнавання вирішуються завдання композиції та приведення до чіткості належності класам фізичних станів досліджуваних ІМС.

При дослідженні багатьох задач діагностування та управління технологічним процесом виготовлення мікросхем в умовах невизначеності впливаючих факторів можна і доцільно процес прийняття рішень розподілити на два етапи. На першому етапі виявляються два основні класи: придатний і брак. На другому етапі уточнюють різновиди браку, пов'язані з причинами їх появи і, як наслідок, надають рекомендації щодо аналізу відмов та позапланових деградаційних процесів при експлуатації.

Розглянемо методику складання нечіткої класифікаційної моделі на прикладі діагностування мікросхем.

Для побудови нечіткої класифікаційної моделі діагностування спочатку виявляється множина ознак, яким відповідатимуть наступні лінгвістичні змінні: рівень забруднення Si - T, механічна деформація кристала - S, тріщина - G. Названі змінні мають наступні лінгвістичні значення: T({«малий» (мт) «середній» (ст), «великий» (бт)}; S({«-сильна» (бs),»слабка» (cs)}; G({«глибока» (rg), «дрібна» (mg)}.

Далі на основі опитування технологів будуємо якісну структуру моделі діагностування. Припустимо, що в результаті отримана вирішальна таблиця.. Передбачається, що діагноз мікросхеми визначається по сполученню ознак. За методикою, приведеною в [2], будуємо функції належності для всіх значень лінгвістичних змінних T, S і G.

Далі на вирішальній таблиці будуємо два класи Z_1, Z_2 наборів лінгвістичних значень, що відповідають рішенням «придатний», «брак». Одержимо

Z_1={<мт, бs, mg>, <мт,cs,mg >, ст,cs,mg >,<бт,cs,mg >};
Z_2={<мт, бs, rg>, <мт, cs ,rg>,<ст,бs,rg>,<ст,бs,mg>, <ст,cs,rg>, <бт,бs,rg >, <бт,бs,mg >,< бт,cs,rg >}.

По класах Z_1, і Z_2, відповідно до формули (1)

$$\mu_{P_i}(x,y,z) = \underset{(\tilde{\alpha},\tilde{\beta},\tilde{\gamma}) \in L_i}{V} \mu_\alpha(x) \& \mu_\beta(y) \& \mu_\gamma(z) (i = \overline{1,k}), \quad (6)$$

де L_i - множина наборів $(\tilde{\alpha},\tilde{\beta},\tilde{\gamma})$ до рішення r_i, будуємо функції належності $\tilde{f}_1 = \mu_{\tilde{P}_1}$ й $\tilde{f}_2 = \mu_{\tilde{P}_2}$ еталонних класів $\tilde{P}_1$ і $\tilde{P}_2$, що відповідають рішенням «придатний», «брак». Одержимо

$$\tilde{f}_1(t,s,g)=\mu_{\tilde{D}}(t)\&\mu_{\tilde{A}}(s)\&\mu_{\tilde{G}}(g)\vee\mu_{\tilde{D}}(t)\&\mu_{\tilde{B}}(s)\&\mu_{\tilde{G}}(g)\vee\mu_{\tilde{B}}(t)\&\mu_{\tilde{E}}(s)\&\mu_{\tilde{G}}(g)\vee\mu_{\tilde{A}}(t)\&\mu_{\tilde{B}}(s)\&\mu_{\tilde{G}}(g) \quad (7)$$

$$\tilde{f}_2(t,s,g)=\mu_{\tilde{I}\tilde{A}}(t)\&\mu_{\tilde{A}\tilde{S}}(s)\&\mu_{\tilde{A}\tilde{G}}(g)\vee\mu_{\tilde{I}\tilde{A}}(t)\&\mu_{\tilde{B}\tilde{S}}(s)\&\mu_{\tilde{A}\tilde{G}}(g)\vee\mu_{\tilde{N}\tilde{O}}(t)\&\mu_{\tilde{A}\tilde{S}}(s)\&\mu_{\tilde{A}\tilde{G}}(g)\vee\mu_{\tilde{N}\tilde{O}}(t)\&\mu_{\tilde{A}\tilde{S}}(s)\&\mu_{\tilde{I}G}(g)\vee$$
$$\mu_{\tilde{N}\tilde{O}}(t)\&\mu_{\tilde{B}\tilde{S}}(s)\&\mu_{\tilde{A}\tilde{G}}(g)\vee\mu_{\tilde{A}\tilde{O}}(t)\&\mu_{\tilde{A}\tilde{S}}(s)\&\mu_{\tilde{A}\tilde{G}}(g)\vee\mu_{\tilde{A}\tilde{O}}(t)\&\mu_{\tilde{A}\tilde{S}}(s)\&\mu_{\tilde{I}G}(g)\vee\mu_{\tilde{A}\tilde{O}}(t)\&\mu_{\tilde{B}\tilde{S}}(s)\&\mu_{\tilde{A}\tilde{G}}(g).$$

Функції $\tilde{f}_1$, і $\tilde{f}_2$ розбивають трьохвимірний простір ознак T*S*G на дві нечіткі області, що відповідають рішенням «придатний» і «брак», і використовуються в класифікаційній моделі діагностування мікросхем..

Знаходимо множини L_1, L_2, L_3 наборів лінгвістичних значень, яким у вирішувальній таблиці відповідають рішення r_1, r_2. Приймається рішення r_j, що відповідає еталонному класу $\tilde{P}_j$.

ВИСНОВОК

Даний підхід дозволить при розбраковці мікросхем досліджувати вибірки даних більших розмірів з меншими витратами часу та більшою точністю. Загальна перспектива розвитку цих досліджень пов'язана з розробкою нових методів розв'язання задач по оптимальному управлінню складними інфраструктурами.

ЛІТЕРАТУРА

1. Савчук О.В., Кривенко К.С. Інтелектуальний аналіз діагностичної інформації складних технічних комплексів// Інтелектуальний аналіз інформації IAI-2014 / Зб. праць. – К.: Просвіта. – 2014. – С.172-177.

2. Теленик С.Ф., Моргаль О.М. та ін.. Нечітке оцінювання в задачах управління рівнем обслуговування. / Наукові записки УНДІЗ, №2(18), 2011. - С.24-43.

3. Люгер Д. Искусственный интеллект: стратегии и методы принятия решений сложных проблем, 4-е изд.

4. М.:Издательский дом «Вильямс». – 2003. – 864 с.

Improving of the efficiency of multi-processor systems, controlled by the data descriptors flow

V.I.Zhabin
National Technical University of Ukraine "KPI"
Kiev, Ukraine

V.V.Zhabina
National Technical University of Ukraine "KPI"
Kiev, Ukraine

Annotation. The concept of dynamic allocation of tasks between multiprocessor systems computing modules that enables to identify automatically parallel branches in the process of task performance for maximum load of system computing resources is considered. The possibility to accelerate the data exchange between computing modules as well as system reconfiguration in case of failure is shown.

Keywords. Parallel computing, descriptors flow, hidden parallelism, dynamic allocation of tasks, reconfiguration.

Повышение эффективности мультипроцессорных систем, управляемых потоком дескрипторов данных

В.И. Жабин
Национальный технический университет Украины «КПИ»
Киев, Украина

В.В.Жабина
Национальный технический университет Украины «КПИ»
Киев, Украина

Аннотация. Рассматривается концепция динамического распределения заданий между вычислительными модулями мультипроцессорных систем, позволяющая автоматически выявлять параллельные ветви в процессе решения задачи, что обеспечивает максимальную загрузку вычислительных средств системы. Показана возможность ускорения обмена данными между вычислительными модулями и реконфигурации системы при отказах.

Ключевые слова. Параллельные вычисления, управление потоком дескрипторов, скрытый параллелизм, динамическое распределение заданий, реконфигурация.

ВВЕДЕНИЕ

Для систем управления и моделирования в реальном времени на длительность обработки данных накладываются внешние ограничения. Сокращение времени обработки информации может быть достигнуто путем применения параллельных систем. Продолжительность решения задач в параллельных вычислительных системах во многом определяется возможностью распараллеливания процессов с целью максимальной загрузки вычислительных узлов. Для разработки параллельных программ, созданы различные средства программирования. Кроме параллельных языков, таких как Ada, Java и ряд других, которые являются самостоятельными средствами для разработки параллельных программ, находят применение различные расширения последовательных языков (например, mpC, C-DVM, HPF, Fortran-DVM), а также библиотеки параллельного программирования (OpenMP, MPI, PVM и др).

Большинство известных технологий относятся к средствам статического распараллеливания процессов. Основные задачи распараллеливания в этом случае решаются программистом на этапе разработки программ. Однако при статическом анализе алгоритмов возможности выявления параллельных ветвей весьма ограничены, в частности, не всегда удается выявить скрытый параллелизм. Это обусловлено рядом причин, основной из которых является недостаток информации о продолжительности процессов.

В связи с этим важной задачей является создание методов динамического планирования вычислений в параллельных системах.

НЕДОСТАТКИ СТАТИЧЕСКОГО ПЛАНИРОВАНИЯ ВЫЧИСЛЕНИЙ

Пусть задан граф в ярусно-параллельной форме (ЯПФ) $G=(V_i,D_i,W_i)$, где $V_i=\{v_j \mid j=\overline{1,k_i}\}$ – множество вершин i –го яруса ($i=\overline{1,s}$); D_i – множество входных дуг вершин i –го яруса; $W_i=\{w_j \mid j=\overline{1,k_i}\}$ – множество весов вершин i –го яруса. Вершинам графа соответствуют операторы, осуществляющие определенное преобразование информации. Оператор i –го яруса активизируется только после завершения работы всеми операторами предшествующих ярусов, связанных с ним дугами. Каждой вершине v_j приписан вес w_j, определяющий длительность преобразования информации.

Оператор считается активным, если в данный момент осуществляет преобразование информации. Понятно, что в каждый момент времени число активных вершин определяет максимальное число вычислительных узлов, которые осуществляют вычисления параллельно во времени.

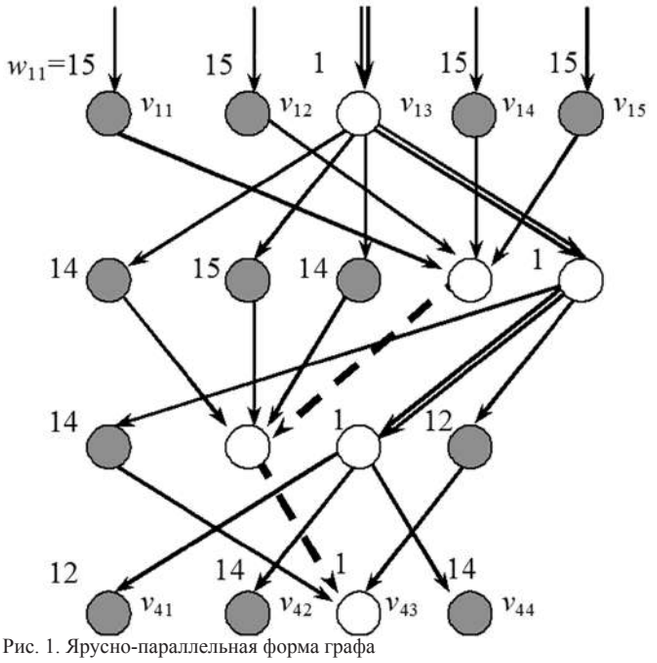


Рис. 1. Ярусно-параллельная форма графа

Зачастую на этапе статического анализа программист определяет число параллельных процессов по ширине графа. Однако задача может обладать скрытым параллелизмом и распараллеливаться в значительно большей степени. Например, в ЯПФ с шириной два и более при отсутствии ограничений на значения весов w_j число активных вершин в один момент времени может достигать значения

$$N = 2 + \sum_{i=1}^{s} (k_i - 2), \qquad (1)$$

где s - число ярусов, k_i - число вершин на i -м ярусе.

На рис.1 видно, что цепочка вершин, связанных двойными дугами (все вершины имеют только по одному входу), обеспечивает быструю активизацию вершин на следующем ярусе, а цепочка вершин, соединенных пунктирными дугами (все вершины имеют только по одному выходу), обеспечивает поэтапный переход вершин в пассивное состояние. В каждом ярусе, кроме первого и последнего, указанные функции выполняют по две вершины, а в первом и последнем ярусе – только по одной. В зависимости от значения весов одновременно в активном состоянии может находиться разное число вершин, но не более чем указано в (1). Вершины, которые могут одновременно находиться в активном состоянии, на рис. 1 затемнены (обозначения и веса вершин показаны выборочно, чтобы не затенять рисунок). В рассматриваемом случае при ширине ЯПФ, равной 5, одновременно могут выполняться 12 операторов. Очевидно, что число одновременно обрабатываемых операторов может существенно превышать ширину ЯПФ. Например, для регулярного ортогонального графа с шириной k и длиной s нетрудно показать, что число активных вершин может составлять $N = k - 2s + 2$. В статике возможности широкого распараллеливания вычислений не очевидны.

К другим недостаткам статического анализа алгоритмов можно отнести следующее. При разработке программ,

как правило, учитывается конфигурация аппаратных средств системы, изменение которой может потребовать повторной разработки программы.

Серьезные проблемы возникают при многозадачном режиме функционирования систем. Взаимодействие процессов может быть предусмотрено для задач, зарегистрированных предварительно в одной группе (коммуникаторе). Обеспечить это весьма проблематично, например, при решении задач управления в реальном времени, когда стратегия управления может изменяться под воздействием различных факторов.

Средства динамического распараллеливания вычислений позволяют выявить параллельные ветви, которые возникают непосредственно в процессе вычислений. С этой целью были предложены модели вычислений под управлением потоков данных (data flow) [1, 2]. Распараллеливание в таких системах осуществляется преимущественно на аппаратном или микропрограммном уровне, что существенно разгружает операционную систему и уменьшает задержку при реализации мелкозернистых параллельных алгоритмов. Определяющим в данном случае является не порядок выполнения команд, а доступность данных для команды. Однако с увеличением зернистости алгоритмов существенно возрастают объемы пересылаемых данных, что делает такую модель вычислений неэффективной.

КОНЦЕПЦИЯ ДИНАМИЧЕСКОГО РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ МЕЖДУ ВЫЧИСЛИТЕЛЬНЫМИ СРЕДСТВАМИ СИСТЕМ

В работах [3–6] предложены методы динамического распределения заданий в параллельных системах, ориентированные на реализацию алгоритмов с крупнозернистой структурой. В данной работе обобщаются полученные результаты, и рассматриваются возможности повышения эффективности мультипроцессорных систем, управляемых потоком дескрипторов данных.

Предполагается, что мультипроцессорная система содержит однотипные вычислительные модули (ВМ), которые обмениваются данными через общее адресное пространство и имеют распределенные средства аппаратной реконфигурации. При этом каждый ВМ имеет локальную память для независимого выполнения собственной программы.

В основу концепции динамического распределения заданий между ВМ положены следующие положения.

1. Задачи представляются с помощью потокового графа (его описанием), каждой *i*-й вершине которого соответствует определенный объем работы (вычислений), а каждой дуге – поток данных, необходимых для выполнения работы.

2. Вычисления, соответствующие вершинам графа, являются независимыми и взаимодействуют между собой только через данные.

3. Для преобразования информации создается библиотека функций. В качестве компонентов библиотеки могут использоваться программные модули, функционирующие в заданной операционной среде и позволяющие осуществить необходимое для решения задачи преобразование данных.

4. Граф задачи разрабатывается без учета числа ВМ в системе.

Потоковый граф $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{D})$ имеет множество вершин $\boldsymbol{V}$ и множество дуг $\boldsymbol{D}$.

Каждой вершине соответствует задание (определенный объем работы, процесс), которое описывается дескриптором $\boldsymbol{V_i} =\{N_i,I_i,\boldsymbol{P}_i,Q_i\}$, где N_i – имя данного дескриптора; I_i – идентификатор задания (определяет функцию преобразования данных); $\boldsymbol{P}_i=\{p_{ij}\}$ – множество имен выходных данных (соответствуют дугам, выходящим из *i*-й вершины и входящим в *j*–ю вершину); Q_i –число дуг графа, входящих в *i*-ю вершину). Имя потока выходных данных p_{ij} может быть представлено кортежем $p_{ij}=<N_j, n_{ij}, Q_j>$, где n_{ij} – имя входа *j*-й вершины графа, связанной с *i*-й вершиной.

Поток данных, соответствующий дуге графа, соединяющей *j*-ю вершину с *i*-й вершиной, характеризуется дескриптором данных $\boldsymbol{D}_{ji} =\{p_{ji}, A_{ji}\}$, где A_{ji} – элемент адресации данных, определяющий расположение данных в памяти системы.

Из элементов дескрипторов в соответствии с определенной процедурой *F* формируются заявки на выполнение *i*-го задания $F(V_i, D_i) \rightarrow \boldsymbol{Z}_i =\{I_i, \boldsymbol{P}_i, \boldsymbol{A}_i\}$, где $\boldsymbol{A}_i$ – множество элементов адресации данных для *i*-го задания.

Для формирования заявок $\boldsymbol{Z}_i$ используется управляющий ВМ (УВМ), в котором создается таблица (массив объектов), *i*-я строка S_i которой имеет вид $N_i : S_i = <I_i, \boldsymbol{P}_i, \boldsymbol{A}_i, \boldsymbol{L}_i, C_i, Q_i >$, где N_i – имя строки; C_i –счетчик дескрипторов, $\boldsymbol{L}_i=\{\delta_{ji}\}$ – признаки наличия поступивших дескрипторов данных. Число признаков в строке S_i равно числу входных потоков данных Q_i.

В УВМ вводятся дескрипторы и исходные данные. При поступлении дескрипторов заданий в соответствующие позиции строк таблицы вводятся значения I_i и $\boldsymbol{P}_i$. При поступлении дескрипторов данных в соответствующие позиции строк записываются элементы множества $\boldsymbol{A}_i$ и соответствующие им признаки δ_{ji}. Полное накопление множества $\boldsymbol{A}_i$ элементов адресации данных для задания определяется с помощью счетчика C_i. При поступлении любого дескриптора сравниваются значения Q_i и C_i, затем C_i увеличивается на единицу. Равенство указанных значений является условием активизации заявки.

Заявка передается в свободный исполняющий ВМ (ИВМ) по его запросу. При этом обнуляется счетчик C_i, то есть система готова для повторного формирования заявки с данным именем, если это необходимо. В системе одновременно могут решаться несколько задач.

На стадии формирования заявок между модулями пересылаются только короткие сообщения (дескрипторы). Потоки непосредственно данных перемещаются только на этапе выполнения заданий. В этом случае вычислительным процессом управляют стандартные средства операционной системы.

МИНИМИЗАЦИЯ ПЕРЕСЫЛОК ДАННЫХ

В мультипроцессорных системах обмен данными между ВМ осуществляется через общее адресное пространство, которое физически может представляться разными модулями памяти с различным способом и временем до-

ступа к ним [6]. Для ускорения вычислений необходимо стремиться к минимизации числа пересылок данных. Рассмотрим несколько таких возможностей.

ИВМ после выполнения задания возвращает УВМ только дескрипторы полученных данных. Сами данные при этом целесообразно оставлять в локальной памяти ИВМ, которая доступна другим ВМ. Это может быть коммуникационная память или вся локальная память ВМ при организации оконного доступа [6]. При такой организации вычислений для минимизации пересылок данных можно использовать весьма простой механизм. Как правило, элементы адресации A_{ji} в явной или неявной форме содержат информацию о ВМ, в котором сохраняются данные. Для ИВМ среди готовых заявок ищется такая, в которой имеется максимальное число ссылок на его локальную память. Если поиск оказывается успешным, то отпадает необходимость, по крайней мере, одной пересылки данных.

При реализации итерационных процессов целесообразно сохранять в таблице формирования заявок некоторые данные для повторного счета. Элементы адресации таких данных должны иметь соответствующий признак. После передачи готовой *i*–й заявки в ИВМ в счетчике C_i устанавливается число, которое равно количеству сохраняемых данных для повторного формирования заявки. Это также приводит к уменьшению пересылок.

Кроме того, для уменьшения пересылок в качестве элементов A_{ji} могут передаваться непосредственно данные, имеющие короткий формат. Естественно, для этого в передаваемом сообщении должен быть предусмотрен признак непосредственной адресации.

ПРОГРАММНО-АППАРАТНАЯ РЕКОНФИГУРАЦИЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Для систем с общей магистралью, построенных на однотипных процессорных модулях естественным является скользящее резервирование. Это объясняется не только универсальностью резерва (все модули одинаковы), но и возможностью подключения резерва в любую точку общей шины. Замена модуля не требует включения резервного модуля именно в то место шины, где находился отказавший модуль.

Предлагается метод контроля и реконфигурации систем, который обладает следующими свойствами: вносит минимальную информационную избыточность (минимально усложняет программные средства); для уменьшения расхода команд обеспечивает голосование и отключение неисправных модулей с помощью распределенных между модулями аппаратных средств; автоматически производится реконфигурация системы (вместо отказавшего модуля вводится резервный); позволяет использовать любое число резервных модулей без усложнения схемы реконфигурации.

Метод базируется на контроле временных интервалов и событий. Управление процессами (объектами) в реальном времени может быть реализовано в виде повторяющихся циклов. В каждом цикле управления вводится информация о состоянии процесса, затем осуществляется преобразование информации с целью определения необходимого воздействия на управляемый процесс. Полученные результаты используются для изменения параметров управления. На основании длительности цикла определяются необходимые временные интервалы для таймеров УВМ и ИВМ.

Между процессорными модулями распределены аппаратные средства, предназначенные для ускорения процесса реконфигурации системы при отказе процессоров. Они представляют эстафетную цепочку передачи сигнала назначения нового УПМ при отказе текущего и логические цепи голосования ИВМ.

Работоспособность УВМ проверяется совместным голосованием ИВМ. Стратегия голосования выбирается исходя из необходимого минимального числа работоспособных процессоров, которые, с учетом деградации системы, могут обеспечить решение поставленной задачи. Если система может выполнять свои функции при наличии g из n ВМ ($g<n$), то УВМ считается неисправным, если за его отключение проголосуют g или более процессорных модулей. В соответствии со значением g настраивается логическая схема голосования.

Один из простых алгоритмов работы системы состоит в следующем. В начале каждого цикла УВМ запускает все ИВМ в широковещательном режиме. Затем выполняет свою программу. ИВМ запускают свой таймер и начинают выполнять свою программу, после завершения которой ожидают срабатывания своего таймера. Если при срабатывании таймера отсутствует сигнал запуска нового цикла со стороны УВМ, то ИВМ голосует за отключение УВМ. Отключенные ВМ в голосовании участия не принимают. При наличии не менее g голосов эстафетная цепочка передает функции УВМ следующему по цепочке ИВМ, который активизируется для работы в качестве УВМ.

Работоспособность ИВМ контролируется со стороны УВМ. Простой алгоритм контроля также связан со срабатыванием таймера в УВМ. Таймер запускается на время гарантированного выполнения программ в ИВМ. По срабатыванию таймера УВМ проверяет готовность каждого ИВМ к новому циклу. При отсутствии признака готовности соответствующий ИВМ отключается от системной шины, а вместо него подключается и запускается резервный ВМ или происходит перераспределение заданий между ВМ.

Процессоры затрачивают на взаимный контроль и голосование всего несколько команд на протяжении цикла управления. Если в цикле управления у процессоров имеется оперативный запас времени, то они могут для самоконтроля выполнять тестовую программу. Это позволяет повысить достоверность вычислений.

Рассмотренный программно-аппаратный метод реконфигурации позволяет по сравнению с программными методами, требующими при голосовании многократного обращения процессоров к общей памяти, ускорить процесс реконфигурации систем.

ЗАКЛЮЧЕНИЕ

Рассмотренные методы реализации вычислений в мультипроцессорных системах позволяет устранить ряд проблем, связанных с традиционным планированием вы-

числений. Существенно упрощается процесс подготовки задачи, поскольку нет необходимости на основе статического анализа выявлять параллельные процессы и заниматься их синхронизацией.

Подготовка задач не зависит от числа процессорных модулей в системе. Благодаря этому реконфигурация системы не приводит к необходимости повторной подготовки задач, что создает потенциальную возможность продолжать вычисления при отказе процессоров.

Система может работать в многопрограммном режиме, для чего не требуется обязательная регистрация всех задач перед началом счета. Начинать решение новой задачи можно в любой момент времени, независимо от состояния других задач. При этом благодаря управлению вычислениями со стороны потоков дескрипторов данных возможна реализация скрытого параллелизма задач, что создает дополнительные возможности распараллеливания вычислений.

За счет применения аппаратных средств голосования и реконфигурации уменьшается расход команд при восстановлении системы. Это позволяет уменьшить непроизводительные затраты времени, что также создает предпосылки для ускорения вычислений, то есть повышения эффективности применения мультипроцессорных систем.

ЛИТЕРАТУРА

1. Silva J. Design of processing subsystems for Manchester data flow computer / J.Silva, J.Wood // IEEE Proc. N.Y. – 1981. – Vol. 128, N 5. – P. 218 – 224.

2. Watson R. A practical data flow computer / R.Watson, J.Guard // Computer. – 1982. – Vol. 15, N 2. – P. 51-57.

3. Жабин В.И. Метод распараллеливания процессов в вычислительных системах / В.И.Жабин // Вісник Національного технічного університету України “Київський політехнічний інститут”. Інформатика, управління та обчислювальна техніка. – 2000. – № 34. – С. 136-142.

4. Жабин В.И. Реализация параллельных процессов в вычислительных системах / В.И.Жабин // Искусственный интеллект. – 2002. – №3. – С. 235-241.

5. Жабин В.И. Реализация вычислений под управлением дескрипторов данных в мультипроцессорных системах / В.И.Жабин // Электронное моделирование. – 2003. – Т. 25, № 1. – С. 35-47.

6. Жабин В.И. Архитектура вычислительных систем реального времени / В.И.Жабин. – К.: ТОО “ВЕК+”, 2003. – 176 с.

Developing of a recommender algorithm for the books

Dmytro Svynarenko
Software Engineer, EPAM Systems
Ukraine, Kyiv

This work is about recommender systems for the books. The aim of this work is to develop a special recommender algorithm for the books social graph through a combination of existing recommender algorithms that use the social graph as a model. Scientific innovation is that it proposed a new recommender algorithm, which is based on using the social graph.

Keywords: recommender systems, social graphs, centrality, PageRank, Link Prediction

Розробка рекомендаційного алгоритму книжок

Дмитро Свинаренко
Інженер програмного забезпечення, EPAM Systems
Україна, Київ

Ця робота присвячена рекомендаційним мережам книжок. Метою даної роботи є розробка спеціального рекомендаційного алгоритму для соціального графу книжок на основі поєднання існуючих рекомендаційних алгоритмів, які використовують соціальний граф в якості моделі. Наукова новизна полягає в тому, що було запропоновано новий рекомендаційних алгоритм, який в основі використовує соціальний граф.

Ключевые слова: рекомендаційні системи, соціальний граф, центральність, PageRank, Link Prediction

Рекомендаційні системи виникли і почали розвиватися з середини 90-х років минулого століття. Основне завдання рекомендаційної системи – це надання персоналізованих рекомендацій користувачу, які враховують його уподобання при виборі предметів (товарів, об’єктів або послуг).

Рекомендаційні сервіси книжок є дуже актуальними зараз, оскільки дозволяють значно скоротити час пошуку схожих книжок за змістом до даної книги. Проте значною

проблемою є точність рекомендацій. Дуже важливо мати рекомендаційну мережу, яка може надавати точні результати.

У наш час не так багато дійсно якісних рекомендаційних мереж нехудожньої літератури. Більшість з них надають дуже неточні результати і такі мережі не являються ефективними.

В якості моделі рекомендаційних мереж, як правило, використовується соціальний граф. Соціальний граф (англ. Social graph) — це граф, вузли якого представлені соціальними об›єктами, такими як профілі користувача з різними атрибутами (наприклад: ім›я, день народження, рідне місто, тощо), співтовариства, медіа-контент, тощо, а ребра — соціальними зв›язками між ними [1, с. 2].

В якості основних методик, які використовуються для створення рекомендаційного графа можна зазначити наступні: фільтрація вмісту та колаборативна фільтрація.

При фільтрації вмісту створюються профілі користувачів і об›єктів. Профілі користувачів можуть містити демографічну інформацію або відповіді на певний набір питань. Профілі об›єктів можуть містити назви жанрів, імена акторів, імена виконавців, тощо. Або якусь іншу інформацію в залежності від типу об›єкта.

При колаборативній фільтрації використовується інформація про поведінку користувачів у минулому — наприклад, інформація про придбання або оцінки. В цьому разі не має значення, з якими типами об›єктів ведеться робота, але при цьому можна брати до уваги неявні характеристики, які складно було б врахувати при створенні профілю. Основна проблема цього типу рекомендаційних систем — «холодний старт»: відсутність даних про користувачів чи об›єкти, які нещодавно з›явились у системі.

Наведені вище методики мають один значну специфіку – вони орієнтовані на графи, які містять данні про користувачів та об’єкти. Проте при створенні рекомендаційного сервісу книжок немає даних про користувачів, а є інформація тільки про книжки. Але інформація про книжки є відносно зв’язною, оскільки кожна книжка має бібліографічний покажчик на інші книжки. Цю властивість «є в бібліографії» можна використати при побудові соціального графу книжок. Якщо в соціальному графі в якості вершини виступає профіль користувача, то у випадку з книжками профіль книги буде вершиною, а властивість «є в бібліографії» будуть слугувати ребрами графу. Таким чином отримаємо соціальний граф для книжок.

Після отримання соціального графу з книжками можна приступати до його аналізу, для цього можна використати: алгоритми, які використовують метрику центральності [1, с. 3]; алгоритм випадкового графу [2, с. 913]; алгоритм PageRank [3, с. 163]; алгоритм Link Prediction [4, с. 2].

Центральність (англ. Centrality) — ступінь, яка показує «важливість» або «вплив» певної вершини (кластера вершин) всередині графа. Стандартні методи вимірювання «центральності» включають в себе центральність з посередництва, центральність по близькості, центральність власного вектора, альфа центральність та центральність за ступенем. Таким чином можна знайти «найвпливовішу» книгу в графі.

PageRank — сімейство алгоритмів оцінки важливості вершини графу за допомогою розв’язання систем лінійних рівнянь. Для кожної вершини обчислює дійсне число, чим більше число — тим «важливіша» ця вершина. Дуже схожий на центральність, проте використовує інший підхід при пошуку найвпливовішої книги.

Link Prediction – алгоритми прогнозування потенційних зв’язків між двома вершинами графу в майбутньому. В основі лежать еволюції графу за певні проміжки часу, звідки й прогнозуються ймовірні зв’язки. Такі алгоритми дають змогу зв’язувати старі книжки з новими, які між собою не мають властивості «є в бібліографії», проте мають спільну тематику.

У результаті роботи цих алгоритмів можна отримати соціальний граф книжок, вершини якого доповнилися відповідними атрибутами «важливості» книги (за тим чи іншим алгоритмом) та ребрами «потенційно» зв’язаних вершин.

Як уже було зазначено раніше, в якості моделі рекомендаційного графу книжок буде використовуватися соціальний граф.

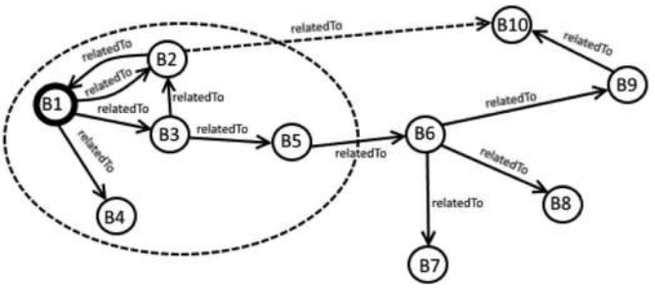
Вершини будуть містити властивості притаманні книжкам. До таких властивостей можна віднести: ISBN, назва, рік та назва видавництва, автор, коротка відомість про книгу, категорія/УДК. Ця інформація є необхідною умовою для подальшої роботи.

Ребра являють собою зв’язок між книжками типу «є в бібліографії». Наприклад, якщо книга А має в своїй бібліографії посилання на книгу Б, то якщо перейти на мову графів – з вершини А до вершини Б буде проведено відповідне однонаправлене ребро.

Для того, щоб звузити кількість потенційно схожих книжок зазначається категорія книжки або УДК. Це дає змогу виділити в соціальному графі соціальне коло, яке й буде підграфом, і воно надалі буде використовуватися в якості вхідних даних для рекомендаційного алгоритму.

Рік видавництва являється основною вимогою для роботи алгоритму Link Prediction, оскільки таким чином існує можливість відтворити відображення графу для певного проміжку часу. Наприклад, можна відстежити які книги були видані до 2015 року, потім додати нову книгу 2016 року, на яку не можуть посилатися книжки, які були видані раніше. Проте після роботи алгоритму Link Prediction старі книжки можуть отримати зв’язки з більш новими книжками, що суттєво збільшує точність і якість роботи алгоритму.

У загальному випадку запропонований соціальний граф книжок буде мати наступний вигляд:



B1,…B10 – вершини, які представляють книжки;

B1 – вершина, для якої будуються рекомендації;

B1, B2, B3, B4, B5 – соціальне коло, яке буде використовуватися для роботи алгоритму;

зв'язок «Bn «relatedTo» Bm» означає те, що книжка n має бібліографічне посилання на книжку m;

ребро, яке виділено пунктиром – результат роботи алгоритму Link Prediction, коли більш стара вершина отримує зв'язок з більш новою.

Основною задачею рекомендаційного алгоритму є пошук книжок, які можна рекомендувати для подальшого опрацювання після даної книжки. Таким чином алгоритм повинен знайти дуже схожі за змістом книги. Якщо перейти до мови графів, то в якості вхідних даних є соціальний граф книжок та вершина, для якої необхідно побудувати рекомендації.

Крок 1. Для роботи алгоритму спочатку необхідно виділити соціальне коло з соціального графу книжок. Для цього можна визначити декілька можливих методів:

- для даної вершини будується підграф, глибина якого є довільне число. Таким чином обираються всі зв'язані вершини до даною з певною глибиною. Це є дуже простий метод, який дозволяє значно звузити область пошуку. Проте такий метод має свої недоліки. Основним недоліком є те, що такий метод дуже грубо відсікає область пошуку. Та водночас постає питання пошуку такого довільного числа глибини, при якому точність роботи алгоритму є найбільшою;
- для даної вершини будується підграф, який являється категорію або УДК даної книги. Такий метод дозволяє покращити якість звуження області пошуку. Недоліком такого метода є те, що він відсікає рекомендації з інших категорій, що також може виключити з пошуку досить релевантні результати;
- для даної вершини будується підграф, який являється всіма книгами даного автора. Навряд чи доцільно використовувати в якості основного метода, проте даний підхід може покращити результати, оскільки часто один автор може мати декілька книжок с одною тематикою.

Крок 2. Після того, було отримане соціальне коло далі необхідно провести ранжування всіх вершин. Для цього можна застосувати наступні методи:

- для кожної вершини підраховується кількість спільних зв'язків з даною вершиною – k_{cr}. Наприклад, книжка А має бібліографічне посилання на книжки Б, В, Г, Д, а книжка Я має посилання на книжки В, Г, Д, Ж. Таким чином кількість спільних зв'язків буде дорівнювати 3. Фактично для двох книжок кількість спільних зв'язків показує кількість спільних книжок в їх бібліографіях. Чим більше таких зв'язків, тим більше ці книжки пов'язані між собою і тим більша ймовірність того, що в ній можна знайти додаткову інформацію, якої немає в даній книжці, а отже її можна рекомендувати для подальшого опрацювання;
- для кожної вершини розраховується значення центральності за посередництвом – k_c. Таким чином можна обрати книги, через які можна провести найбільше зв'язків. Це дозволяє ранжувати книги за важливістю та центральністю;
- для кожної вершини розраховується значення PageRank – k_{PR}. Якщо розглянути всі книжки як мережу, а кожну книгу як сторінку, то можна використати алгоритм PageRank, який зможе виділити книгу, посилання якої має найбільшу вагу.

Крок 3. Використавши ці методи можна отримати значення рекомендації – R. Чим більше це значення, тим більше ймовірність того, що саме ця книга являється максимально схожою до даної і саме її можна рекомендувати для подальшого опрацювання.

У загальному випадку значення рекомендації для вершини i буде виглядати наступним чином:

$$R_i = b_{cr} \cdot k_{cr_i} + b_c \cdot k_{c_i} + b_{PR} \cdot k_{PR_i},$$

де b_{cr} – значення коефіцієнту підсилення значення кількості спільних зв'язків,

k_{cr_i} – значення кількості спільних зв'язків для вершини i,

b_c – значення коефіцієнту підсилення значення центральності за посередництвом,

k_{c_i} – значення центральності за посередництвом для вершини i,

b_{PR} – значення коефіцієнту підсилення значення PageRank,

k_{PR_i} – значення PageRank для вершини i.

Значення коефіцієнтів підсилення підбирається експериментальним шляхом на реальних даних.

Для покращення якості роботи алгоритму час від часу для соціального графу необхідно застосовувати алгоритм Link Prediction. Це дозволить створити зв'язки між старими та новими книжками.

Отриманий алгоритм можна широко використовувати для рекомендаційних мереж нехудожньої літератури, проте слід зазначити, що максимальної точності необхідно налаштувати коефіцієнти підсилення для різних наборів даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. J.M. Podolny, J.N. Baron. Resources and relationships: Social networks and mobility in the workplace. — American Sociological Review, 1997

2. L. Lu. "The Diameter of Random Massive Graphs". In SODA'00, Proceedings of the Twelth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 912–921. ACM/SIAM Press, 2000.

3. David Liben-Nowell, Anand Rajaraman, Jeffrey D. Ullman (2014). 5.1 PageRank. Mining of Massive Datasets.

4. J.M. Podolny, Jon Kleinberg. The Link Prediction Problem for Social Networks. — ACM/SIAM Press, 2004

Рецензент: к.т.н., доц. каф. ТК НТУУ «КПІ» Сирота О.П.

CPU/GPU hybrid cluster usage in cryptography HPC

Ripnevsky O.O.
NTUU "KPI" student
Ukraine, Kyiv

This article describes CPU/GPU cluster systems, their advantages and disadvantages,usability in cryptography crack systems. Example of such system demonstrated further.

Key words: CPU, GPU, parallel computing, cloud computing

Використання гібриду CPU/GPU у криптографічних високопродуктивних обчисленнях

Ріпневський О.О.
Студент НТУУ "КПІ"
Україна, Київ

В даному докладі розглянуто кластерні системи CPU/GPU,їх переваги і недоліки, їх використання у системах криптографічного злому.Демонструється один з зломщиків,що використовує таку структуру.

Ключові слова: CPU, GPU, паралельні обчислення, хмарні обчислення

1.ВСТУП

Розвиток систем з різноманітними архітектурами основного процесорного елементу, що раніше рухав індустрію у шлях, відомий нам зараз, потрохи починає відставити в зв'язку з технічними або практичними недоліками таких систем. Доцільним є використання найкращих рис різних систем за для найвищої продуктивності.

У час, коли завдання починають вимагати все більших потужнотей для адекватної роботи, використання застарілих та неефективних засобів не ефективно. Важливо також пам'ятати, що даних стає більше, тому необхідно використовувати систему, що може масштабуватися в залежності від поставленої задачі, наприклад роботи з Big Data. За останні час, хмарні технології та кластери стали дуже популярні, через необхідність в обробці значних об'ємів даних і їхню можливість до масштабування, разом з тим кластери мають мати високу надійність, доступність, і мати можливість швидко адаптуватися розмірами системи до задач.

2.КЛАСТЕРИ

Актуальним є використання гібридних кластерів, тобто тих що складаються з декількох CPU, які працюють разом з декількома GPU. GPU дозволяють виконувати паралельні обчислювання, тому що більшість операцій підтримуються на рівні ядер. Це дозволяє отримувати максимальну обчислювальну потужність, через відсутність обробників команд, та більш практичне використання комп'ютерних ресурсів. Сьогодні використання таких технологій, як OpenGL або CUDA, дозволяє програмам працювати значно швидше, ніж на мультиядерних, чи навіть мультипроцесорних, системах.

Хмарні технології - це наступник традиційних кластерів та дата центрів. Основа роботи хмарних технологій полягає в тому, що не потрібно придбати дорогу мультипроцесорну систему, та надавати технічну підтримку і супроводження. Кінцевому споживачу потрібно платити за робочий час конкретної машини, яка може знаходитись в іншому кінці земної кулі. При цьому (згідно досвіду роботи автора з Amazon) при завантаженні значних об'ємів даних ціна виявиться значно менше ніж купівля апаратного забезпечення. Також на основі хмарних технології реалізуються різноманітні послуги, такі як зберігання даних в хмарному просторі і моніторинг різноманітних систем. Через свою маркетингову особливість, а саме оплату лише за використаний час, хмарні технології є дуже багато обіцяючою віхою комп'ютерного розвитку [1].

Ці технології є дуже практичними у розв'язанні складних обчислювальних задач, що можуть бути поділені на багато незалежних частин. Конкретно ця особливість дозволяє подолати парадигму SPMD (одна і та сама програма, що має працювати з декількома одиницями даних).

Це робить шифрування\дешифрування та відновлення паролів ділом простим і не тривалим. З іншої сторони гі-

бридні обчислювання, що виконанні у хмарних середовищах, можуть бути використанні у криптографії і криптоаналізі.

3.ВІДНОВЛЕННЯ ПАРОЛІВ

У багатьох ситуаціях (наприклад, відновлення даних, або тестування на проникнення) необхідно відновити відкритий текст пароля, що шифрується за допомогою криптографічного одностороннього хешу. Однією з визначаючих характеристик криптографічного хешу, на відміну від не криптографічної хеш-функції (наприклад, CRC-32) є те, що він призначений для демонстрації сильного лавинного ефекту (зміна в одному біті на вході потягне за собою зміну половину вихідних бітів)[2].

Деякі алгоритми хешування, такі як MD5, демонструють слабкі колізії: можна створити два повідомлення, які будуть мати однаковий хеш. Це, як правило, більш проста проблема, ніж так звана атака «знаходження прообразу» , де вхідне значення обчислюється на основі хешів до даного параметру.

Пароль можна розглядати як *n* символів (з можливим повторенням) з «набору символів» с., тому існує n^c можливих паролів. В середньому потрібна буде лише їхня половина [3]. Збільшення або *n* або *c* підніме число комбінацій в геометричній прогресії, як показано нижче:

Табл. 1 Залежність кількості комбінацій від n та c

c	*n*	$(N^c)/2$
26	6	154,457,888
26	7	4,015,905,088
26	8	9,885,304,832
52	7	104,413,532,288
52	8	514,035,851,264
62	8	109,170,052,792,448

Попередньо обраховані атаки, такі як веселкова таблиця, можуть здійснюватися проти деяких алгоритмів хешування. Багато додатків, наприклад, Unix-подібні операційні системи, використовують хеш з сілью (англ. salt): згенеровані випадковим чином значення, які не потрібно ховати, в поєднанні з паролем під час хешування, що збільшує кількість необхідного простору і часу, потрібного для генерації таблиці. Добре розроблений алгоритм засолу (англ. salting) може зробити перебір єдиним ефективним засобом відновлення паролю.

Швидкість перебору в 40 мільйонів хешів в секунду – цілком можливий результат для MD5 шифрування на помірно швидкій багатоядерній системі, проте використання перебору на важкому паролі займе дуже багато часу. Для звичайного пароля з 8 символів, вибраних з символів A-Za-Z0-9, ми маємо c = 62 і *n* = 8. Це займе в середньому 2,729,251 секунд, або трохи більше місяця, щоб відновити один хеш. Деякі системи, такі як MD5crypt або GPG шифрування файлів ОС Linux / FreeBSD, виконують кілька ітерацій хеш-функції (зазвичай> 1000), щоб подолати можливість перебору - потенційно збільшуючи час відновлення для гіпотетичного паролю до 83 років [3].

Цю проблему можна розподілити шляхом розбиття простору пошуку між декількома системами. У теорії, лінійне масштабування може бути реалізоване через повну відсутність залежностей між блоками простору пошуку: 31 еквівалентний комп›ютер зможуть відновити MD5-хеш за один день, і шляхом додавання додаткових систем (або використання прискорення GPU), час відновлення потенційно може бути зменшено до декількох годин [4].

4. ОПТИМІЗАЦІЯ АЛГОРИТМУ ДЛЯ ВИКОРИСТАННЯ З GPU

4.1.Інструкція

Зниження загальної кількості інструкцій у процедурі обробки завжди призводить до більш високої продуктивності. AMD GPU забезпечує деякі інструкції, що можуть бути використані для прискорення SHA1 реалізації.

Одним з них є інструкція BFI_INT. Три макро-функції в блоці коду нижче генерують один і той же результат. F1 є основним , виразом, що займає 4 інструкції.F2 також вираз, яке зменшує кількість інструкцій на 1. F3 використовує біт вибору функції в OpenCL, яка використовує інструкцію BFI_INT представлену AMD GPU і при цьому займає всього 1 інструкцію.

```
#define F1(b,c,d) ((b&c)|(~b)&d)
#define F2(b,c,d) (d^(b&(c^d)))
#define F3(b,c,d) bitselect(d, c, b)
```

Ще одна інструкція що може бути використана, є rotate(). Хоча x86 підтримує rol(rotate left) інструкцію, C і багато інших мов програмування високого рівня не підтримують оператор rotate. Тому, як правило, на мові C реалізація rotate left 32-бітового цілого числа може бути виражена як (1) в наступному блоці коду.

```
#define ROTL(x, n) ((x<<n)|(x>>(32-n)))
#define ROTL(x, n) rotate(x, n)
```

4. 2. Модифікація алгоритму

Так як SHA1 і MD5 мають чотири або п›ять різних нелінійних функцій для використання під час обробки одного блоку, він може очікувати декілька if застосувань для випадку, коли потрібно вибрати іншу не лінійну функції відносно декількох кроків, як показано в наступному прикладі псевдокоду.

```
//Головний цикл для  SHA1
for i from 0 to 79
if 0 ≤ i ≤ 19 then
f = (b and c) or ((not b) and d)
k = 0x5A827999
else if 20 ≤ i ≤ 39
f = b xor c xor d
k = 0x6ED9EBA1
else if 40 ≤ i ≤ 59
f = (b and c) or (b and d) or (c and d)
k = 0x8F1BBCDC
else if 60 ≤ i ≤ 79
f = b xor c xor d
k = 0xCA62C1D6
temp = (a leftrotate 5) + f + e + k + w[i]
e = d
d = c
```

```
c = b lefrotate 30
b = a
a = temp
end for
```

Проте використання GPU неефективне для обробки перемикачів, оскільки вони не залежать від значень під час виконання, вони можуть бути усунені за допомогою ручного розгортання основного циклу.

Іншим, що можна оптимізувати є значення регістру свопу. Як було показано в коді вище, в кожній ітерації значення A, B, C, D, E (SHA1) мінялися місцями. Замість заміни значення регістра, використовуючи п›ять 32-бітних інструкцій прив'язки, простою зміною регістру в коді можна досягти того ж ефекту.

Для SHA1, розгорнута версія є не найкращим варіантом для використання з GPU. Основна проблема полягає в різноманітності слів. Попередньо обчислюваний масив слів, який містить 80 32-розрядних цілі числа, має зберігатися в регістрах для підвищення продуктивності. Проте, регістри є обмеженим ресурсом для GPU. Використання такої кількості регістрів призведе до низької зайнятості і продуктивності. Таким чином, альтернативний метод обчислення SHA-1, запропонований в [5], має використовуватися в GPU. Замість того щоб використовувати 80 регістрів, слова, що мають від 16 до 79 символів обчислюються під час роботи, щоб зберегти 64 регістри, як показано в наступній дій.

```
#define W(i) \
( \
s = i & 0x0f,  \
W[s] = W[(s+13)&0x0f] ^ W[(s+8)&0x0f] ^
W[(s+2)&0x0f] ^ W[(s)&0x0f],\
W[s] = rotate(W[s], 1) \
)
```

5.ВИСНОВОК

В цій роботі було виконано аналіз літератури, що описує використання гібридної системі CPU/GPU.Ця система довела свою корисність для паралельних обчислень, так як гібридні обчислення є дуже важливими для криптографії та криптоаналізі.

Ці системи можуть бути швидко масштабованими і гарно використовують надані ресурси, тому їхній подальший розвиток є дуже важливим.

ЛІТЕРАТУРА

1. Karbowski, A. and Niewiadomska-Szynkiewicz, E. Parallel and distributed computing. Warsaw University of Technology Press,2009. – 458 c.

2. Kunzman, D. M. and Kalé, L. V. . Programming hetero-geneous clusters with accelerators using object-based programming, Scientific Programming Volume 19 (2011), Issue 1, 2011, c.47-62

3. Scott Hauck and André DeHon. Reconfigurable Computing, 1st Edition,2007,c. 944

4. Weidong Qiu, Zheng Gong, GPU-Based High Performance Password Recovery Technique for Hash Functions,2016

5. N. I. of Standards and Technology, "Secure hash standard (SHS)," http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf, 2012.

Рецензент: к.т.н., доц. каф. ОТ НТУУ «КПІ» А. М. Волокита

Face Outline Alignment via Constrained Average Displacement

Moroz I.D.
Student of the National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, Kiev

Dorogyy Y.Y.
Cand. Sc. (Eng.), Assoc. Prof, National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, Kiev

Abstract

Over the last ten years deformable model fitting has been gained popularity in the computer society. Thus various methods were introduced with varying degrees of success. This article offers optimization strategy that based on nonparametric distribution of the landmark. Updated equation slightly reminds mean shifts method but with a subspace constraint placed on the shape's variability. This method is shown to outperform common approaches on the task of generic face fitting.

Keywords: dynamic models, deformable, shape alignment

Определение контуров лица методом ограниченного среднего сдвига

Мороз И. Д.
студент Национального Технического Университета Украины «Киевский Политехнический Институт»
Украина, Киев

Дорогой Я. Ю.
Кандидат технических наук, доцент, Национальный Технический Университет Украины «Киевский Политехнический Институт»
Украина, Киев

Аннотация

За последние десять лет динамические модели определения контура лица обрели популярность в компьютерном сообществе. Таким образом были представлены различные подходы с разной степенью успеха. В данной статье предлагается стратегия оптимизации, основанная на непараметрическом распределении вокруг ключевых точек. Этот подход основан на методе среднего сдвига, но имеет ограничением по перемещению. Представленный метод опережает широко известные подходы определения контура лица.

ВВЕДЕНИЕ

Определение динамической модели лица – это проблема проецирования параметризированной математической модели на изображение в необходимом месте, то есть в области лица. Это нелегкая проблема, которая требует работы с 3D-проекцией, где часто бывает затруднительно определить область лица и его контуры из-за изменений в освещении, шума изображения, разного угла наклона лица, плохого качества изображения и т.д.

В этой статье были рассмотрены несколько популярных стратегий оптимизации и введем один новый.

ПОСТАНОВКА ЗАДАЧИ

В последние годы было предложено несколько подходов для определения модели лица. Их можно разделить по принципу построения модели – построение по ключевым точкам либо построение целостной модели. Первый принцип значительно точнее[1], а также требует меньшую вычислительную мощность, но но требует стратегий оптимизации из-за неоднозначности обнаружения и накладывания патчей друг на друга. Основная цель работы состоит в понимании того, что существующие методы оптимизации в целом упрощают непараметрическое распределение вокруг ключевых точек, а также рассмотрении более эффективного метода оптимизации – непараметрического.

МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ

Рассмотрим популярные методы оптимизации для моделей, построенных по ключевым точкам, где модель представляется на основе вспомогательных участков (патчей), которые привязаны к точкам [2, 5, 16]. Этот подход более продуктивен, чем целостное представление модели [10, 11].

В свою очередь для моделей, построенных по ключевым точкам, существует два способа оптимизации: параметризированный (например Active Shape Models, Convex Quadratic Fitting [3]), где пределы возможных отклонений заранее предопределены, и непараметризированный (например Mean Shifts Method), где пределы определяются итеративно. Пример последний был представлен в работе и основан на алгоритме контролируемого среднего сдвига.

ЛОКАЛЬНЫЕ МОДЕЛИ С ОГРАНИЧЕНИЕМ

Вспомним категории геометрических преобразований – бывают линейные (жесткие) и эластичные (нежесткие) преобразования. Первые включают в себя поворот, перемещение, отображение и прочие аффинные преобразования.

Линейные преобразования носят глобальный характер, и их нельзя использовать для моделирования локальных геометрических разностей у изображения. Вторая же категория позволяет проводить локальные деформации, в т.ч. базисные функции и сплайны поверхностей.

Большинство способов накладывания модели используют линейную аппроксимацию при изменения формы объекта вокруг ключевой точки и образуют модель распределения точек (point distribution model – PDM [2]), которая моделирует локальное нежесткое преобразование и сопоставляет его с глобальным жестким, помещая полученный контур на исходное изображение:

$$x_i = sR\left(\overline{x}_i + \Phi_i q\right) + t, \qquad (1)$$

где x_i определяет 2D-координаты PDM для *i*-й ключевой точки.

Зададим $p = \{s, R, t, q\}$ как множество параметров PDM, где s – глобальное масштабирование, R – поворот, t – перемещение и q – нежесткие параметры. $\overline{x}_i$ представляет среднее значение 2D-координат ключевой точки PDM $\left(\overline{x}_1 = \left[\overline{x}_i; \overline{y}_i\right]\right)$, а Φ_i —базисная матрица, относящаяся к ключевой точке.

В последние годы подход, использующий локальные определители (ключевые точки) стал популярен [2, 3, 4, 16], так как нивелирует многие недостатки целостного метода моделирования, таких как сложность построения целостной модели и чувствительность к смене освещения. Эти подходы используют метод моделей с локальным

ограничением (constrained local models, далее – CLM).

Все CLM преследуют две цели: выполнение исчерпывающего поиска для каждой PDM вокруг ключевой точки используя специальный детектор и оптимизация параметров p PDM из-за обильного накладывания патчей друг на друга. На рисунке 1 изображен пример работы CLM в общем виде.

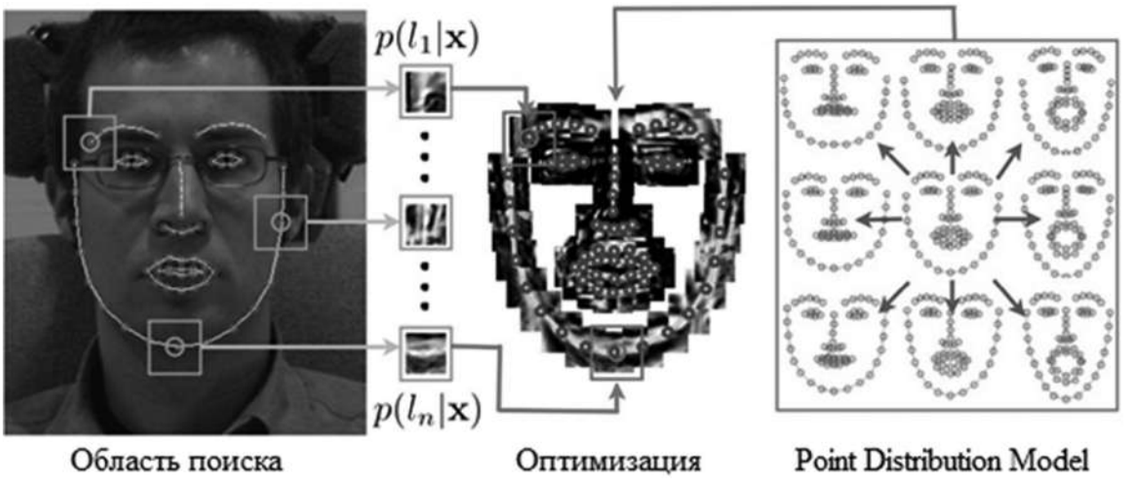


Рисунок 1 – Использование CLM

ИСЧЕРПЫВАЮЩИЙ ЛОКАЛЬНЫЙ ПОИСК

Первый шаг использования CLM состоит в построении матрицы вероятностей для каждой ключевой точки с помощью локальных детекторов для ограниченного пространства вокруг самой точки.

Один из простейших детекторов – линейный логический регрессор [12], который выдает такую матрицу вероятностей для *i-й* ключевой точки. Определим плотность вероятности p*:*

$$p(l_i = aligned \mid I, x) = \frac{1}{1+\exp\{\alpha C_i(I;x)+\beta\}}, \quad (2)$$

где l_i – случайная составляющая, которая определяет, в правильном ли месте находится ключевая точка (aligned), I – исходное изображение, x – координаты, C_i – линейный классификатор:

$$C_i(I;x) = w_i^T\left[I(y_1);...;I(y_m)\right]+b_i, \quad (3)$$

где w_i – весовая функция, b_i – смещение, а $\{y_i\}_{i=1}^m \in \Omega_x$, то есть входит в область патча вокруг ключевой точки.

Оптимизация. После применения детектора для каждой локальной точки (а их порядка 72) и с учетом условной независимости проведём оптимизацию методом максимального правдоподобия относительно параметров p:

$$p\left(\{l_i = aligned\}_{i=1}^m \mid \mathrm{p}\right) = \prod_{y=1}^{n} p\left(l_i = aligned \mid x_i\right), \quad (4)$$

где x_i параметризирован согласно уравнению (1).

При такой оптимизации следует избегать локально оптимального значения функции для достижения общей эффективности функции. Используя общее уравнение (4) для оптимизации можно достичь приемлемого результата при хорошем качестве исходных изображений. Но, так как результаты шумные, то такие стратегии оптимизации обычно нестабильные. Поэтому рассмотрим усовершенствованный способ оптимизации.

ОПТИМИЗАЦИЯ С ОГРАНИЧЕНИЕМ СРЕДНЕГО СДВИГА

При оптимизации вспомогательный участок $\{p(l_i \mid x)\}_{i=1}^n$ заменяют на более простую параметрическую или непараметрическую форму и оптимизируют уже её.

Один из простейших способов оптимизации в методе Active Shape Models [2]. Этот метод сперва предполагает поиск координат в пределах вспомогательные участки, где были присвоены локальные максимумы: $\mu = [\mu_i;...;\mu_n]$. Цель этой процедуры заключается в минимизации методом взвешенных наименьших квадратов разницы между PDM и координат peak response [11].

$$Q(\mathrm{p}) = \sum_{i=1}^{n} w_i \left\| x_i - \mu_i \right\|^2, \quad (5)$$

Уравнение (5) итеративно минимизируется при помощи разложения по Тейлору первого порядка:

$$x_i \approx x_i^c + J_i \Delta p, \quad (6)$$

Но такая простая оптимизация часто не даёт желаемых результатов из-за того, что peak response (максимальное значение) не всегда совпадает с координатами ключевой точки.

Для решения этой проблемы был предложен метод выпуклого квадратического контура (convex quadratic fitting – CQF), который эквивалентен ковариационному распределению [12]:

$$p\left(l_i = aligned \mid I, x\right) \approx N\left(x; \mu_i; \Sigma_i\right), \quad (7)$$

Среднее значение и ковариация являются оценками максимального правдоподобия для вспомогательных участков:

$$\Sigma_i = \Sigma_{x\in\Psi_{x_i^c}} \alpha_x^i \left(x-\mu_i\right)\left(x-\mu_i\right)^T = \Sigma_{x\in\Psi_{x_i^c}} \alpha_x^i x, \quad (8)$$

где $\Psi_{x_i^c}$ – прямоугольная рамка с центром в текущей ключевой точке x_i^c;

α_x^i – нормализированный определитель [12]:

$$\alpha_x^i = \frac{p\left(l_i = aligned \mid I; x\right)}{\sum_{y\in\Psi_{x_i^c}} p\left(l_i = aligned \mid I; y\right)}, \quad (9)$$

Вместо использования аппроксимации для каждого патча с использованием параметрической модели, рассмотрим использование непараметрического представления из-за большей точности метода. Используем ядерную оценку плотности распределения (kernel density estimate, далее – KDE) с изотропным ядром Гаусса [10]:

$$p\left(l_i = aligned \mid I; x\right) \approx \Sigma_{\mu_i\Psi_{x_i^c}} \alpha_{\mu_i}^i N\left(x; \mu_i; \sigma^2\mathbf{I}\right), \quad (10)$$

На рисунке 2 изображены вспомогательные участки для трёх точек и их аппроксимации с использованием разных способов оптимизации. Аппроксимация с KDE показана для σ2 ∈ *{20, 5, 1}*.

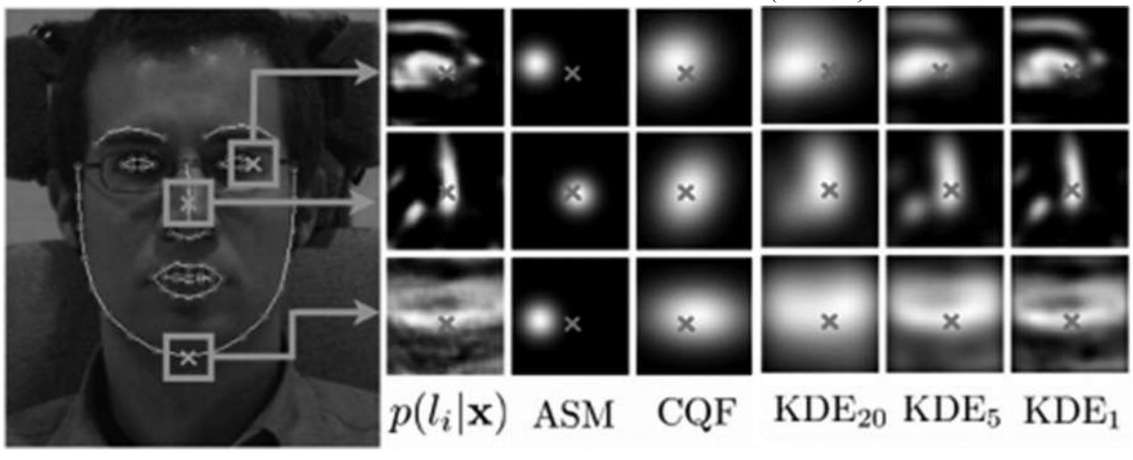


Рисунок 2 – Использование патчей

Способ максимизации с KDE рассмотрен в алгоритме среднего сдвига [1]:

$$x_i^{(\tau+1)} \leftarrow \Sigma_{\mu_i\in\Psi_{x_i^c}} \frac{\alpha_{\mu_i}^i N\left(x_i^{(\tau)}; \mu_i; \sigma^2\mathbf{I}\right)}{\sum_{y\in\Psi_{x_i^c}} \alpha_y^i N\left(x_i^{(\tau)}; y; \sigma^2\mathbf{I}\right)} \mu_i, \quad (11)$$

где – период итеративного процесса, который проходит, пока не удовлетворен критерий сходимости [6].

При использовании линейной модели в уравнении (6) и максимизации уравнения (4) *Q*-функция шага M принимает вид:

$$\Delta p = \mathbf{J}^+ \left[x_1^{(\tau+1)} - x_i^c; \ldots; x_n^{(\tau+1)} - x_n^c \right], \quad (12)$$

где $\mathbf{J}^+$ - псевдообратное **J**, а средний итеративный сдвиг [12].

Используем полученные формулы для алгоритма ограниченного среднего сдвига.

Псевдокод алгоритма ограниченного среднего сдвига

Требуется: *I*, *p* {изображение, набор параметров}
1. **while** not_converged(p) **do**
2. Вычисление вспомогательных областей {Ур. (2)}
3. Построение линейная модели {Ур. (6)}
4. Вычисление псевдообратного Якобиана (J†)
5. Инициализация обновления параметров: Δp ← 0
6 **while** not_converged(Δp) **do**
7. Вычисление среднего сдвига {Ур. (11)}
8. Применения ограничения {Ур. (12)}
9. **end while**
10. Обновить параметры: p ← p + Δp
11. **end while**
12. return p

СРАВНЕНИЕ АЛГОРИТМОВ

Сравним работу рассмотренных методов на открытой базе лиц XM2VTS [9], которая состоит из 2360 фото 295 особей в различных позах и выражениях лица. Результаты эксперимента отображены на рисунке 3, где показано количество изображений, на которых было обнаружено возмущение, определяемое как среднеквадратическое отклонение.

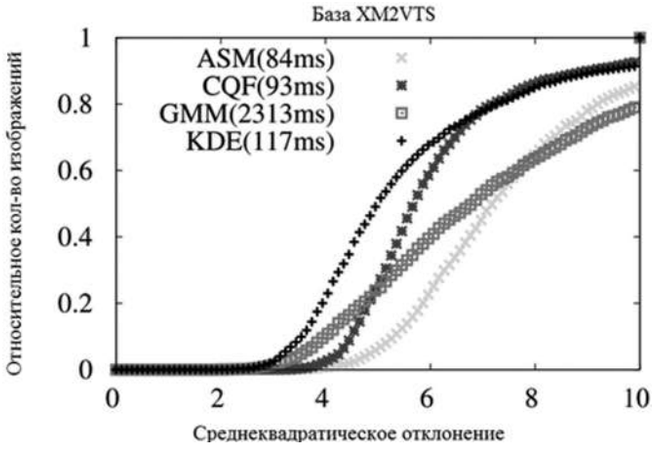


Рисунок 3 – График зависимостей для базы лиц XM2VTS

ИСПОЛЬЗОВАНИЕ ВНЕ БАЗЫ ЛИЦ

Метод ограниченного сдвига с использованием KDE хорошо показал себя при определении контура лица даже с частичной преградой.

На рисунке 4 показано определение контуров лица в стандартных условиях, при частичной преграде и при резкой смене положения.

ЗАКЛЮЧЕНИЕ

В этой работе были рассмотрены существующие методы оптимизации для динамических моделей, построенных по ключевым точкам, а также был представлен один из способов непараметрической оптимизации. Сравнительный график показал, что указанный метод превосходит классические методы оптимизации, а также превосходно показал себя на тестовых испытаниях. В дальнейшем планируется изучить влияние разных типов локальных детекторов и форму модели по умолчанию для улучшения общей стратегии алгоритма оптимизации и улучшения полученных результатов.

Рисунок 4 – Использование алгоритма в разных условиях

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

[1] Y. Cheng. Mean Shift, Mode Seeking, and Clustering, с. 790–799, 1995.

[2] T. F. Cootes and C. J. Taylor. Active Shape Models - ‘Smart Snakes’, с. 266–275, 1992.

[3] D. Cristinacce and T. Cootes. Boosted Active Shape Models, с. 880–889, 2007.

[4] D. Cristinacce and T. F. Cootes. A Comparison of Shape Constrained Facial Feature Detectors. Сборник FG, с. 375–380, 2004.

[5] D. Cristinacce and T. F. Cootes. Feature Detection and Tracking with Constrained Local Models, с. 929–938, 2004.

[6] M. Fashing and C. Tomasi. Mean Shift as a Bound Optimization. Сборник PAMI, 27(3), 2005.

[7] L. Gu and T. Kanade. A Generative Shape Regularization Model for Robust Face Alignment. ECCV’08, 2008.

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Re- port 07-49, University of Massachusetts, Amherst, 2007.

[9] K. Messer, J. Matas, J. Kittler, J. Lu¨ttin, and G. Maitre. XM2VTSDB: The Extended M2VTS Database. Сборник AVBPA, с. 72–77, 1999.

[10] M. A. C.-P. na´n and C. K. I. Williams. On the Number of Modes of a Gaussian Mixture. Lecture Notes in Computer Science, 2695:625–640, 2003.

[11] M. H. Nguyen and F. De la Torre Frade. Local Minima Free Parameterized Appearance Models. CVPR, 2008.

[12] Y. Wang, S. Lucey, and J. Cohn. Enforcing Convexity for Im- proved Alignment with Constrained Local Models. PR, 2008.

Exploring The Usage Of Folksonomies As A Recommender Tool In Social Tagging Systems

Artur Dzidzoiev
student, NTUU "KPI" FICT
Kyiv, Ukraine

Annotation

In this paper author has described main approaches to development recommender systems based on social tagging system, defined the role of folksonomies as a means of building recommender models and enlisted main development issues which may encounter while development such systems: tag sparsity, social network divide, tag idiosyncrasy. Author has proposed the solution for problems depicted above.

Key words: folksonomy, social tagging systems, recommender systems, tag sparsity, social network divide, tag idiosyncrasy

Дослідження використання фолксономій як інструмента для побудови рекомендаційних моделей в системах соціального тегування

Дзідзоєв Артур Юрійович
студент, НТУУ "КПІ" ФІОТ
Київ, Україна

Анотація

В цій роботі розглядаються основні підходи до розробки рекомендаційних систем на основі систем соціального тегування, визначається роль фолксономій як інструмента при побудові рекомендаційних моделей, а також розкриваються основні складності, які виникають при розробці таких систем, а саме розрідженість тегів, соціально-мережевий бар'єр та ідіосинкразія тегів. Автором було запропоновано рішення зазначених вище проблем.

Ключові слова: фолксономія, системи соціального тегування, рекомендаційні системи, розрідженість тегів, соціально-мережевий бар'єр, ідіосинкразія тегів

За останнє десятиліття Всесвітня Мережа (Web) зазнала зміну парадигм: від джерела інформації до платформи соціальної взаємодії, де користувачі можуть легко контактувати між собою, вільно завантажувати та поширювати контент[7], а також анотувати цей контент довільно вибраними ключовими словами - тегами[2], формуючи так звані фолксономії. Цей процес сприяє децентралізації суб'єктів постачанню контенту та його оцінці, тобто Мережа стає автономною від певних організацій чи людей, кожен учасник може як постачати нову інформацію, так і вільно її характеризувати[5]. Проте невпинне зростання інформації викликає певні складності у використанні наявних інформаційних ресурсів. Користувачі мережі більше не здатні опрацьовувати всю наявну в ній інформацію. Для того, щоб дати можливість користувачу задовільняти свою інформаційну потребу, створюються системи, які допагають виділити з великого різноманіття інформації таку, що буде корисна користувачу. Такими системами є пошукові системи, рекомендаційні системи (РС), системи соціального тегування(ССТ) та інші.

Призначення рекомендаційних систем полягає в тому, щоб вирішити проблему інформаційного перевантаження шляхом прогнозування відповідних ресурсів для користувача. Як правило, рекомендаційна система складається з рекомендаційного рушію, який з урахуванням вхідних даних створює список рекомендованих ресурсів, відсортованих за коефіцієнтом релевантності. Коефіцієнт релевантності – це оцінка, яка надається ресурсу за шкалою відповідності до інформаційної потреби користувача[1].

Призначення РС полягає в тому, щоб вирішити проблему інформаційного перевантаження шляхом прогнозування відповідних ресурсів для користувача[1]. В ССТ користувачі можуть додавати різноманітні ресурси, веб-сторінки, публікації, малюнки чи музичні треки та анотувати їх довільно вибраними тегами (рис. 1). В той час, як в більшості з цих систем основною задачею тегів є допомогти конкретним користувачам огранізовувати їхній власний контент, основна ідея тегування полягає також у тому, що теги повинні допомогти іншим користувачам знаходити, категоризувати та переглядати ресурси.

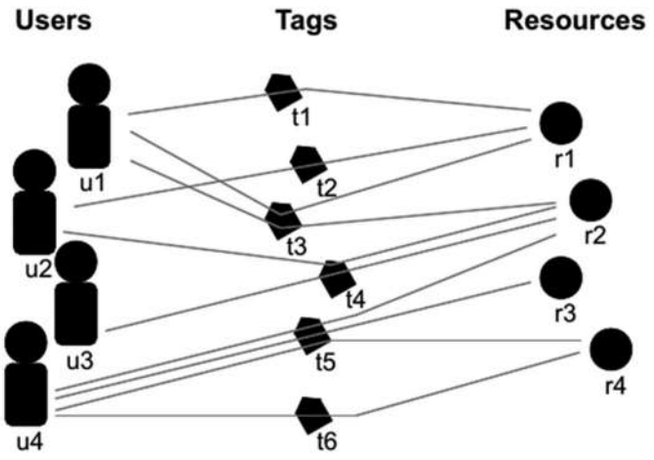


Рис.1 - Соціальне тегування: Користувачі - Теги - Ресурси

Створюючи рекомендаційну систему туристичних об'єктів ми прагнули до максимальної автономії роботи від сторонньої модерації та адміністування. Користувачі мали би змогу самі додавати об'єкти, вносити колаборативні поправки до існуючих об'єктів а також явно та неявно брати участь в їхньому анотуванні. В системі на перше місце було поставлено можливість користувачам помічати існуючі об'єктам так званими ключовими словами - тегами. Наприклад, користувач публікує нове місце зі своєї останньої поїздки до Канева, анотуючи його ключовими поняттями як "кручі", "Дніпро" та "історичне місце". Ці теги допоможуть користувачу в майбутньому знайти раніше відмічені місця, а отже складатимуть його персональномий інформаційномий менеджемнт. Окрім того, інші користувачі зможуть знаходити це місце, якщо вони будуть шукати по цьому конкретному тегу в системі.

Для визначених вище умов було вирішено розробляти рекомендаційну систему на основі системи соціального тегування. Аргументи для сикористання ССТ дуже прості: в той час як при класичній розробці будь-якої системи необхідне дослідження предметної області і інжиніринг, які, як правило, коштують дорого і забирають багато часу через необхідність наймати експертів в предметній області і інженерів для моделювання домену та анотації контенту, в ССТ велика кількість користувачів може робити частину роботи безкоштовно, а саме класифікацію та антотування контенту . Але, незважаючи на привабливий характер ССТ, існують такі проблеми[1] що обмежують реалізацію її повниого потенціалу:

РОЗРІДЖЕНІСТЬ ТЕГІВ

Когнітивне зусилля для підбору хорошого набору тегів в кінцевому підсумку може демотивувати користувачів присвоювати теги, піднімаючи проблему розрідженості тегів: в системі буде анотовано лише малу частину завантажених ресурсів. Зауважимо, що користувачів не можна змусити призначати теги, отже, необхідно розглянути доречні стимули для тегування.

СОЦІАЛЬНО-МЕРЕЖЕВИЙ БАР'ЄР

Соціальні мережі, як правило, поділені на групи по інтересах, інакше кажучи спільноти видів діяльності. Спільнотами видів діяльності називають групи людей, які розділяють інтерес до чогось, чим вони займаються, і вчаться робити це краще, оскільки вони регулярно взаємодіють між собою [5]. Враховуючи вже велику і дедалі зростаючу кількість різноманітних соціальних мереж, буде легко знайти сайти, аудиторія яких "перетинається" в аспекті інтересів. Хоча така надмірність має сприяти можливості взаємодії для тих, хто бере участь в соціально-мережевій екосистемі, в результаті це випливає в соціально-мережевий бар'єр: користувачі з різних соціальних мереж, які зацікавлені в подібних темах, не можуть ділитися своїми тегами/ресурсами між собою через те, що системи, в яких вони беруть участь, не поєднані між собою.

ІДІОСИНКРАЗІЯ ТЕГІВ

Бажано, щоб теги окремого користувача також використовувались іншими користувачами. Це сприятиме поширенню та пошуку контенту. Проте ССТ не вимагають від учасників спільного використання тегів, а використовують це як побічний ефект системи, за умови, якщо тег стає популярним та інші учасники його помічають. Зауважимо, чим більш доступно і вміло узгоджені теги, тим вища вірогідність, що вони будуть поширюватися. Проте користувачі часто обирають такі теги, які мають значення тільки для них самих, що створює проблему ідіосинкразії тегів. Для підвищення здатності до поширення тегів серед користувачів необхідно забезпечити добре узгоджену семантику для тегів, проте за умови збереження для користувачів свободу тегування.

Отже, коли сформовано конкретні проблеми, що описані вище, визначимо наступні задачі дослідження:

- Тернарні реляційні дані (рис. 2). На відміну від типових рекомендаційних систем, в яких між користувачами і об'єктами існує тільки бінарний зв'язок, дані соціального тегування утворюють тернарний зв'язок між користувачами, ресурсами і тегами. Таким чином, ми хочемо відповісти на наступне питання: *як вирішувати проблеми, пов'язані з тернарними реляційними даними ССТ для того, щоб розробити ефективні рекомендатори тегів?*

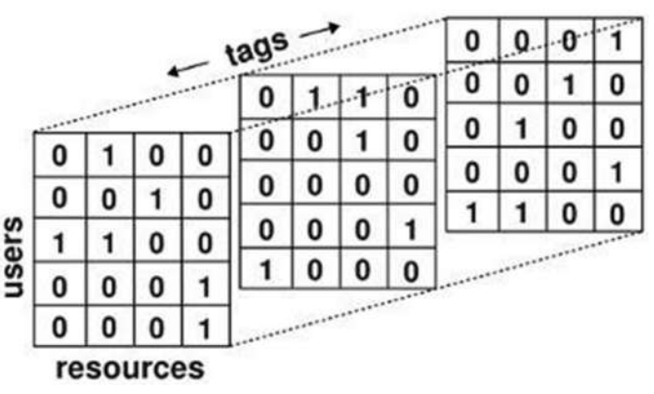


Рис. 2 - Тернарні реляційні дані

• Семантична відповідність. Як було зазначено раніше, неконтрольований словник користувачів ССТ може викликати небажані проблеми, що в кінцевому підсумку ставить під загрозу поширення та пошук контенту. Просто змусити користувачів використовувати контрольований професійний галузевий словник не є слушною думкою, оскільки це може погіршити проблему розрідженості тегів, тобто контрольовані словники, як правило, містять дуже спеціалізовані терміни, про які користувач може не бути обізнаним. Проблема полягає в наступному: *як досягти відповідності між словниками експертів в галузі та користувачів ССТ таким чином, щоб семантика, призначена користувачами, була явно зрозумілою?*

• Оцінювання. Хоча існують стандартні процедури оцінювання для традиційних рекомендаційних систем, які можна просто перенести на сценарій рекомендаторів тегів, наступний предмет дослідження залишається: *як кількісно оцінити ефективність підходів для поєднання соціальних мереж або подолання проблеми ідіосинкразії тегів?*

ВИКОРИСТАНІ ДЖЕРЕЛА:

1. Leandro Balby Marinho (2009). Recommender Systems for Social Tagging Systems, University of Hildesheim

2. Angeletou, S., Sabou, M. & Motta, E. (2009). Improving search in folksonomies: A task based comparison of wordnet and ontologies. In K-CAP ’09: Proceedings of the 5th International Conference on Knowledge Capture, 169–170, ACM. 105

3. Cattuto, C., Loreto, V. & Pietronero, L. (2007). Semiotic dynamics and collaborative tagging. PNAS , 104, 1461–1464. 17, 74

4. Guan, Z., Bu, J., Mei, Q., Chen, C. & Wang, C. (2009). Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In SIGIR’09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 540–547, ACM. 50, 51

5. Lipczak, M., Hu, Y., Kollet, Y. & Milios, E. (2009). Tag sources for recommendation in collaborative tagging systems. In DC ’09: Proceedings of the ECML/PKDD Discovery Challenge, vol. 497 of CEUR-WS.org. 50, 90

6. T. O’Reily. What is web 2.0? - design patterns and business models for the next generation of software, September 2005.

7. X. Li, L. Guo, and Y. E. Zhao. Tag-based social interest discovery. In Proc. of the 17th Int. World Wide Web Conference (WWW’08), pages 675–684. ACM Press,2008.

8. E. Rich. Users are individuals: individualizing user models. International Journal of Man-Machine Studies, 18(3):199 – 214, 1983.

Рецензент: к.т.н. ст. викл. каф. ТК НТУУ «КПІ» Е.П. Сирота

Access model based on mobile agents for the protection of cloud computing

Vu Duc Thinh
graduate graduate school NTUU “KPI”,
Candidate of Engineering Sciences
Ukraine, Kyiv

A.Volokyta
NTUU “KPI”, Candidate of
Engineering Sciences, Docent
Ukraine, Kyiv

P. Rehida
NTUU “KPI” graduate student
Ukraine, Kyiv

Abstract: This paper addresses the issue of information’s security in cloud systems. Some modern solutions of protection are presented. These solutions have capabilities. Capabilities were compared, and result is shown in a corresponding table. The model, which supports all protection options from that table is presented. Multiagent system and method of data access that based on a modified five-dimensional Hartson space. An architecture of wandering mobile agent is presented.

Keywords: mobile agent, cloud security

Clouds provide three types of resources: a repository of images of virtual machines, a set of computer servers on which they can be launched and an array of data repositories. There are solutions that provide security support. The following table present some of these solutions and their task of security support. Rows presents tasks of security support; columns present these solutions.

Table 1 Security Tasks Coverage

Tasks	***Storage of images of virtual machines***	***Privacy Manager***	***Interim protection system of cloud computing***
Infrastructure security	+	-	+
Data security	+	+	+
Identity management and access control	+	+	-
Security Management	+	+	-
Privacy	+	+	-
Audit	+	+	+
Security Service	-	+	-

Infrastructure Security[1] - security at various levels of “cloud”. They are: network layer, host platform layer, and application level. The table [2,3,4] shows that described solutions

do not provide full coverage of tasks of security support in cloud computing. To solve this problem, we can do integration of these solutions in a single model (Fig. 1). All information about resource usage is kept in monitoring unit. The user has access to the monitoring data if user and these data are related. Privacy Manager provides security of interaction between user and clouds, by supporting encryption and providing additional capabilities for managing access rights and control data. Support of interaction provided by the cloud system manager. "Access data" block gives access to the user data and provides information on their use. Images of virtual machines are stored in a special storage, which supports version control system. This vault is protected by a system of access control, integrity control and filters. When a guest operating system start running, images of virtual machines loading from storage on the host platform. The monitoring system works all times and all potentially dangerous events are recorded in special journals.

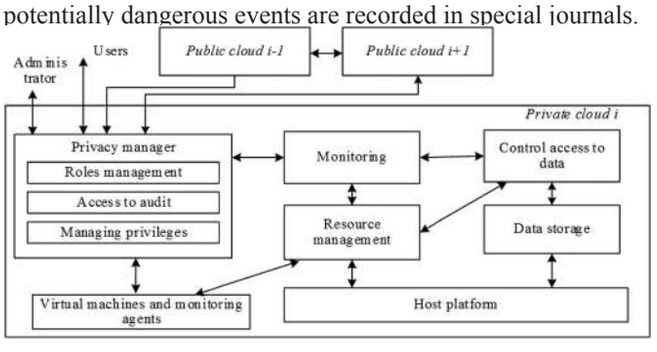


Fig. 1 – Generalized model of cloud security

The issue of security is a key problem in cloud computing. Basic solutions of cloud computing protection and how they work are presented. It was determined that these products do not provide full protection of private clouds. It is proposed single system that consists of a set of subsystems, each subsystem is responsible for a separate area of security.

Multiagent protection system is as follows: $A^s = \left\{ A_M, \left\langle A_{sc}, A_{ssw}, A_{usw}, A_{ac}, A_{net}, A_{db}, A_{dev} \right\rangle \right\}$, де: A_M – monitoring agent-coordinator; A_{sc} – agent monitoring of system components; A_{ssw} – agent monitoring systems software; A_{usw} – agent monitoring users software; A_{ac} – agent monitoring of access; A_{net} – agent monitoring of network connection; A_{db} – agent monitoring of database; A_{dev} – agent monitoring of external devices.

Model of discretionary access: To create a model of discretionary access we will use modified five-dimensional space Hartson[5]. We need to extend area of security. Final area will consist of six sets. It includes set of agents. *{I}* (Fig. 2):

- Set of users *U*;
- Set of resources *R*;
- Set of states *S*
- Set of powers *A*;
- Set of operations *E*;
- Set of hierarchical agents *I*;

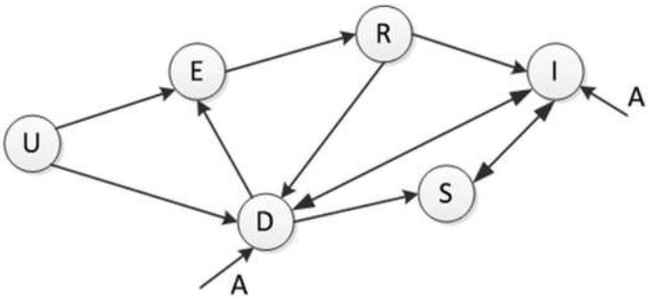


Fig 2 – Modified Hartson's space

Then security area can be represented as Cartesian multiplication:

$$A \times U \times E \times R \times S \times I$$

Users make a request for access to resources. When the system executes these requests, it enters into a new state. These requests represented as five-level cortege.

$$q = (u, e, R', s, I'); u \in U, e \in E, s \in S, R' \subseteq R, I' \in I.$$

Agent architecture The main components of this architecture are: private cloud users, agents of cloud and manager confidentiality. There are a number of different categories of users in an environment of distributed computing, for example: professors, graduate students, interns. They can use cloud system for secure data transmission, data storage, or making experiments.

Agent can provide secure services. Also, agents exchange messages inside their hierarchy, about changing user data or allocation of resources to the user, it allows to allocate resources within the network [6].

Mobile wandering agent. Mobile agent uses a database of information and authority (DBIA) to manage security of segment of distributed computer system(DCS). DBIA also allows to get access to service information of other agents hierarchy. DBIA consists of tables that containing information about the agent and set of the service information about security and capabilities relevant resources segment of DCS. Monitoring tool integrated into every agent and is used to control the access. This allows for more effective planning the allocation of resources. Each layer agent has several modules that interact with each other for control of powers during data transfer

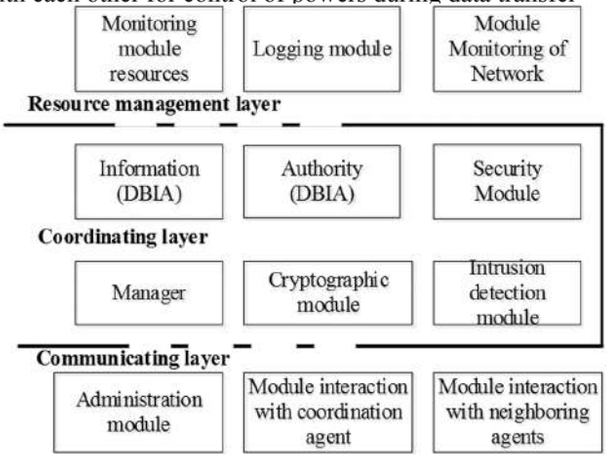


Fig 3 – Wandering agent structure

The communication layer has communication function and acts as the interface to the external environment. The agent uses communication module to receive and to send official communications or to transfer information to the modules of communication layer. Making decisions about how the agent should act when receiving messages formed in coordination layer.

Resource management layer of agent is designed to control the flows of data, distributing and monitoring resources.

Data transfer tasks are sent from the coordinating layer to the local manager of agent. These tasks include scheduling information for the data (start time of the transfer, allocation of data channels, IDs, etc.). Part of the flow control system is also responsible for managing the queues streams, that were scheduled for transfer to locally managed resources. At the time of the transfer, the data aimed at component of resource allocation.

Manager of monitoring in coordination layer compares the data from the activity module with plans of loads (start time of the transfer, allocation of data channels, IDs, etc.). An important component of resource management layer is a module who monitors resources and activities on the node. Resource monitor controls the data flows and resource allocation and collects information about node activity to send these data to the coordinating layer of agent.

This paper deals with the protection of private clouds. An analysis of existing security solutions is done. Table of cover is built, it illustrates the positive and negative aspects of existing solutions. From table of cover is determined that such solutions do not fully protect. To solve this problem, system that consists of subsystems, each of which is responsible for their part of security, is presented. Multiagent system provides adequate protection, a list of agents that this system requires is presented. Proposed to use discretionary access model to ensure data access. This model is based on a modified 5-dimensional Hartson space. Also, this article considers mobile agent architecture, that used to test emergency situations.

BIBLIOGRAPHY:

1. Volokyta A.M. The security maintenance model of cloud computing at the infrastructure level / A.M Volokyta, Vu D.T., I.V Kokhanevych., A.E Bidkov. // Adaptive systems of automatic control. -2012.- №. 21(41)-p. 123-131.

2. Wei Jinpeng. Managing Security of Virtual Machine Images in a Cloud Environment / Wei Jinpeng, Zhang Xiaolan, Ammons Glenn, Bala Vasanth, Ning Peng // CCSW ‘09: Proceedings of the 2009 ACM workshop on Cloud computing security pages 91-96. November 2009.

3. Mowbray Miranda. A Client-Based Privacy Manager for Cloud Computing. / Mowbray Miranda, Pearson Siani // COMSWARE ‘09: Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middleware. June 2009.

4. Lombardi Flavio. Transparent Security for Cloud. / Lombardi Flavio, Di Pietro Roberto // SAC ‘10: Proceedings of the 2010 ACM Symposium on Applied Computing, pages 414-415. March 2010.

5. Volokita A.N. / Hierarchical Security Agents in Distributed Computing Systems // Volokita A.N., Vu Dyk Tkhin // Visnyk NTUU “KPI”. Informatics, operation and computer science. – 2012. - № 55. - p.117-124.

6. Tarasov V. B. From multi-agent systems to intelligent organizations: philosophy, psychology, computer science. / V. B. Tarasov., 2002.

СТАТТІ КОНФЕРЕНЦІЇ

ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

Active network system monitoring with indicators display in real time

Vovk Yevgeniy Andriyovych
Student «ACTS» NTUU «KPI»
Kyiv, Ukraine

Mart Bohdan Anatoliyovych
Postgraduate «ACTS» NTUU «KPI»
Kyiv, Ukraine

During the development of the monitoring and management of IT infrastructure raises task to timely alert administrator on changes of the state of target and about critical situations of the system in real time. Study describes main mechanisms of timely delivery of data to the web interface. We consider a distributed monitoring system based on the agent approach, which optimizes network traffic usage by sending information to the central server not continuously, but only when it is needed to the user, or when the internal buffer is full. Agent uses different modes, for regular collection of metrics of target machine, preserving information in a local buffer and packet sending and real-time mode1, which foresees more frequent surveys, and immediate delivery of all changes. To ensure timely and reliable data sending we are using Enterprise service buss technology, and Web sockets library.

Keywords: real time, Signal-R, monitoring, client sever architecture

Система активного моніторингу мережі, з відображенням показників у реальному часі

Вовк Є. А.
Студент каф. «АУТС» НТУУ «КПІ»
Київ, Україна

Март Б. А.
ас. каф. «АУТС»
Київ, Україна

У ході розробки системи моніторингу та управління ІТ-інфраструктурою необхідне своєчасне оповіщення адміністратора про зміну стану її елементів зі всіма показниками, повідомлення про появу критичних ситуацій в роботі системи. Робота розглядає основні механізми забезпечення своєчасної доставки даних на веб-інтерфейс, без необхідності перезавантаження сторінки. Розглянуто розподілену систему моніторингу, базовану на агентському підході, яка з метою оптимізації мережевого трафіку відсилає інформацію до центрального сервера не постійно, а лише у разі, коли вона необхідна користувачеві, або коли переповнився внутрішній буфер. Запропоновано використовувати різні режими роботи: для штатного збору метрик із цільової машини зі збереженням інформації в локальному буфері та пакетною відправкою і режим збору в реальному часі, що передбачає більш часте опитування та моментальну відправку всіх змін. Для забезпечення своєчасної та надійної відправки даних використовується технологія шини даних і бібліотека веб-сокетів.

Ключові слова: Клієнт-серверна архітектура, моніторинг, SignalR, реактивне відображення

ВСТУП

Показники ефективності роботи великих підприємств безпосередньо залежать від стабільності роботи корпоративної ІТ-інфраструктури. У свою чергу стабільність роботи ІТ-інфраструктури безпосередньо залежить від своєчасних дій адміністраторів, що базуються на основі отриманих параметрів функціонування ІТ-інфраструктури. Це потребує відображення таких параметрів у реальному часі.

ІТ-інфраструктуру доцільно подавати як ієрархічну структуру, на верхньому рівні якої знаходяться функціональні та технологічні підсистеми, на нижньому – апаратні та програмні елементи [1]. Стан кожного елементу визначається за результатами обробки значень власних параметрів, отриманих у процесі моніторингу. Оцінка якості функціонування підсистем або елементів, що містять інші елементи, здійснюється шляхом аналізу станів елементів, що входять до їхнього складу або впливають на їхню роботу, з урахуванням аналізу значень власних параметрів функціонування, отриманих у результаті моніторингу підсистем або елементів. ІТ-інфраструктура надає апаратно-програмні засоби для автоматизації бізнес-процесів, забезпечує інформаційну й телекомунікаційну взаємодію між функціональними системами, підрозділами, співробітниками та ін.

Для управління корпоративними ІТ-інфраструктура-

ми використовуються універсальні фірмові або розроблені власні, адаптовані під потреби підприємства, системи управління IT-інфраструктурою (СУІ) [2]. СУІ повинні виконувати такі функції: відстеження основних показників функціонування обладнання; відстеження стану запущених процесів та сервісів; агрегація даних моніторингу; збереження історії зміни параметрів до БД; відображення інформації про зміни параметрів роботи IT-інфраструктури у вигляді часових діаграм та графіків із можливостями масштабування та вибору часових проміжків; аналіз та оцінка станів параметрів; інформування відповідальних осіб при перевищенні якимось значенням певного порогу; автоматичне формування реакції на зміни станів параметрів із метою забезпечення певного рівня сервісу; відправка команд віддаленого управління на підконтрольне обладнання. Проте в будь-якому разі стійкість IT-інфраструктури і всієї організації залежить від своєчасного отримання показників із кожної робочої станції, наприклад для сервера баз даних, на якому активно йде запис; одним із найважливіших факторів є вільне місце на накопичувачі, і в години найактивнішого використання оператор повинен отримувати поточні показники в реальному часі.

Розроблено модуль оповіщення в реальному часі для СУІ. СУІ базується на клієнт-серверній архітектурі з використанням шини даних (ESB – enterprise service bus) і бази даних для накопичення та зберігання інформації [3]. Основні компоненти СУІ: серверний компонент, що включає базу даних, сервіс для збереження інформації, веб-подання для управління системою; шина даних; агенти.

У базі даних можна виділити ключові таблиці (рис 1).

Таблиця OMMTypes містить опис типів об'єктів моніторингу, таких як обладнання, сервіси, сервери. Кожний OMMType містить від 1 до N параметрів/метрик (OMMParameters). У таблиці OMMs містяться конкретні сутності об'єктів моніторингу. Історія зміни метрик зберігається в таблиці з композитним ключем (за рахунок чого досягається фізичне сортування рядків БД за об'єктом, типом метрики та датою) OMMParameterValues.

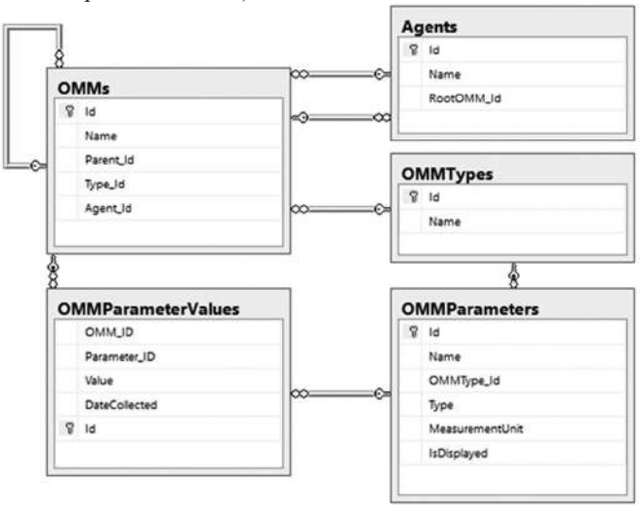


Рисунок 1. ER-Діаграма сутностей моніторингу

Таблиця агентів указує на межу розподілу об'єктів моніторингу між агентами, окрім того, кожен ОМУ містить посилання на його агент, це дозволяє визначати потрібний ОМУ, коли приходить пакет даних з агенту, та визначати відповідний агент по ОМУ з користувацького інтерфейсу.

Серверний компонент розмежований на дві частини, сервіс, як служба ОС «Windows» отримує дані моніторингу з шини даних та зберігає їх до бази даних. Завдяки використанню шини даних є можливість прозоро розподілити сервіси на різних фізичних машинах, і вся робота з балансування пакетів лягає на сервісну шину [3]. Оскільки шина даних являє собою чергу, яка накопичує пакети з даними моніторингу від усіх агентів, – проблемою такого опрацювання даних є забезпечення синхронізації операцій вставки та оновлення даних, – цю проблему вирішено використанням засобу синхронізації потоків на рівні бази даних завдяки застосуванню службової процедури «sp_getapplock», що дозволяє блокувати одночасну зміну даних із розподілених у просторі екземплярів сервісу.

Компонент веб-подання для управління IT-інфраструктурою. Даний компонент відповідає за взаємодію з оператором і реалізує відображення даних моніторингу та інтерфейс управління агентами.

Модуль міжкомпонентної взаємодії відповідає таким вимогам:

- Автономність (легка заміна поточної реалізації на будь-яку іншу з чітким інтерфейсом взаємодії).
- Збереження та накопичування даних при відключенні сервера.
- Можливість винесення на окрему фізичну машину (для зниження навантаження на поточну серверну машину).

З урахуванням вищеперерахованих пунктів було вирішено використовувати шину даних Apache ActiveMQ – через найвищий показник швидкості та легкості впровадження.

Важливим компонентом загальної архітектури є агент [2]. Даний компонент реалізовано за принципом тонкого клієнта, що вміє виконувати команди, надіслані з сервера, та план моніторингу, що включає в себе набір об'єктів моніторингу машини, на якій знаходиться агент, та відправку даних на сервер через вказаний у налаштуваннях інтервал часу. Також агент можна запускати як сервіс чи як звичайний процес.

На агенті реалізовано механізм виконання команд, що отримують або певну кількість параметрів, або жодного, і повертають результат у вигляді JSON-об'єкта. Цей механізм уніфікований і розширюваний, оскільки кількість зареєстрованих команд у системі є динамічно змінюваною величиною, що дає можливість додавати нові функції без зміни всієї системи. Його було спроектовано для виконання віддалених команд, наданих адміністратором, але універсальність даного механізму дозволила використовувати функції для здійснення активного моніторингу, і вважати повернений об'єкт поточним станом відповідного компоненту машини, на яку інстальовано агент. У агенті виділяють такі групи команд:

- реального часу, такі як запуск та відключення механізму оповіщення;
- для роботи з файловою системою створення /

видалення файлів;

- для роботи з процесами запуску / завершення;
- моніторингу – здійснюють активний моніторинг різних показників;
- системна – заміна та отримання конфігурації агенту.

При штатній роботі агент здійснює активний моніторинг, періодично опитуючи команди/функції, згідно з «планом моніторингу», який конфігурується адміністратором; зберігає результати в локальну базу даних, і дані, що змінилися з моменту останнього відправлення, через задані в конфігурації інтервали часу надсилає на сервер із допомогою сервісної шини (рис. 2).

Рисунок 2. Відправлення даних моніторингу

У разі, якщо адміністраторові потрібно віддалено виконати будь-яку з доступних команд, необхідно, використовуючи веб-подання, вибрати агент і команду, після чого система оповістить агент через шину даних, агент виконає команду і поверне результат її виконання (рис. 3).

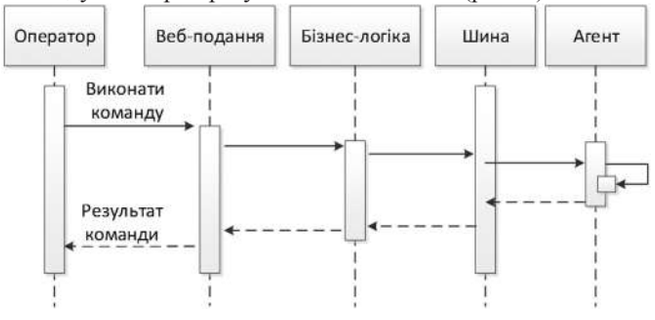


Рисунок 3. Виконання команди

У разі, коли стан системи чи її компонентів динамічно змінюється, адміністраторові необхідно бачити поточні зміни в реальному часі.

При використанні вищезгаданих механізмів, з урахуванням того, що агент відсилає інформацію не відразу, а поміщаючи її в певний буфер (із метою оптимізації мережевого трафіку), адміністратор зможе бачити інформацію про поточний об'єкт із затримками в розмірі періоду відправки даних моніторингу. Окрім того, йому буде необхідно здійснювати повторний запит до бази даних (рис. 4), що створить додаткові затримки, оскільки веб-інтерфейс базується на протоколі HTTP, який після повернення даних на бік клієнта розриває з'єднання. Для динамічного оповіщення веб-клієнтів прийнято використовувати технології WebSocket, Server Sent Events, Forever Frame та Ajax long polling. Це дозволяє оповіщувати адміністратора з мінімальними затратами часу. Також необхідно реалізувати на агентському боці механізм, який відсилатиме дані частіше, коли користувач переглядає даний об'єкт.

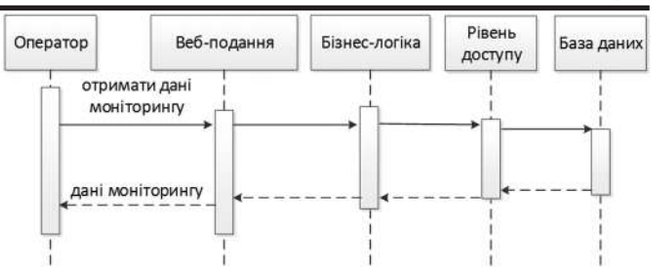


Рисунок 4. Отримання даних моніторингу

Однією з найпопулярніших наявних бібліотек для роботи з оповіщенням веб-клієнтів є SignalR. Самостійна реалізація користувачем аналогічного алгоритму має ряд мінусів:

- Розробка модуля, який буде кращим чи принаймні аналогічним, – завдання ресурсозатратне.
- Проблемою є реалізація логіки визначення інтерфейсу підключення, оскільки SignalR із доступного набору алгоритмів підтримує найшвидший, що підтримується серверною та клієнтською ОС, а також браузером.
- Розширення поточного компоненту можливе завдяки незалежності реалізації бібліотеки SignalR від інших компонентів.

Із врахуванням вищенаведених пунктів було вирішено інтегрувати SignalR як бібліотеку для роботи з оповіщеннями в реальному часі. Розглянемо її роботу. В ній виділяються три ключові компоненти: Hub, ядро SignalR, клієнтські JavaScript-обробники повідомлень (перший і останній необхідно визначити користувачеві, а ядро надає як серверний, так і клієнтський функціонал).

Hub являє собою клас мови C#, що надає набір базових подій, таких як підключення клієнта, додавання поточного ідентифікатора підключення до колекції наявних підключень та вибору поточного типу взаємодії з результуючим клієнтом залежно від підтримки способів комунікації, розсилку групових повідомлень, дана бібліотека підтримує створення так званої групи, що містить у собі список підключень. Відключення поточного користувача (видалення поточного connectionId зі списку клієнтів), повторного підключення поточного користувача: обробка події, яка виникає, коли відключений користувач спробує повторно підключитись.

Ядро бібліотеки реалізує автоматичний вибір найкращого типу підключень із-поміж таких, як:

1. WebSocket
2. Server Sent Events
3. Forever Frame
4. Ajax long polling

Веб-сокети – найбільш нова та оптимізована технологія, що базується на створенні повнодуплексного зв'язку над HTTP-з'єднанням, проте є обмеження версії ПЗ і ОС: веб-сокети можна використовувати лише при версії серверної ОС більшій чи рівній за Windows Server 2012 та встановленій на платформі .Net, версія якого є більшою або рівною 4,5. За відсутності можливості встановити з'єднання через веб-сокети, бібліотека SignalR намагається встановити з'єднання з використанням Server Sent

Events, що є обгорткою над механізмом, що реалізує JS API і надає серверу можливість відправляти події у вигляді DOM-об'єктів. Наступний спосіб комунікації (Forever Frame) використовується лише в Internet Explorer, де обмін повідомленнями реалізовано через створення прихованого Iframe-компоненту, в результаті чого ігноруються тайм-аути щодо відключення поточного користувача і життєвий цикл поточного з'єднання завершується закриттям сторінки. Ajax long polling є реалізацією для старих версій браузерів та реалізує найпростіший спосіб опитування сервера, а саме: відправляє HTTP Get-запит, після чого сервер очікує появи події, адресованої клієнту, та відправляє дані.

Основою роботи SignalR є можливість опрацьовувати події, згенеровані сервером на боці клієнта. Даний механізм працює таким чином: JavaScript-бібліотека SignalR знає про спосіб з'єднання з сервером та працює з ним, програмістові залишається лише визначити свої обробники кожної конкретно взятої події на боці клієнта, які динамічно змінюватимуть вміст сторінки залежно від отриманих даних.

На боці сервера знаходиться Hub, що інкапсулює логіку підключення/відключення та оповіщення оператора в реальному часі.

Рисунок 5. Оповіщення в реальному часі

При переході на сторінку сутностей конкретного об'єкта моніторингу веб-переглядач повідомляє про це Hub, який у свою чергу через сервісну шину передає агенту команду на виконання, а саме відправлення даних моніторингу в реальному часі, за вказаними параметрами генеруються відповідні команди моніторингу (рис. 5). Варто зазначити, що інтеграція механізму реального часу є ресурсозатратною операцією, через що важливо перевіряти необхідність використання даного оповіщення, яка присутня лише у разі, коли користувач переглядає той чи інший параметр, звідки витікає, що після того, як користувач залишив поточну сторінку чи перейшов на іншу ОМУ, пакети з даними моніторингу відправляти не потрібно. З огляду на вищесказане було введено такі обмеження:

1. Агент відправляє інформацію моніторингу покомандно залежно від типу ОМУ;
2. На сервері реалізовано групи користувачів, кожній «групі» відповідає один, конкретний, об'єкт моніторингу, під час підписки на отримання даних моніторингу в реальному часі користувач додається до групи, щоб однією відправкою даних оповіщати користувачів, які переглядають поточну інформацію.
3. Механізм відключення оповіщення в реальному часі. При оповіщенні агенту вказується час, через якій команду моніторингу в реальному часі буде вимкнено, сервер, у свою чергу, знаючи про даний проміжок часу, відновлює режим оповіщення в реальному часі. У разі покидання користувачем сторінки видаляє поточного користувача з групи і шле команду на зупинення оповіщення в реальному часі.

У роботі реалізовано модуль оповіщення адміністратора про зміни в системі, здатний відправляти на сервер у реальному часі лише ті дані, які в даний момент відображаються на веб-інтерфейсі, та автоматично припиняти цю передачу, коли дані більше не потрібні, що, в свою чергу, надає операторові можливість приймати рішення про внесення на віддаленій машині тих чи інших змін для оптимізації ресурсів і недопуску критичних ситуацій.

ДЖЕРЕЛА:

1. Ролик А.И. Концепция управления корпоративной ИТ-инфраструктурой / А.И. Ролик // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: «ВЕК+», 2012. – № 56. – С. 31–55.

2. Ролик А.И. Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский, А.А. Покотило // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: «ВЕК+», 2010. – № 52. – С. 39–52.

3. Ролік О.І., Автоматизована система моніторингу та управління критичними системами/ О.І. Ролік, д.т.н.; Б.А. Март, М.Д. Литвиненко, Є.А. Вовк // XII міжнародна наук.-техн. конф. «АВІА-2015», 28–29 квітня 2015 р.. – К.: НАУ. – 2015. – С. 6.86–6.90.

4. Что такое Long-Polling, WebSockets, SSE и Comet [Електронний ресурс]. – Режим доступу: https://myrusakov.ru/long-polling-websockets-sse-and-comet.html (дата звернення 10.05.2016). – Что такое Long-Polling, WebSockets, SSE и Comet.

5. SignalR Transports Explained [Електронний ресурс]. – Режим доступу: http://kevgriffin.com/signalr-transports-explained/ (дата звернення 10.05.2016). – SignalR Transports Explained.

6. Understanding and Handling Connection Lifetime Events in SignalR [Електронний ресурс]. – Режим доступу: http://www.asp.net/signalr/overview/guide-to-the-api/handling-connection-lifetime-events (дата звернення 10.05.2016). – SignalR Transports Explained Understanding and Handling Connection Lifetime Events in SignalR

ОБРОБЛЕННЯ ІНФОРМАЦІЇ В СКЛАДНИХ СИСТЕМАХ

Implementation of the power function with floating point in on-line mode

Zhabin Valeriy Ivanovich Kokhan Olena Sergiyivna Tokar Andrey Gennadievich

Annotation. An algorithm for calculating the function $Z = X^2$ with floating point, which enables to overlap processing, bit-by-bit input and output of information from high positions using redundant number system, is proposed. It is shown that the use of computer modules implementing this calculation mode enables you to perform related operations in the concurrent mode and reduces hardware complexity.

Keywords. On-line calculations, operations overlap, power function, flow graph, the system-on-chip.

Реалізація степеневої функції з плаваючою комою в неавтономному режимі

Жабін Валерій Іванович Кохан Олена Сергіївна Токар Андрій Геннадійович

Анотація. Пропонується алгоритм обчислення функції з плаваючою комою, що дозволяє з використанням надлишкової системи числення поєднувати процеси порозрядного введення, обробки і порозрядного виведення інформації, починаючи зі старших розрядів. Показано, що застосування обчислювальних модулів, що реалізують такий режим обчислень, дозволяє виконувати залежні операції в режимі суміщення і зменшує апаратурні витрати на реалізацію системи.

Ключові слова. Неавтономні обчислення, суміщення операцій, степенева функція, потоковий граф, система на кристалі.

Введення. Аналіз особливостей роботи обчислювальних систем у контурі керування об'єктами і процесами, а також алгоритмів обробки даних дозволяє сформулювати вимоги до систем реального часу. Ефективність паралельних обчислень залежить від реалізованого рівня паралелізму, що пов'язано з розміром зерна подання графа обчислень. При вирішенні траєкторних задач використовуються методи лінійної алгебри, інтерполяції функцій за допомогою поліноміальної апроксимації і т. і. [1].

Зазначені алгоритми обробки даних мають дрібнозернисту структуру [2, 3], що визначає архітектуру обчислювальних систем (ОС), які функціонують в контурі управління. ОС повинні забезпечувати високу швидкість реалізації алгоритмів з дрібнозернистою структурою, забезпечувати переважно диференціальний тип передач інформації між обчислювальними вузлами. Прискорення обчислень можливо при паралельній обробці даних.

Високий рівень розпаралелювання може бути досягнутий при поданні алгоритмів у вигляді графа з дрібнозернистою структурою, коли вершинам графа відповідають окремі операції. При цьому збільшується число паралельних гілок, що дає потенційну можливість використовувати більшу кількість паралельно працюючих обчислювальних модулів (ОМ). Однак, швидкість обробки інформації пов'язана не тільки з мінімізацією часу виконання паралельних гілок, але і мінімізацією витрат на обмін інформацією між ОМ. Виграш в швидкості обчислень при збільшенні рівня розпаралелювання алгоритмів пов'язаний зі зростанням інтенсивності обміну між гілками, що збільшує час вичисленій. Цей фактор є дуже важливим і повинен враховуватися при виборі архітектури ОС. Для реалізації дрібнозернистих алгоритмів недоцільно використовувати класичні багатопроцесорні системи типу MIMD (Multiple Instruction – Multiple Data) [4], оскільки в них використовуються складні процедури обміну інформації, що неприйнятно для обміну на рівні окремих слів.

Реалізація потокових обчислень у неавтономному режимі. Використання сучасної елементної бази (програмованих та замовних НВІС) і технології проектування SoC (System on Chip – система на кристалі) дозволяє побудову систем з різною архітектурою. Для скорочення витрат часу на обмін даними застосовують паралельні системи з безпосередніми зв'язками (ПСБЗ) між обчислювальна модулями (ОМ) [5, 6].

У ПСБЗ виходи одних обчислювальних модулів (ОМ) з'єднуються з входами інших відповідно з графом потоків

даних (ГПД). У процесі обчислень дані пересилаються від одних модулів до інших, перетворюючись у кожному з них відповідно до заданої операції. Дані пересилаються безпосередньо між ОМ, що прискорює обчислення. Важливою проблемою при цьому є зменшення числа зв'язків між ОМ з метою економії внутрішнього апаратного ресурсу мікросхем, що пов'язано з енергоспоживанням і надійністю ОС. Одним з підходів до вирішення проблеми зменшення зв'язків між ОМ і виводів мікросхем є використання квазіпаралельній арифметики, що дозволяє поєднувати процеси порозрядного введення операндів і порозрядного формування результатів зі старших розрядів.

Режим роботи таких ОМ називають неавтономним, так як для виконання послідовності операцій необхідно кілька ОМ, які обмінюються інформацією в процесі роботи. За рахунок порозрядний передачі інформації скорочується число зв'язків між ОМ.

Обробку даних у неавтономному режимі зі старших розрядів операндів можна здійснювати лише в надлишкових системах числення, в яких відсутнє поширення переносів у старші розряди.

Відомі методи неавтономних обчислень призначені для чисел з фіксованою комою в симетричних надлишкових системах числення [6]. Однак, обчислення з фіксованою комою мають наступні недоліки: обмеження на діапазон подання чисел; складність синхронізації просування даних для забезпечення необхідного співвідношення їх величин у певному місці ланцюжка операцій (наприклад, при додаванні розряди, що підсумовуються, повинні мати однакову вагу).

Для усунення зазначених недоліків доцільно використовувати в ПСБС методику обчислень з плаваючою комою. Зазначена методика в даний час розроблена для основних арифметичних операцій. Для розширення можливостей ПСБС необхідна розробка алгоритмів виконання різних операцій, в тому числі одномісних функцій.

Реалізація степеневої функції у неавтономному режимі. При вирішенні завдань лінійної алгебри, інтерполяції функцій, перетворення координат, цифрової обробки сигналів використовується одномісна функція $Z = X^2$. Розглянемо реалізацію такої функції з плаваючою комою в неавтономному режимі.

Особливість неавтономного режиму полягає в тому, що розряди результату операції формуються з затримкою на p кроків. Тому будемо розглядати функцію $Y = k^{-p} X^2$. У формі з плаваючою комою функцію можна представити як $Y = 2^{P_x} \cdot M_x$, ГДж P_x – порядок, M_x – мантиса числа X. Порядок представлений в канонічній двійковій системі числення, а мантиса - в квазіканонічній системі числення з основою $k = 2$ і цифрами {-1,0,1}.. Квазіканонічна система є надлишковою, одне і те ж число може мати кілька подань, наприклад, $0{,}1011 = 0{,}1\bar{1}\bar{1}1$. Це дозволяє формувати результат порозрядно без переносів в старші розряди.

Код операнда в квазіканонічній системі має вигляд

$$X = \sum_{i=1}^{n} x_i \cdot k^{-i}, \tag{1}$$

де $x_i \in \{-1,0,1\}$.$x_i \in \{-1, 0, 1\}$

Оскільки обчислюється функція $k^{-p} \cdot X^2$, то для функції X^2 потрібно змістити кому в результаті на p розрядів вправо. Для отримання n розрядів після коми функції X^2 потрібно зробити $m = n + p$ кроків обчислень. Функція Y буде мати вигляд $Y = \sum_{i=1}^{m} y_i \cdot k^{-i}$, де $y_i \in \{-1,0,1\}$ $z_i \in \{-1, 0, 1\}$. Операнд X вводиться із старших розрядів. Позначимо за X_i та Y_i коди операнда та функції, що містять тільки i розрядів справа від коми, наприклад $X_5 = 0, x_1 x_2 x_3 x_4 x_5 0 \ldots 0$, $Y_3 = 0, y_1 y_2 y_3 0 \ldots 0$.

Після виконання m кроків отримаємо функцію $Y_m = Y$ з похибкою, яка не перевищує k^{-m-1}, якщо на кожному кроці формувати цифру y_i так, щоб виконувалась умова

$$Y_i - \frac{k^{-i}}{2} \le k^{-p} \cdot X_i^2 < Y_i + \frac{k^{-i}}{2}. \tag{2}$$

Якщо ввести заміну

$$R_i = (k^{-p} \cdot X_i^2 - Y_i) \cdot k^i, \tag{3}$$

то вираз (2) матиме вигляд

$$-\frac{1}{2} \le R_i < \frac{1}{2}. \tag{4}$$

Нерівність (4) виконується у вихідному стані при $R_0 = 0$ Будемо вважати, що на (i-1)-му кроці нерівність (4) також виконується, та визначимо при якому мінімальному значенні p співвідношення (4) буде виконуватись на будь-якому наступному кроці.

З використанням виразів (1) та (2) подамо (3) у вигляді

$$R_i = kR_{i-1} + k^{-p+1} x_i X_{i-1} + k^{-p-i} x_i^2 - y_i. \tag{5}$$

Ввівши заміну

$$H_i = kR_{i-1} + k^{-p+1} x_i X_{i-1} + k^{-p-i} x_i^2, \tag{6}$$

одержимо (5) у вигляді

$$R_i = H_i - y_i. \tag{7}$$

Згідно рівності (7), запишемо (4) у вигляді

$$y_i - \frac{1}{2} \le H_i < y_i + \frac{1}{2}. \tag{8}$$

На основі аналізу нерівності (8) можна визначити мінімальне значення p, при якому виконується нерівність (4), а саме

$$p = \left\lfloor \log_k \frac{4}{2 - k + 1} \right\rfloor. \tag{9}$$

Для основи $k = 2$ відповідно з (9) визначаємо $p = 2$. Підставляючи в нерівність (8) значення $y_i \in \{-1,0,1\}$, отримаємо правило формування цифри для функції Y

$$y_i = \begin{cases} -1, \text{якщо } H_i < -\frac{1}{2}; \\ 0, \text{якщо } -\frac{1}{2} \le H_i < \frac{1}{2}; \\ 1, \text{якщо } \frac{1}{2} \le H_i. \end{cases} \tag{10}$$

Таким чином, на кожному кроці потрібно по формулі

(6) знайти значення H_i, згідно якого визначити за допомогою правила (10) цифру y_i функції, та знайти значення R_i для наступного кроку за допомогою формули (7).

Розряди операнду і результату передаються між ОМ за допомогою двох провідників. Цифри з множини {-1,0,1} можуть кодуватись відповідно {10,00,01}.

Алгоритм обробки порядку дуже простий і зводиться до зсуву та додавання кодів. Пересилання порядків в ОС можна виконувати послідовним кодом в канонічній системі числення.

Алгоритм роботи ОМ.

Прийняти порядок P_X в канонічній системі числення з цифрами {0,1}.

Отримати попередній порядок результату за формулою $P_Y = 2 \cdot P_X + p$.

В кожному циклі обчислень формувати цифру мантиси результату. Починаючи з першого циклу, нульові цифри мантиси не видавати з ОМ. При цьому корегувати порядок наступним чином: $P_Y := P_Y - 1$.

При формуванні першої ненульової цифри результату Y_i видати із ОМ остаточний порядок P_Y і першу ненульову цифру мантиси результату.

Приймати всі наступні цифри мантиси операнду і видавати наступні цифри мантиси результату до отримання необхідною кількості розрядів після першого ненульового розряду.

На кожному кроці в ОМ вводиться по одному розряду операндів і формується один розряд результату. При цьому розряд проміжного результату, отриманий на i-му кроці в одному ОМ при виконанні j-ї операції, може бути використаний на (i+1)-му кроці в іншому ОМ при виконанні (j+1)-ї операції.

При такому режимі обчислень виконання наступної операції буде починатися не після завершення виконання попередньої операції, а відразу після отримання першого розряду результату цієї операції. Такий режим роботи ОМ дозволяє виконувати ланцюжки залежних за даними операцій у режимі суміщення, що створює передумови для прискорення обчислень порівняно з ОМ, які працюють за правилами паралельної арифметики.

Модулі, що виконують операції при порозрядному введення і виведення даних, за структурою ближче до паралельним ОМ, а не послідовних пристроїв, що визначило їх назву «квазіпаралельні».

ВИСНОВКИ

ОС на базі квазіпаралельних ОМ використовує істотно менше ресурсів на кристалі (ПЛІС або замовних НВІС), що пов'язано з малим числом зв'язків. Економляться як внутрішні ресурси мікросхем, так і її виводи. Це дає можливість реалізувати на тій же мікросхемі ряд інших пристроїв, що відносяться до однієї або різних систем.

Побудова системи на одному кристалі забезпечує підвищення її надійності, зменшення енергоспоживання та габаритів. З використанням квазіпаралельних ОМ реалізується паралелізм на рівні обробки розрядів операндів, що дає потенційну можливість прискорити обробку інформації.

Таким чином, отримані результати підтверджують ефективність застосування неавтономних методів порозрядної обробки інформації зі старших розрядів у системах типу ПСНС на базі програмованих і замовних НВІС.

ЛІТЕРАТУРА

1. Байков В.Д. Решение траекторных задач в микропроцессорных системах ЧПУ / В.Д.Байков, С.Н.Вашкевич. – Л.: Машиностроение, 1986.– 105 с.

2. Сосонкин В.Л. Принципы построения систем ЧПУ с открытой архитектурой / В.Л.Сосонкин, Г.М.Мартинов // Приборы и системы управления. – 1996. – №8. – С. 18-21.

3. Благовещенский Ю.В. Вычисление элементарных функций на ЭВМ / Ю.В.Благовещенский, Г.С.Теслер. – Киев, «Техника», 1977. – 208 с.

4. Воеводин В.В. Параллельные вычисления / В.В.Воеводин, Вл.В.Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.

5. Жабин В.И., Жабина В.В., Безгинский М.А. Эффективность потоковых вычислений в системах с непосредственными связями, реализованных на ПЛИС / В.И.Жабин. В.В.Жабина М.А.Безгинский // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. праць. – К.:

6. Жабин В.И. Построение быстродействующих специализированных вычислителей для реализации многоместных выражений / В.И.Жабин, В.И.Корнейчук, В.П.Тарасенко. – Автоматика и вычислительная техника. – 1981. – № 6. – С. 18-22.

The algorithm for formation rating list of entrants according to their priority

Shapoval Oleksandr Serhiiovych
student, faculty of Informatics and Computer Science, department of Technical Systems Automation and Control, National technical university of Ukraine "Kyiv Polytechnic Institute"
Ukraine, Kiev

The article describes algorithm for formation rating list of entrants according to their priority. It allows calculating passing scores of specialties with quite high accuracy before the admission timeline ended.

Keywords: data analysis, university admission, abit-poisk, vstup info

Алгоритм формування рейтингових списків абітурієнтів з урахуванням пріоритетності заяв

Шаповал Олександр Сергійович
студент, факультет Інформатики та обчислюваної техніки, кафедра Автоматики і управління в технічних Системах,
Національний технічний університет України «Київський політехнічний інститут»
Україна, Київ

В роботі описано алгоритм формування рейтингових списків вступників з урахуванням пріоритетності заяв. Він дозволяє розрахувати прохідні бали спеціальностей з досить високою точністю ще за декілька днів до закінчення роботи приймальних комісій.

ВСТУП

Ключовим етапом в житті кожної людини є вступ до вищого навчального закладу. Завдяки впровадженню системи ЗНО у кожної дитини є чудовий шанс показати свої знання, склавши вибрані тести незалежного оцінювання та вступити до омріяного ВНЗ. Після отримання результатів складання тестів абітурієнт має змогу подати певну кількість заяв до вищих навчальних закладів. В заяві абітурієнт обов'язково повинен вказати її пріоритетність – заздалегідь встановлена вступником черговість заяв (від 1 до 15, де 1 – найбільш пріоритетна заява)[1]. Цей показник використовується при формуванні списку після закінчення подання заяв від вступників. Для відстеження своїх рейтингових позицій був створений спеціальний сайт – «ІС Конкурс»[2]. За допомогою цього ресурсу кожен бажаючий може відслідковувати заяви абітурієнтів.

Проблема полягає в тому, що вступник не має змоги до закінчення терміну подання заяв ознайомитися із попередньо сформованими рейтинговими списками із врахуванням пріоритетності заяв. Наявність таких попередньо сформованих даних надасть абітурієнтам, які, наприклад, не потрапляють на бюджет, подати нову заяву на той напрям, в якому програма розрахувала недобір на бюджетні місця. Таким чином вирішуються дві проблеми: перша – для вступників, які не є рекомендованими на зарахування за кошти держави , мають можливість все ж таки потрапити на бюджет; друга проблема – проблема наявності недобору на деякі спеціальності. Попередньо сформовані рейтингові списки дозволять керівникам ВНЗ повідомляти вступникам про наявні бюджетні місця, або коректувати державне замовлення, шляхом відмови бюджетних місць на тих напрямах, де присутній недобір.

Актуальність даної роботи полягає у необхідності вдосконалення відкритих систем для надання додаткової інформації під час вступної кампанії – вступу абітурієнтів до вищих навчальних закладів України.

Метою даної роботи є створення алгоритму та програмної системи розрахунку рейтингових списків із урахуванням пріоритетності заяв на основі відкритих даних, які доступні через мережу Інтернет.

ОПИС АЛГОРИТМУ

1.1. Підготовка вхідних даних

Для правильної роботи алгоритму перш за все необхідно визначитись із джерелом відкритих даних. Було використано інформацію із «ІС Конкурс» для збору та обробки основних даних, сайт «Перевірка сертифікатів зовнішнього незалежного оцінювання 2015 року»[3] – для ідентифікації абітурієнтів. У цій статті розглянуто приклади формування рейтингових списків за 28 липня 2015 року – за 5 днів до закінчення прийому заяв від вступників.

На підготовчому етапі було створено три таблиці – спеціальності вузів (university_specialities), абітурієнти (enrollees), заяви абітурієнтів (enrolle_statements).

Таблиця university_specialities.

Обов'язкові поля:

- university_id – назва або ідентифікатор вузу
- speciality_code – шифр спеціальності
- speciality_id – унікальний ідентифікатор запису
- free_places – кількість бюджетних місць
- speciality_k – коефіцієнт квоти для вступу поза конкурсом
- is_crrection_complete – показник завершення формування списку за даним напрямом

Обов'язково в таблицю слід вносити лише ті спеціальності, за якими передбачена пріоритетність заяв.

Таблиця enrolle_statements.

Обов'язкові поля:

- statement_id – ідентифікатор заяви
- enrolle_full_name – піб абітурієнта
- speciality_id – ідентифікатор конкретної спеціальності конкретного вузу
- statement_pk – право на поза конкурсний вступ
- statement_pch – право на першочерговий вступ
- statement_score – конкурсний бал заяви
- statement_target_direction – цільове направлення
- statement_university_status – статус заяви – рекомендований (1), не рекомендований (0)

• statement_university_tmp_status – тимчасовий статус заяви: не рекомендований(0), рекомендований(1), ця заява не може братись участь у наданні рекомендації, оскільки абітурієнта вже рекомендовано на іншому напрямку за вищим пріоритетом(3)

Таблиця enrolles. Заповнюється за допомогою виокремлення унікальних записів ПІБ із таблиці та додатковою перевіркою на сервісі «Перевірка сертифікатів зовнішнього незалежного оцінювання 2015 року». Даний сервіс після вводу даних видає перелік предметів ЗНО даної особи. Якщо є кілька вступників з однаковими ПІБ – система видасть відповідну інформацію. Таким чином можна визначати абітурієнтів з однаковим ПІБ та однозначно їх ідентифікувати лише за допомогою відкритих Інтернет ресурсів.

Обов'язкові поля:

abiturient_id – ідентифікатор абітурієнта

abiturient_full_name – ПІБ абітурієнта

abiturient_rec_statement_id – ідентифікатор рекомендованої заяви на бюджет з найвищим пріоритетом

statement_rec_recount – кількість наданих рекомендацій до зарахування на бюджет

1.2. Розробка алгоритму

Після формування відповідних таблиць та заповнення їх даними можна виконувати алгоритм автоматичного формування рейтингових списків.

На першому етапі алгоритм знаходить усі спеціальності, на яких ще не завершено формування списків та здійснює перші рекомендації на вільні бюджетні місця з урахуванням конкурсного балу та інших важливих показників, які будуть описані нижче.

На другому етапі серед усіх рекомендованих заяв залишаються лише заяви із найвищим пріоритетом для кожного абітурієнта. Далі програма рекурсивно повторює перший та другий етапи поки це можливо.

ЕТАП 1.

1. Отримати всі записи із таблиці спеціальності вузів (university_specialities), які задовольняють умові is_correction_complete = 0. Якщо такі записи існують – перейти до пункту 2, інакше – алгоритм завершив свою роботу.

2. Для кожного запису із пункту 1 виконати наступні дії:

• Отримати кількість рекомендованих заяв – позначемо буквою A

• Отримати кількість можливих рекомендацій – B. Цей вираз задовольняє умові statement_university_status = 0 та statement_university_tmp_status = 0

Якщо кількість можливих рекомендацій B більше нуля та кількість бюджетних місць даної спеціальності теж більше нуля – перейти до пункту 2.1, інакше – для запису даної спеціальності поставити позначку is_correction_complete = 1 та перейти до пункту 3.

2.1 Виконати наступні дії

• Отримати кількість рекомендованих заяв зі статусом statement_pk = 1 – C.

• Отримати кількість можливих рекомендацій заяв зі статусом statement_pk = 1 – D.

Якщо з числа можливих рекомендацій залишились лише заяви зі статусом statement_pk =1 – для запису даної спеціальності поставити позначку is_correction_complete = 1 та перейти до пункту 3, інакше – перейти до пункт 2.3.

2.3 Виконати наступні дії

• Отримати кількість вільних місць для надання рекомендацій на дану спеціальність – E

Якщо E > 0 – перейти до пункту 2.4, інакше – для запису даної спеціальності поставити позначку is_correction_complete = 1 та перейти до пункту 3.

2.4 Отримати із таблиці заяви абітурієнтів (enrolle_statements), попередньо відсортовані за спаданням записи за наступними полями: statement_target_direction, statement_pk, statement_score, statement_pch . Обрати лише перші n відсортованих записів, де n = E – кількість вільних місць для надання рекомендацій на дану спеціальність. Для кожного запису заяви абітурієнта виконати наступні дії:

• враховуючи квоту заяв для вступу поза конкурсом statement_pk встановити поле statement_university_tmp_status = 1

Таким чином в цьому пункті ми тимчасово рекомендуємо до зарахування нову партію заяв абітурієнтів з урахуванням конкурсного балу вступника та інших полів за умови наявності вільних місць. Далі переходимо до пункту 3.

ЕТАП 2.

3. Після надання тимчасових рекомендацій для всіх абітурієнтів на вільні бюджетні місця потрібно для кожного абітурієнта знайти рекомендовану заяву з найвищим пріоритетом, поставити даній заяві тимчасову найвищу рекомендацію. Всім іншим заявам, пріоритет яких менший – поставити позначку statement_university_tmp_status = 3. Це означає, що на наступній ітерації алгоритму такі заяви не будуть розглядатися при наданні рекомендації. Таким чином після кожної ітерації алгоритму вивільняється певна кількість вільних бюджетних місць. Алгоритм буде працювати до тих пір, поки вільних місць зовсім не залишиться, або поки буде технічна можливість надавати рекомендації – наприклад на деяку спеціальність залишились вільні бюджетні місця, але усі вільні заяви для даного напрямку вже рекомендовані на іншому напрямку за кращими пріоритетами. Отже на деяких спеціальностей можуть виникнути недобір на бюджет.

На цьому пункті додатково після вивільнення бюджетних місць потрібно встановлювати значення поля is_correction_complete в нуль для тих спеціальностей, які після першого етапу мали заповнені бюджетні місця рекомендаціями, але після другого етапу внаслідок ануляції заяв з нижчим пріоритетом знову отримали певну кількість незаповнених бюджетних місць.

Після закінчення другого етапу алгоритм знову переходить до пункту 1.

1.3. Результат роботи алгоритму

Результат роботи алгоритму – сформовані рейтингові списки із урахуванням пріоритетності заяв. За такими списками дуже легко вирахувати недобір та прохідний бал на бюджет. Точність таких даних у порівнянні із офіційними списками залежить від деяких факторів, а саме:

• чим раніше до закінчення прийому заяв від абіту-

рієнтів буде розрахований рейтинговий список – тим більша неточність розрахунку

• за допомогою відкритих даних неможливо отримати квоту на зарахування абітурієнтів поза конкурсом. Вам потрібно самостійно підібрати квоту для всіх спеціальностей. Отже алгоритм на основі лише відкритих даних не може дати точність 100% співпадання з офіційно сформованими рейтинговими списками, які зазвичай доступні у «ІС Конкурс» після кількох днів після завершення прийому заяв від абітурієнтів

Для реалізації даного алгоритму використана СУБД Mysql для зберігання даних та мова програмування PHP для безпосередньої реалізації алгоритму.

На рисунку нижче представлено фрагмент сформованого списку

‹ 1 2 3 4 5 … 13 ›

№	ПІБ	Всього балів	Ц	ПК	Пріоритет	М.ст	К.р	Куди
101	[redacted]Олегович	189.771	–	–	2	1	1	сюди
102	Олесюк[redacted]	199.500	–	+	2	0	0	КПІ, системний аналіз (1й пр.)
103	[redacted]огдан[redacted]	192.597	–	+	2	0	0	КПІ, програмна інженерія (1й пр.)
104	[redacted] Євгенійович	192.240	–	+	5	0	1	КНУ ім. Тараса Шевченка, програмна інженерія (1й пр.)
105	Терпіловський[redacted]	187.694	–	+	4	0	1	КНУ ім. Тараса Шевченка, програмна інженерія (1й пр.)
106	[redacted] Віталійович	183.228	–	+	2	0	1	КПІ, програмна інженерія (1й пр.)
107	Осіпов[redacted]	183.172	–	+	5	0	0	КПІ, комп'ютерна інженерія (1й пр.)
108	Закусило[redacted]	179.176	–	+	3	0	1	КНУ ім. Тараса Шевченка, комп'ютерна інженерія (1й пр.)
109	Булах[redacted]	177.418	–	+	2	0	0	КНУБіА, архітектура (1й пр.)
110	[redacted]Володимирович	175.976	–	+	2	0	0	НУДПС, правоохоронна діяльність (1й пр.)
111	[redacted]Віталійович	175.564	–	+	2	0	2	КПІ, програмна інженерія (1й пр.)
112	[redacted]Анатолійович	171.764	–	+	8	0	1	КПІ, комп'ютерна інженерія (2й пр.)

Рисунок 1 – фрагмент розрахованого списку із урахуванням пріоритетності в системі пошуку абітурієнтів «Абіт Пошук»[4]

На рисунку 1 навпроти кожної заяви присутня колонка «Куди». Вона означає, куди абітурієнт може бути рекомендованим на зарахування за кошти держави. Якщо навпроти заяви значення цієї колонки дорівнює «Сюди» - це значить, що програма розрахувала проходження абітурієнта на бюджет саме на цей напрям. Також значення може мати прочерк – це означає, що алгоритму не вдалося рекомендувати абітурієнта на місця держзамовлення. В іншому випадку – буде відображено назву ВНЗ, напрям та пріоритет заяви, яка була рекомендована до зарахування.

Було проаналізовано точність роботи алгоритму на прикладі рейтингових списків НТУУ «КПІ» (рис. 2), в даному випадку точність роботи алгоритму формування списків із урахуванням пріоритетності заяв – це таке значення, наскільки прохідний бал, розрахований на основі сформованих алгоритмом списків відрізняється від розрахованого прохідного балу на основі офіційних рейтингових списків, які були оприлюднені пізніше і викладені на «ІС Конкурс». На осі абсцис – допустима похибка розрахунку прохідного балу – від ± 0 балів (100% співпадання) до ± 5 балів. На осі ординат – відсоток правильно розрахованих рейтингових списків у межах допустимої похибки.

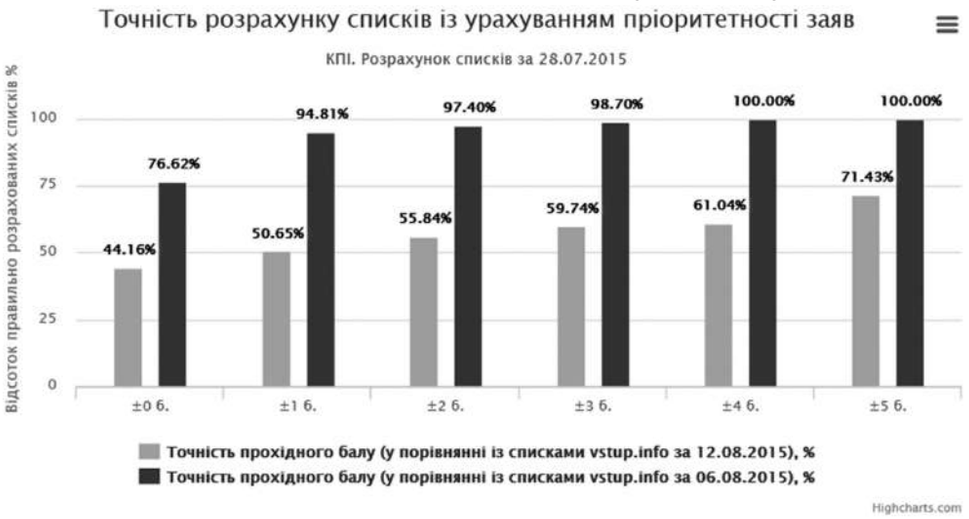


Рисунок 2 – залежність точності розрахованих рейтингових списків від допустимої похибки

На рис. 2 колонки справа (чорного кольору) – відповідність розрахованих алгоритмом списків офіційному розподілу, колонки зліва (синього кольору) – відображення точності розрахунку прохідного балу після закінчення терміну зарахування вступників за кошти держави.

Важливі дати, зображені на рисунку:

Для демонстрації роботи алгоритму використовувались рейтингові списки «ІС Конкурс» за 28 липня 2015 року – за 5 днів до закінчення прийому заяв від вступників.

6 серпня 2015 року – дата, коли «ІС Конкурс» оприлюднила офіційно сформовані рейтингові списки. Для демонстрації точності алгоритму, який розглядається в даній статті було збережено списки vstup.info за 6 серпня – для подальшого розрахунку прохідного балу на бюджет та його порівняння з даними алгоритму.

12 серпня 2015 року – оприлюднення системою «ІС Конкурс» сформованих списків із зарахуванням абітурієнтів за кошти держави, які були рекомендовані в першу хвилю та виконали відповідні правила прийому до вищих навчальних закладів – віднесли оригінали документів у відповідні терміни. Також в ці списки входять абітурієнти, які були зараховані на вакантні бюджетні місця та виконали відповідні правила вступу.

Головний недолік алгоритму – недостатня кількість відкритої інформації, потрібної для більш точного розрахунку списків. Для покращення результату роботи програми необхідні дані про квоти пільгових категорій вступників.

Переваги алгоритму:

- простота реалізації
- всі необхідні дані для роботи алгоритму можна самостійно отримати із відкритих джерел
- досить точний розрахунок дає змогу абітурієнтам коректувати свої пріоритети та з великою обирати напрям, на який вступник з великою ймовірністю потрапить на бюджет.

ВИСНОВКИ

В роботі було показано розробку алгоритму формування рейтингових списків із урахуванням пріоритетності заяв на основі відкритих даних в мережі Інтернет. Результат роботи даного алгоритму показав, що навіть з неповних відкритих джерел можна досить точно розрахувати попередні списки вступників та отримати приблизні дані щодо прохідного балу на бюджет та кількості незаповнених бюджетних місць.

ПЕРЕЛІК ПОСИЛАНЬ

1. Про затвердження Умов прийому на навчання до вищих навчальних закладів України в 2015 році [Електронний ресурс] – Режим доступу: http://zakon4.rada.gov.ua/laws/show/z1390-14

2. Інформаційна Система «Конкурс» [Електронний ресурс] – Режим доступу: http://vstup.info

3. Перевірка сертифікатів зовнішнього незалежного оцінювання 2015 року [Електронний ресурс] – Режим доступу: http://certs.testportal.com.ua

4. Система пошуку абітурієнтів «Абіт Пошук» [Електронний ресурс] – Режим доступу: http://abit-poisk.org.ua

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК