

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ТЕХНОЛОГИИ MICROSOFT

В ТЕОРИИ И ПРАКТИКЕ ПРОГРАММИРОВАНИЯ

Новые подходы к разработке программного обеспечения
на примере технологий Microsoft и EMC

**Материалы межвузовского конкурса-конференции
студентов, аспирантов и молодых ученых**

25 марта 2014 года



**Санкт-Петербург
Издательство Политехнического университета
2014**

Технологии Microsoft в теории и практике программирования (Новые подходы к разработке программного обеспечения на примере технологий Microsoft и EMC): Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада. СПб.: Изд-во Политехн. ун-та, 2014. 138 с.

В сборнике публикуются материалы докладов студентов ряда вузов, представленные на конкурс-конференцию, проводимую Санкт-Петербургским государственным политехническим университетом и организованную Центром инноваций Microsoft и факультетом технической кибернетики при участии Российских представительств Microsoft и EMC. Доклады отражают современный уровень подготовленности студентов политехнического университета и других вузов – участников конференции в области применения современных средств и технологий разработки программного обеспечения с использованием продуктов компаний Microsoft и EMC.

Представляет интерес для специалистов в области информационных технологий, методов разработки программных проектов различного назначения, систем и средств автоматизации инженерного проектирования, для учащихся и работников системы высшего образования.

Печатается по решению редакционно-издательского совета Санкт-Петербургского государственного политехнического университета.

Редакционная коллегия:

профессор И.Г. Черноруцкий,

профессор В.П. Котляров

Приветствие Российского представительства корпорации Microsoft

Уважаемые участники конференции
«Технологии Microsoft в теории и практике программирования»
(Новые подходы к разработке ПО на примере технологий Microsoft и ЕМС)!

Мир информационных технологий стремительно меняется, появляются новые концепции, направления, технологии, продукты. Современные специалисты должны обладать большим объемом фундаментальных знаний и практических навыков, позволяющих легко ориентироваться в море информации, правильно и своевременно анализировать ее и принимать на ее основе важные решения.

Российские ученые, разработчики и исследователи традиционно славились своей фундаментальной подготовкой и отличными способностями к самообучению, и, к счастью, эти особенности российского образования не только не потеряны, но и продолжают выгодно отличать наших выпускников.

Целью нашей конференции является поддержка научных исследований студентов, аспирантов и молодых ученых в области информатики и информационных технологий, помощь в изучении современных концепций, подходов и программных средств, а также выявление талантливых молодых специалистов в области разработки программных систем и содействие их подготовке к работе в профессиональных коллективах.

На этой конференции вы имеете отличную возможность пообщаться со своими коллегами, поделиться своими мыслями и взглядами на устройство информационного мира, продемонстрировать собственные программные решения. Надеемся, что эта конференция даст всем участникам возможность представить их текущие достижения, а также будет полезна для дальнейшего профессионального роста.

Желаем вам успешной работы и творческих успехов!

С уважением,

Александр Викторович Гаврилов

к.т.н.

Руководитель направления по работе с системой образования

Департамент стратегических технологий

Приветствие Российского представительства корпорации EMC

Уважаемые участники конференции
«Технологии Microsoft в теории и практике программирования»
(Новые подходы к разработке ПО на примере технологий Microsoft и EMC) !

Рынок ИТ в России развивается стремительно. Уровень высшего образования в нашей стране по-прежнему очень высок, однако для многих экспертов очевидно, что не всегда высшая школа успевает за взрывным ростом в ИТ-индустрии и справляется с задачей подготовки профильных специалистов в требуемом количестве. Важная роль успешного преодоления этого дисбаланса как раз отведена построению эффективных партнерских взаимоотношений между наукой и бизнесом, и в частности проведению и поддержке совместных образовательных инициатив между высшей школой и представителями ИТ компаний.

Корпорация EMC уже не первый год поддерживает конференцию «Технологии Microsoft в теории и практике программирования» (Новые подходы к разработке ПО на примере технологий Microsoft и EMC), проводя свою специальную секцию под названием «Подходы к разработке ПО на основе технологий EMC». На мой взгляд, это отличная возможность для студентов, аспирантов и молодых ученых Северо-Запада проявить себя, раскрыть свой творческий потенциал и заявить о себе науке и бизнесу как о молодых перспективных ученых.

Поддержка перспективных студентов именными грантами EMC, проведение совместных научно-исследовательских проектов и студенческих исследований, а также организация и проведение совместных конференций, включая данную, и шесть лет активного партнерства (с 2007 г) привели к логическому заключению меморандума о взаимопонимании между Корпорацией EMC и Санкт-Петербургским государственным политехническим университетом (СПбГПУ) в декабре 2013 года, и торжественному открытию в феврале 2014 года научно-образовательного центра «Политехник – EMC» на базе Института информационных технологий и управления СПбГПУ. Данный центр является уникальной платформой для подготовки высококвалифицированных кадров по направлениям систем хранения и обработки информации в соответствии со стандартами EMC; для поддержки перспективных студентов и проведения совместных исследований; а также для возможности прохождения сертификации в рамках программы курсов Академического партнерства корпорации EMC. Студенты, аспиранты и молодые ученые могут обменяться опытом, получить новые бесценные знания и навыки, необходимые в их будущей профессии и в жизни.

Целью как данной конференции в целом, так и в частности секции EMC является в первую очередь поддержка молодых ученых, их научных исследований в области информатики и информационных технологий; информирование молодых ученых о передовых технологических трендах в индустрии, о современных программных продуктах и решениях,

представленных крупнейшими мировыми ИТ-корпорациями; знакомство молодых специалистов со стратегией развития ИТ компаний, обзор рынка в целом; выявление талантливых и перспективных студентов и аспирантов для потенциального сотрудничества в будущем. Рынок стремительно развивается, поэтому нам хочется помогать вузам и студентам овладевать всеми необходимыми навыками и технологиями, в частности получая с помощью подобных научно-образовательных конференций фундаментальные знания и практические навыки.

Желаю вам успешной работы на конференции, профессионального роста и творческих побед!

С уважением,

Вячеслав Михайлович Нестеров

д.ф.-м.н.

Генеральный директор

Санкт-Петербургского Центра Разработок ЕМС

**Информационное сообщение о проведении учебно-практической
конференции студентов, аспирантов и молодых ученых
«Технологии Microsoft в теории и практике программирования»
(Новые подходы к разработке ПО на примере технологий Microsoft и EMC)**

1. В целях содействия подготовке к будущей работе в профессиональных программистских коллективах, обеспечивающих высокое качество программного продукта, в целях поддержки изучения современных информационных технологий и инструментальных средств в соответствии с мировыми стандартами и действующими международными сертификационными требованиями, а также для выявления талантливых молодых специалистов в области разработки и использования программных систем, инженерных проектов и моделей, компании Microsoft и EMC совместно с Санкт-Петербургским Государственным Политехническим Университетом объявляют учебно-практическую конференцию **«Технологии Microsoft в теории и практике программирования» (Новые подходы к разработке ПО на примере технологий Microsoft и EMC)**.
2. В рамках мероприятия планируется провести:
 - Мастер-класс по созданию приложений на платформе Windows 8/Windows Phone/ Windows Azure
 - Конкурс-конференцию теоретических и практических работ - претендентов на именные дипломы компаний Microsoft и EMC
3. Сроки проведения мероприятия: **25 марта 2014 года.**
4. Место проведения
Санкт–Петербургский Государственный Политехнический Университет
Главное здание
ул. Политехническая д.29
195251, г. Санкт–Петербург
Регистрация 25 марта 9.30 Белый зал
5. Предлагаемая тематика конкурса:
 - Программная инженерия: приложения, продукты и системы
 - Программная инженерия: инструментальные средства и технологии проектирования и разработки
 - Программная инженерия: методы и алгоритмы теории программирования
 - Подходы к разработке ПО на основе технологий EMC

Данный конкурс рассчитан на студентов 1-6 курсов, аспирантов и молодых ученых, а так же выпускников физико-математических классов школы, обучающихся или работающих по направлениям информатики, вычислительной техники, и по направлениям использования информационных технологий, владеющих основами современных промышленных технологий и инструментарием разработки программных продуктов и проектов. Предполагается, что участник должен проявить свои знания и умения не столько в области программирования различных математических головоломок, сколько в области разработки и использования программных продуктов и систем в условиях, максимально приближенных к реальным

процессам проектирования и разработки современных систем различной степени сложности.

Темы проекта могут быть взяты в соответствии с тематикой секции конференции.

На конкурс принимаются работы, оформленные в соответствии с заданными требованиями и представленные в организационный комитет фестиваля не позднее **27 февраля 2014 года**. Требования к представлению и оформлению проектов представлены в документах «Порядок представления проектов на конкурс-конференцию» [0] и «Требования к проведению стендовых докладов и секционных выступлений на конкурсе-конференции» [II-V]

6. Количество работ конкурсантов не ограничено.
7. Конкурс принятых к участию работ проводится в один этап и протекает в виде представления презентаций и секционных докладов. Приглашение отобранным участникам будут рассылаться в период с 10 по 13 марта 2013г. Решение о награждении участников конкурса принимается конкурсной комиссией. Требования к проведению докладов представлены в документе «Требования к проведению стендовых докладов и секционных выступлений на конкурсе-конференции» [IV].
8. Объявления об условиях проведения конкурса и вся дополнительная информация представлены на сайте СПбГПУ в раздел Конференции. Приглашения на участие в конкурсе-конференции будут разосланы в ведущие университеты Северо-Запада.

ЖЕЛАЕМ УСПЕХОВ И ИНТЕРЕСНОЙ РАБОТЫ !

Научный руководитель конференции Чернооруцкий Игорь Георгиевич,
секретарь Эламик Татьяна Николаевна тел. 552-76-66, часы приема: понедельник и среда с 10 до 17 часов.

Адрес: Политехническая ул., 29, 2 уч.корпус, к.358

E-mail: <mailto:msconf@spbstu.ru>

Порядок представления проектов на конкурс-конференцию.

Для представления разработанных проектов в Конкурсный центр устанавливаются следующие правила:

- Желаящие принять участие в конкурсе – конференции на этапе предварительного сбора материалов на почтовый адрес <mailto:msconf@spbstu.ru> присылают анкету участника и тезисы разрабатываемого **проекта, приложения или модели** в виде секционного доклада. Тезисы докладов, представленные на конкурс - конференцию, будут напечатаны в Сборнике материалов конкурса – конференции.
- На заочном этапе – первом туре - конкурсная комиссия на основе тезисов отбирает работы для непосредственного участия в конкурсе, после чего высылает приглашение по электронной почте (при наличии) или по телефону.
- Очный этап проводится в **виде секционных докладов для теоретических или прикладных поисковых работ.**

Конкурсная комиссия оценивает представленные доклады по представлению руководителей секций и отбирают наиболее значимые из них для награждения.

- Все участники награждаются специальными дипломами Microsoft.

I. Требования к содержанию конкурсных работ.

Предполагается, что конференция будет проходить в рамках четырех секций:

C1. Программная инженерия: приложения, продукты и системы

Задача конкурса в разделе C1: продемонстрировать в тезисах и презентации знания и умения в создании программного продукта на основе доклада, содержащего обоснование предлагаемого решения и анализа преимуществ по сравнению с существующими.

(Например, программные проекты, представляющие завершённый продукт или незавершённый, но исполняемый в соответствии с требованиями к промышленному продукту)

Рекомендуется особо отмечать передовые технологии, в особенности технологии Microsoft, поддержанные в программных продуктах и оценку эффективности их применения.

C2. Программная инженерия: инструментальные средства и технологии проектирования и разработки

Задача конкурса в разделе C2: продемонстрировать знания и умения в создании и/или применении перспективных методов и технологий разработки программного обеспечения на основе доклада, содержащего обоснование предлагаемого решения и анализа преимуществ по сравнению с существующими. Наряду с технологиями разработки программных продуктов на секции рассматриваются программные средства систем управления жизненным циклом информации.

(Например, автоматизация разработки спецификации, доказательства корректности спецификаций, их использование при генерации кода, автоматизация

тестирования, средства обеспечения качества программного продукта, средства управления разработкой программного продукта, средства идентификации и информационной безопасности, управляющие и встроенные применения, беспроводная телекоммуникация, моделирование контроллеров и других аппаратно-программных решений, средства управления качеством реализации программного продукта, системы управления электронным документооборотом, безопасностью хранения данных, виртуализация вычислительных ресурсов.)

Рекомендуется особо отмечать передовые технологии, в особенности технологии Microsoft, поддержанные в программных продуктах и оценку эффективности их применения.

С3. Программная инженерия: методы и алгоритмы теории программирования

Задача конкурса в разделе С3: продемонстрировать знания и умения в разработке и применении формальных методов при создании и/или улучшении либо оптимизации характеристик программного обеспечения. на основе доклада, содержащего обоснование предлагаемого решения и анализа преимуществ по сравнению с существующими.

(Например, алгоритмы и методы для проверки корректности программного продукта, исполнимые спецификации, формальные методы для применения в программировании и т.п.)

Оценка применимости предлагаемых подходов в технологиях Microsoft и оценка эффективности его применения

С4. Подходы к разработке ПО на основе технологий EMC

Задача конкурса в разделе С4: продемонстрировать знания и умения в разработке ПО с применением новых технологий EMC. на основе доклада, содержащего обоснование предлагаемого решения и анализа преимуществ.

II. Требования к оформлению студенческих работ на конкурс-конференцию.

Для участия в конкурсе необходимо представить в электронном варианте:

1. **Анкету участника**, составленную в произвольной форме (Ф.И.О., название вуза, факультета, № группы, домашний адрес, телефон, адрес электронной почты, предполагаемая секция, предполагаемое название работы или доклада; сведения о руководителе: Ф.И.О., место работы, адрес электронной почты (если возможно)).
2. **Тезисы доклада и презентацию**, содержащие название работы, постановку задачи, краткое описание проекта, оценку характеристик демонстрационной версии. Объем тезисов 1 страница напечатанного текста.

III. Требования к оформлению тезисов доклада:

электронный вариант текста набирать в редакторе **Microsoft Word версии 2003 (XP)** со следующими параметрами настройки:

- шрифт — Times New Roman;
- стиль шрифта — нормальный (обычный);
- размер кегля шрифта — 12;
- межстрочный интервал — 1;
- параметры страницы: формат А4, поля верхнее — 20 мм, нижнее — 20 мм, левое — 20 мм, правое — 20 мм;
- формулы набирать, пользуясь Microsoft Equation (настройка символов в редакторе формул пропорциональна основному тексту; по возможности, использовать запись формулы в строчку);
- если для понимания сути работы необходим рисунок, он выполняется в виде единой картинки (формата типа bmp) в пределах поля для текста (**не допускаются рисунки, составленные из отдельных элементов**);

Текст тезисов в текстовом поле располагается следующим образом:

- на первой строчке (выровнять влево — левом верхнем углу): УДК (вместе с цифрами печатать прописными буквами);
- на следующей строчке (выровнять вправо): инициалы, фамилия студента строчными буквами, в скобках курс и название кафедры и ВУЗа); инициалы, фамилия, ученая степень, должность руководителя (использовать принятые сокращения); если не умещается в строчку, то можно в две строчки;
- через пробел (выровнять по центру): НАЗВАНИЕ ТЕЗИСОВ ДОКЛАДА (прописными буквами);
- через пробел (с красной строки, равной 1 см): текст тезисов.

ВНИМАНИЕ: Тезисы доклада должны быть написаны ясным языком, без орфографических ошибок.

Образец оформления работы:

УДК 621.319

Андреев Д.В., Литвинов А.П. (4 курс, каф. ИУС, СПбГПУ),
Гаригин А.В., Никитин М.А. (асп., каф. ИУС, СПбГПУ),
Котляров В.П., к.т.н., проф.

УДАЛЁННОЕ УПРАВЛЕНИЕ РС С ИСПОЛЬЗОВАНИЕМ МОБИЛЬНЫХ УСТРОЙСТВ ROCKETPC И SMARTPHONE

Целью работы является исследование возможностей новой среды разработки Microsoft Visual Studio 2005 и проектирование системы универсального удалённого управления ПК с использованием мобильных устройств на базе Windows CE/Windows Smartphone. Эта система должна обеспечивать взаимодействие между устройствами путём обмена командами, результатами их выполнения и информационных сообщений.

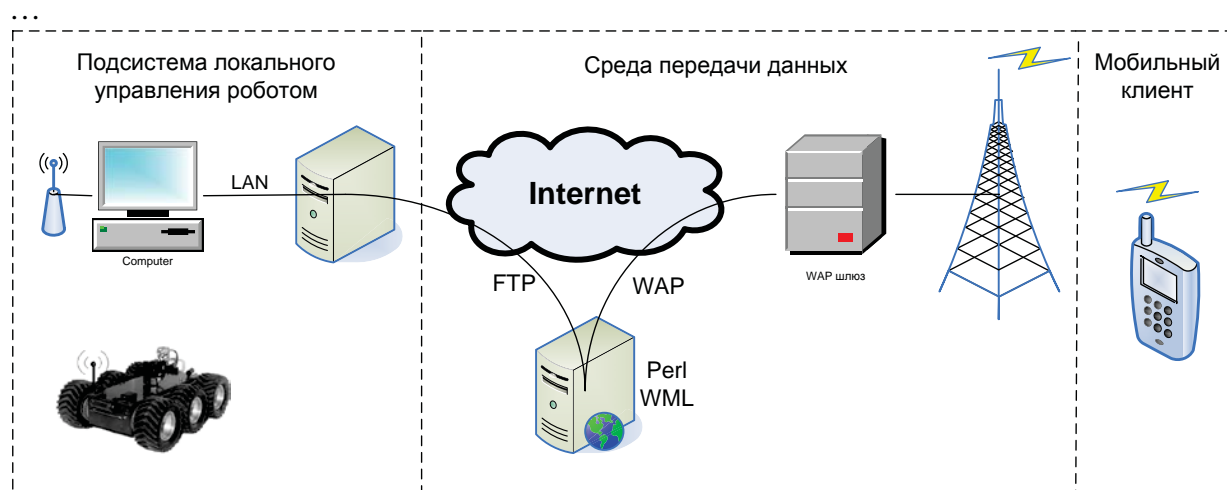


Рис.1 Система дистанционного управления роботом с помощью мобильного телефона.

...

Особенности реализации системы управления:

Сетевое взаимодействие реализовано на базе технологии Wi-Fi, как наиболее удобной в программировании. Однако далеко не все мобильные устройства оснащены необходимым оборудованием, поэтому система был разработан отдельный модуль для работы с Bluetooth. В перспективе предполагается полностью абстрагироваться от технологии связи и предоставить возможность пользователю выбирать её самостоятельно.

В качестве формата передачи команд и данных используется XML. Для работы с этой формой представления данных используется система чтения XML DOM, доступная в Comcast Framework. Важно обеспечить относительно простую схему XML для передаваемых данных, чтобы её можно было разбирать на мобильных устройствах в режиме он-лайн без видимых задержек.

IV. Методика проведения презентаций и выступлений.

В процессе сбора материалов на конкурс-конференцию конкурсная комиссия отбирает лучшие работы для представления презентаций и выступлений, о чем участник конкурса ставится в известность.

1. Порядок проведения выступления на конкурсе-конференции.

1.1. В ходе конференции используется проектор Multi Media.

1.2. Для доклада при помощи Microsoft PowerPoint готовится презентация.

1.3. Презентация должна содержать не менее 10 слайдов. Обязательными являются следующие слайды:

- Титульный, на котором должна быть представлена следующая информация:
 - Название проекта.
 - Фамилия и имя докладчика;
 - Учебное заведение, которое он представляет;
 - Фамилия И.О. научного руководителя.
- Область применения проекта.
- Какие задачи решаются проектом.
- 2 слайда на описание концепции проекта.
- 2 слайда на особенности реализации проекта.
- Представление полученных результатов проекта.
- Перечень использованных для реализации продуктов Microsoft (указать, какие технологии и для чего использовались в проекте).
- Заключение.

1.4. Длительность доклада не должна превышать 12 минут.

V. Положение о системе награждения победителей конкурса – конференции.

Победителей конкурса определяет конкурсное жюри, в состав которого включены ведущие преподаватели вузов. Жюри оценивает представленные материалы и отбирает лучшие из них для участия в работе конференции. Работы будут премироваться следующим образом:

1. Тезисы докладов по всем отобранным комиссией проектам будут опубликованы в сборнике тезисов докладов конкурса-конференции.
2. Победители получают именные дипломы.
3. Лучшие доклады Комиссия будет рекомендовать к опубликованию в сборнике статей, входящем в список ВАК.

Тезисы докладов конкурса-конференции

Секция «Программная инженерия: приложения, продукты и системы»

УДК 004

Е.А. Лихоманова (5 курс, каф. САиИ, СПбГМТУ),
К.В.Пшеничная, к.т.н., доцент

МЕТОДИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ АНАЛИЗА СТЕПЕНИ ЗАЩИЩЕННОСТИ WEB-САЙТОВ

В настоящее время актуальны вопросы анализа защищенности Web-сайтов и разработки методических и программных средств для проведения такого анализа.

Одним из распространенных видов взломов сайта являются SQL-атаки. Любой начинающий «PHP + MySQL» программист сталкивается с таким видом взлома. Со временем, анализируя код, программист старается его оптимизировать и минимизировать, забывая порой о такой элементарной вещи, как защита.

В работе предложены методические и программные средства, помогающие определить степень защищенности Web-сайта и выбрать дополнительные способы защиты на стороне клиента.

В работе представлен Web-сайт и база данных, позволяющие имитировать SQL-атаки с целью проверки используемых способов защиты.

Внедрение SQL, в зависимости от типа используемой СУБД и условий внедрения, может дать возможность злоумышленнику выполнить произвольный запрос к базе данных (например, получить информацию о структуре базы, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), а также получить несанкционированную возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

Рассматриваются следующие виды атак:

- атака через формы авторизации,
- атака через формы регистрации,
- атака через строку поиска,
- атака через Cookies,
- атака через URL строку.

Имитация таких атак позволяют выявить уязвимости в коде защиты Web-сайта.

Способы защиты представлены в виде рекомендаций по созданию и ведению Web-сайтов, а также программных кодов, написанных на языке JavaScript и PHP.

Представленные методические и программные средства могут быть использованы для проведения лабораторных работ в университете.

В работе для целей документирования был использован MS Office.

УДК 004.5

П.Ю. Ковыршин (4 курс, каф. ИМО, ПетрГУ)

А.С. Вдовенко (5 курс, каф. ИМО, ПетрГУ)

Д.Ж. Корзун, к.ф.-м.н., доцент ПетрГУ

МОБИЛЬНЫЕ КЛИЕНТЫ ДЛЯ СЕРВИСОВ ИНТЕЛЛЕКТУАЛЬНОГО ЗАЛА

Система интеллектуального зала (ИЗ) предназначена для проведения научных и учебных мероприятий (напр., конференции). Программная часть реализуется на платформе Smart-M3 [1]. Сервисы системы разворачиваются на вычислительном оборудовании в аудитории (рис. 1). Пользователь получает доступ к сервисам через мобильного клиента, запущенного на персональном устройстве [2] (напр., смартфон).

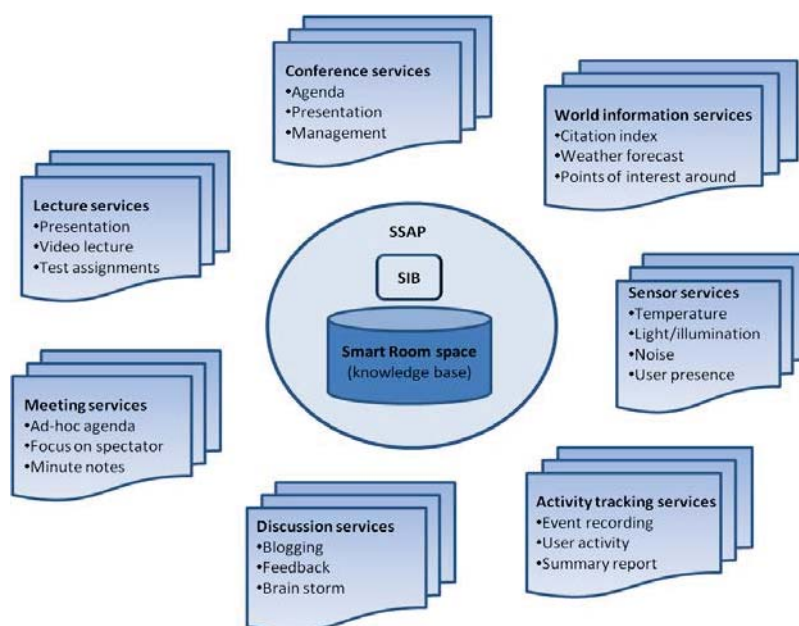


Рис. 1. Структура сервисов ИЗ.

Клиент взаимодействует с семантическим информационным брокером, который управляет хранением разделяемых в системе знаний. Пользователь получает информацию от сервисов на дисплее мобильного устройства. Клиент взаимодействует с сервисами Agenda-service (программа выступлений) и Presentation-service (презентация материалов докладчика). Поддерживается функция предварительной регистрации пользователя через веб-сервис ИЗ. Без предварительной регистрации пользователь может участвовать в качестве гостя (ограниченный доступ к сервисам). Клиент позволяет просматривать программу выступлений, профили участников и редактировать личные данные. Выступающий участник через клиента

управляет показом презентации и может использовать свое персональное устройство в качестве микрофона.

Получены реализации клиента для мобильных платформ Windows Phone, Android и Symbian. Реализация для Windows Phone использует технологию WPF. Реализация для Android использует SDK и NDK инструменты разработки. Реализация для Symbian использует программный каркас Qt.

ЛИТЕРАТУРА

1. D. Korzun, I. Galov, S. Balandin. Smart Room Services on Top of M3 Spaces. Proc. 14th Conf. Open Innovations Framework Program FRUCT. Helsinki, Finland, 11-15 Nov. 2013. pp. 37-44.
2. A. Vdovenko, S. Marchenkov, D. Korzun. Mobile Multi-Service Smart Room Client: Initial Study for Multi-Platform Development. Proc. 13th Conf. Open Innovations Framework Program FRUCT and 2nd Regional Seminar on e-Tourism. Petrozavodsk, Russia, 22-26 Apr. 2013, pp. 143-152.

УДК 004.67

О.Д. Елесина (5 курс, каф. ИУС, СПбГПУ),
С.Ю. Когин (6 курс, каф. ИУС, СПбГПУ),
Д.Ф. Дробинцев (ст. преподаватель каф. ИУС, СПбГПУ),
И.В. Кауров (ст. инженер-программист ООО «DECOSP»)

РАЗРАБОТКА ПОДСИСТЕМЫ ПЛАНИРОВАНИЯ ЗАГРУЗОК БАНКОМАТОВ НА ОСНОВЕ MICROSOFT ANALYSIS SERVICES

Целью проекта является разработка подсистемы планирования загрузок сети банкоматов. Особенностью такой системы является адаптивное вычисление прогноза спроса на денежную наличность в рамках аналитического хранилища данных в среде Microsoft Analysis Services.

Подсистема должна формировать по расписанию краткосрочные прогнозы спроса на денежную наличность в банкоматах и по требованию пользователя выдавать ближайшие дни загрузки в рамках заданного набора ограничений.

В докладе рассматриваются вопросы построения моделей прогноза, использующих встроенные алгоритмы DataMining среды Analysis Services и собственный алгоритм на языке MDX, а также сравнение полученных прогнозов.

Оптимальный прогноз вычисляется в рамках сборки CLR с подключенным к ней OLAP-кубом [Рис.1].

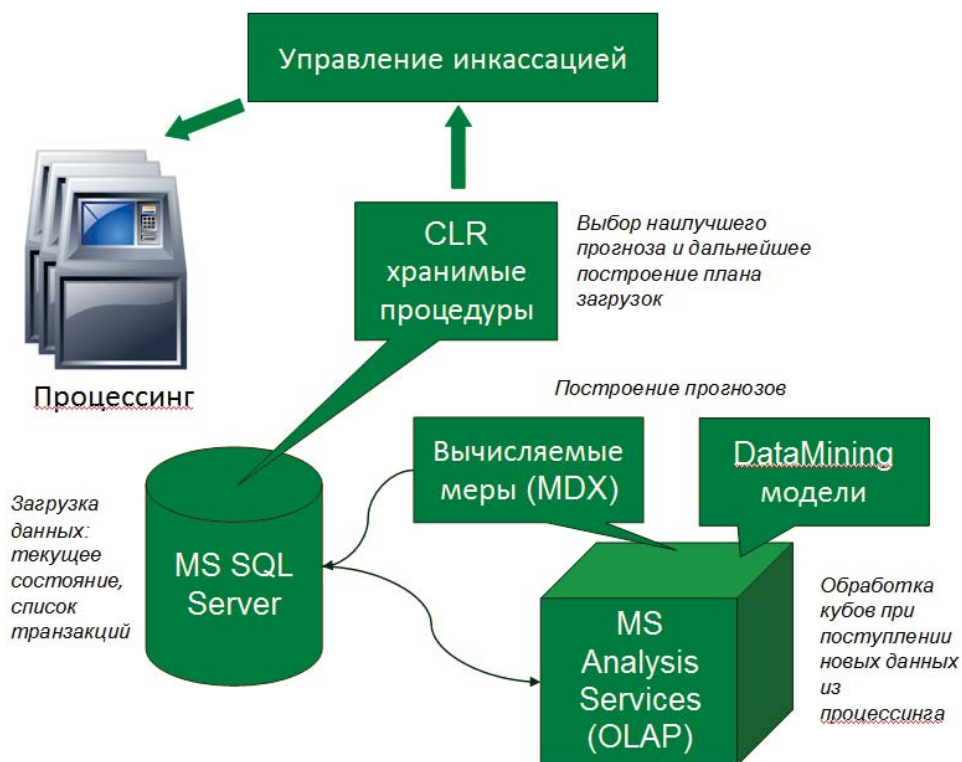


Рис.1 Подсистема планирования загрузок сети банкоматов

Сборка CLR вызывается по расписанию. Происходит выбор оптимальной модели прогноза на текущий день по каждому банкомату, затем импортируется вектор прогнозных значений, строится план и происходит загрузка результатов в реляционную БД.

УДК 004.4+004.7

А. А. Андреев (3 курс, кафедра ИМО, ПетрГУ),
Ю. А. Богоявленский, к.т.н., доцент

ПОСТРОЕНИЕ ГРАФА ИКТ-ИНФРАСТРУКТУРЫ ПРЕДПРИЯТИЯ НА КАНАЛЬНОМ УРОВНЕ С УЧЕТОМ ВИРТУАЛЬНЫХ ЛОКАЛЬНЫХ СЕТЕЙ

Экспериментальная платформа Nest [1], разрабатываемая на кафедре ИМО ПетрГУ, предназначена для исследования моделей и методов управления ИКТ-инфраструктурами поставщиков сетевых услуг (далее Сеть) и эффективности их использования в бизнес-процессах. В настоящее время в подсистеме Nestopo [2], предназначенной для построения графа Сети, реализовано построение графа на сетевом уровне с применением данных получаемых из MIB сетевого оборудования сети по протоколу SNMP.

Целью данной работы является разработка модуля подсистемы NesToro для построения графа Сети на канальном уровне и основанного на нем графа виртуальных локальных сетей, построенных по стандарту IEEE 802.1Q.

Для описания графа Сети в Nest используется объектная модель SON [3]. Текущая версия объектной модели SON позволяет описать конфигурацию виртуальных локальных сетей (далее VLAN) только на уровне одного устройства путем группировки канальных интерфейсов в виртуальные интерфейсы (объект `VlanInterface`). Наш анализ показал необходимость описания групп интерфейсов, принадлежащих одной VLAN. С этой целью мы добавили в модель класс `BroadcastDomain`, описывающий широковещательный домен, и два класса-наследника: `VlanDomain` для описания доменов, образованных VLAN, и `NativeDomain` для описания доменов, которые образованы сегментами Сети, ограниченными устройствами третьего уровня. Принадлежность канальных интерфейсов к домену будем показывать связями между объектом домена и канальными интерфейсами (см. рис. 1).

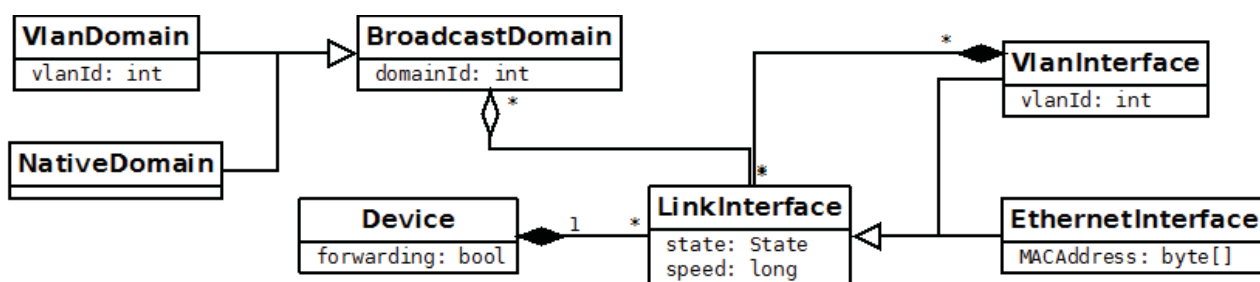


Рис. 1

Существующие в настоящее время технологии обнаружения сетевых устройств на канальном уровне, такие как CDP и LLDP (IEEE 802.1AB), могут быть использованы для построения точного графа канального уровня только в случае, если они полноценно поддерживаются всеми устройствами Сети. Помимо этого, существуют [4,5,6] методы и алгоритмы определения топологии канального уровня на основе различной косвенной информации (в том числе MIB SNMP), которые, однако, не учитывают присутствие в сети VLAN (либо требуют этого в обязательном порядке), а так же полагаются только на один источник информации.

Алгоритм построения графа канального уровня и связанного с ним графа VLAN, разработанный нами в этом исследовании, предполагает использование в качестве источников данных MIB протоколов STP (IEEE 802.1D) и CDP: BRIDGE-MIB (таблицы `dot1dStp` и `dot1dTrFdb`) и CISCO-CDP-MIB (таблица `cdpCacheTable`), а так же таблицы трансляции адресов из RFC1213-MIB и IP-MIB. Информация по VLAN извлекается из таблиц различных MIB (в зависимости от наличия), включая CISCO-VLAN-MEMBERSHIP-MIB, CISCO-VTP-MIB и Q-BRIDGE-MIB.

Алгоритм выполняет следующие шаги: устанавливаются связи путем последовательного обхода устройств и нахождения для их канальных интерфейсов назначенных мостов (в терминологии STP, IEEE 802.1D); в случае отсутствия или недоступности STP-информации устанавливаются связи на основе данные CDP с устранением возможных неоднозначностей; в случае недоступности CDP производится поиск возможных связей через таблицы трансляции; далее производится получение информации о VLAN, которым принадлежат интерфейсы устройства; создаются интерфейсы VLAN и связываются с интерфейсами канального уровня; создаются объекты `VlanDomain` для тех номеров VLAN, для которых нет связи с уже обработанными соседними устройствами; происходит добавление интерфейсов VLAN в

соответствующие домены; объединяются домены с одинаковыми номерами VLAN, в которые входит хотя бы один одинаковый интерфейс канального уровня.

STP-информацию необходимо получать отдельно для каждой VLAN, которой принадлежит устройство, причем информация в пределах этого устройства может быть противоречивой, поэтому следует проводить анализ полученных данных.

В ходе нашего исследования получены следующие результаты: разработан и реализован в рамках ЭП Nest модуль построения графа канального уровня Сети; внесены изменения в модель SON, отражающие специфику технологии VLAN и широковещательных доменов вообще; разработан и реализован в рамках ЭП Nest модуль построения графа VLAN с учетом внесенных в SON изменений.

В подсистему NesToro было добавлено 9 классов, 7 существующих классов подверглись изменению. Общее количество новых строк кода составляет 832, включая 106 строк комментариев.

Тестирование и опытная эксплуатация системы выполняются в вычислительной сети Петрозаводского государственного университета.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. Богоявленский Ю. А. Прототип экспериментальной платформы Nest для исследования моделей и методов управления ИКТ-инфраструктурами локальных поставщиков услуг Интернет // Программная инженерия. — 2013. — № 2. — С. 11–20.
2. Колосов А. С., Богоявленский Ю. А. Параллельный алгоритм построения графа ИКТ-инфраструктуры интернет-провайдера // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — 2013. — № 3. — С. 105–110.
3. Богоявленский Ю. А., Колосов А. С. Организация и автоматизированная поддержка объектной базы данных графа ИКТ-инфраструктуры поставщика услуг Интернета // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — 2011. — № 3. — С. 27–36.
4. Myung-Hee Son, Bheom-Soon Joo, Byung-Chul Kim, and Jae-Yong Lee. Physical Topology Discovery for Metro Ethernet Networks. ETRI Journal, Volume 27, Number 4, August 2005
5. Hassan Gobjuka and Yuri J. Breitbart. Ethernet Topology Discovery for Networks With Incomplete Information. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 18, NO. 4, AUGUST 2010
6. Hassan Gobjuka Topology Discovery for Virtual Local Area Networks. IEEE INFOCOM 2010

РЕКОМЕНДАЦИИ МУЗЫКИ НА ОСНОВЕ ГРАФА ВКУСОВ

Задача рекомендательных систем часто формулируется как предоставление пользователю интересного контента в наиболее подходящее для этого время. В большинстве случаев это делается за счёт анализа взаимодействия пользователей системы с предметами рекомендаций, а также сопоставления свойств самих объектов и знаний о пользователе. Закономерности и факты, полученные таким образом, могут быть интуитивно понятным образом представлены в виде *графа вкусов*, который рассматривается в данной работе. Пользователи и предметы рекомендаций представляются в таком графе вершинами, а отношения между ними в виде дуг. Пользователи *слушают* музыкальные треки и исполнителей, исполнители *создают* треки, треки *похожи* на другие треки – всё это примеры отношений, которые можно поместить в граф вкусов в виде дуг, вес которых зависит от степени связанности двух сущностей и статистической достоверности этой связи.

Построив взвешенный граф сущностей, мы можем решать задачу поиска *степени связанности* двух несмежных вершин в графе, как например пользователя и неизвестный ему трек, оценивая тем самым релевантность трека. Это можно делать с помощью простых метрик, таких как вес кратчайшего пути между ними или количество таких путей. Наиболее же подходящим способом является использование алгоритма *Random Walk with Restarts*, моделирующего случайный обход графа с вероятностью пойти по дуге пропорциональной весу этой дуги. Этот алгоритм хорошо зарекомендовал себя при информационном поиске.

В данной работе с помощью графа вкусов предлагается решать некоторые задачи, связанные с рекомендациями. Задачу в классической постановке (поиск релевантных треков) можно решить, вычислив степень связанности вершины пользователя с несмежными вершинами в графе. Те из них, что имеют наибольшую величину связанности, можно считать наиболее релевантными. Аналогичным способом можно рекомендовать пользователю не только треки, но и более сложные объекты, такие как музыкальные сборники, достаточно лишь добавить в граф вершину сборника, связанную с находящимися в его составе треками.

Помимо поиска релевантных объектов, описанный подход можно применить для персонализации списков музыкальных объектов. Под персонализацией в данном случае подразумевается переупорядочивание списка с целью поднять более релевантные объекты на первые позиции. Так можно обрабатывать имеющиеся списки треков, увеличивая активность пользователей в отношении представленной музыки.

Специфика музыкальных вкусов большинства людей ставит перед рекомендательной системой задачу кластеризации объектов. В самом деле, в зависимости от контекста использования системы, пользователь может хотеть слушать спокойную или агрессивную музыки, классическую или современную. Смешивание тех и других треков, даже при условии высокой релевантности, может пагубно сказаться на удовлетворённости от использования системы. Для решения этой проблемы используется кластеризация подграфа вкусов, соответствующего выдаче рекомендательной системы.

Предложенные методы позволяют лучше удовлетворять потребности пользователей музыкального сервиса, однако требуют анализа статистики большого объёма. Для этого в данной работе используется MapReduce-фреймворк Apache Hadoop.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 004.738.5: 004.451.55

Е.В.Родионов (студент, каф. ВТиИТ, СПбГМТУ),
Т.С. Горавнева (к.т.н., доцент, каф ВТиИТ, СПбГМТУ)

РАЗРАБОТКА МУЛЬТИМЕДИЙНОЙ ПРОГРАММЫ-ТРЕНАЖЕРА

Назначение работы – создание обучающей программы, которая могла бы использоваться в учебном процессе студентами разных специальностей для самостоятельного освоения дисциплины, связанной с выполнением проектных работ в судостроении и машиностроении. Тренажер должен также способствовать повышению качества подготовки студентов специальности 23100.68 по дисциплине «Инженерная и компьютерная графика».

Было выбрано решение разработать две версии программы-тренажера:

- для локального или домашнего использования с названием TRAINING.HOME.
- для корпоративной сети университета с названием TRAINING.NET.

Сценарий работы программы:

- Ввод фамилии и группы студента;
- Выбор учебного модуля из списка;
- Выбор урока для данного модуля;
- Просмотр и выполнение по порядку учебных материалов и тестов для данного урока;
- Отображение статистики прохождения урока и модуля в целом;
- Сохранение статистики для каждого студента.

Исходные данные работы программы-тренажера:

- Список учебных модулей,
- Список уроков для каждого модуля,
- Видеолекции, тесты, графическая информация по уроку.

Особенности разработки программы TRAINING.HOME следующие:

- 1) Хранение файлов формирования тестов в отдельных папках по каждому уроку.
- 2) Файл с вопросами и ответами теоретического теста шифруется по выбранному алгоритму.
- 3) Программа теоретического тестирования создается в виде шаблона, в который подставляются данные из расшифрованного текстового файла с вопросами и ответами.
- 4) Дизайн программы должен соответствовать современному стилю интерфейса программ.
- 5) Статистика по работе с тренажером представлена в текстовом и графическом видах:
 - Текстовый вид – отчет с процентами правильного выполнения уроков и учебного модуля,
 - Графический вид – диаграммы выполнения уроков и модуля.

Представленное приложение разработано на языке C# и использует среду разработки Visual Studio Express 2010.

УДК 681.3.06

А.Н.Цыбуляк (4 курс., каф. УП, СПбГПУ),
Н.Б. Культин, доц., к.т.н. СПбГПУ

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАСЧЕТА ИННОВАЦИОННОГО ПОТЕНЦИАЛА ПРЕДПРИЯТИЯ

В условиях современной экономики эффективного управления предприятием, вне зависимости от его размера, сферы деятельности, номенклатуры выпускаемой продукции, необходимо учитывать многие характеристики предприятия, в том числе и инновационный потенциал.

Целью исследования является повышение эффективности работы предприятия за счет автоматизации процесса определения инновационного потенциала.

Инновационный потенциал предприятия характеризуется следующими показателями:

- Инновационность продукции предприятия;
- Инновационная активность предприятия;
- Инновационность экономики предприятия;
- Инновационное развитие предприятия на рынках.

Наиболее сложным (трудоемким) для расчета является показатель уровня инновационности экономики предприятия. Здесь необходимо большое количество исходных данных и рутинной работы по расчету промежуточных данных. Расчет производится по формулам:

$$\text{УИЭ} = [(\text{рп}(T-1)/\text{рп}(T)) - 1] * 100\%$$

$$\text{рп}(T-1) = \text{РП}(T-1)/\text{ДС}(T-1)$$

$$\text{рп}(T) = \text{РП}(T)/\text{ДС}(T)$$

$$\text{РП} = \frac{\text{ОФ} + \text{ОС} + \text{ЗП}}{\text{В}}$$

$$\text{ОФ} = \text{Фп} + \frac{\text{Фвв} * \text{ИМ}}{12} - \frac{\text{Фл}(12 - \text{М})}{12}$$

$$\text{ОС} = \text{Снг} + \frac{\text{М1}}{12} * \text{Сввед} - \frac{\text{М2}}{12} * \text{Свыб}$$
$$\text{ДС} = \text{СРП} - \text{СЗ}$$

где: РП – ресурсоёмкость; ОФ – среднегодовая стоимость основных фондов; Фп – первоначальная стоимость основных фондов; Фвв – стоимость введенных фондов; ИМ – число месяцев функционирования введенных основных фондов; Фл – ликвидная стоимость; М – количество месяцев функционирования выбывших основных фондов; ОС – среднегодовая стоимость основных средств; Снг – стоимость на начало года; М1 и М2 – число полных месяцев с момента ввода или выбытия (групп объектов) до конца года; Сввед – стоимость введенных в течение года основных средств; Свыб – стоимость выбывших в течение года основных средств; ЗП – сумма заработной платы работников; В – сумма выручки; ДС – добавленная стоимость; СРП – стоимость реализации продукции; СЗ – стоимость затрат всех материальных компонентов исходного сырья и амортизационных отчислений от используемости при производстве оборудования.

Главной задачей является сокращение затрат на получение (расчет) показателя инновационности экономики предприятия, повышение достоверности расчета, уменьшение вероятности получения неверного результата, что можно осуществить за счет автоматизации.

Решением задачи является разработанная авторами прикладная программа (приложение) для автоматизации процесса расчета уровня инновационности экономики предприятия. Приложение создано в среде в Microsoft Visual Studio C# 2013 Express, справочная система – в Microsoft HTML Help Worksshop.

Приложение содержит:

- рабочие формы для ввода исходных данных и расчета промежуточных результатов;
- информационную форму «О программе»;
- справочную систему.

Данное приложение позволит записывать полученные пользователем результаты в базу данных Microsoft Access 2010 (рис.1.), а также выводить отчет на печатающее устройство. Данные функции позволят облегчить процедуру проведения анализа данных, полученных в результате работы с программой.

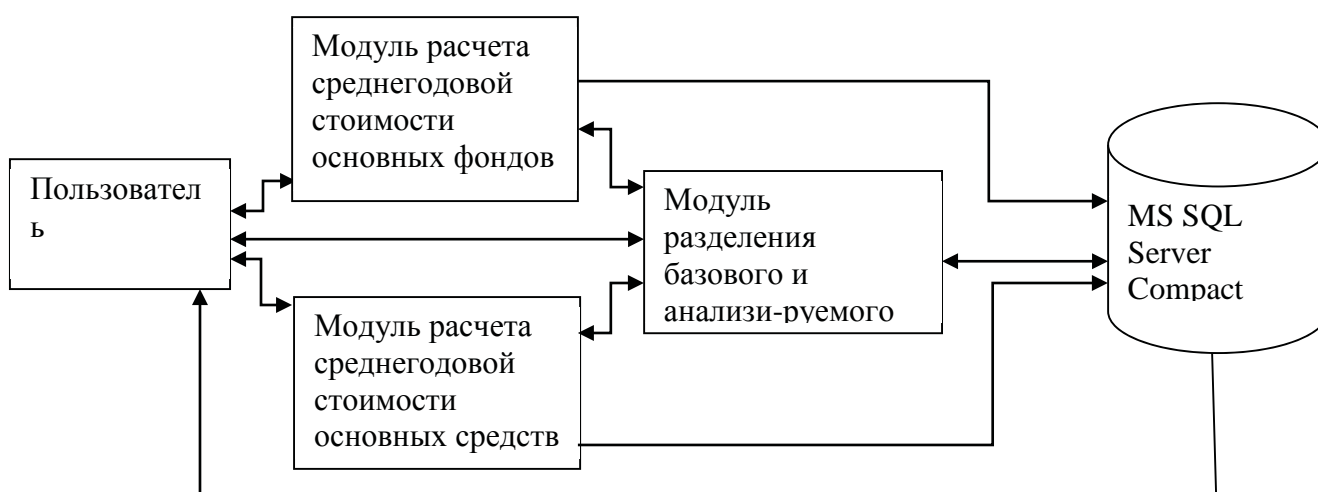


Рис. 1. Архитектура приложения

Программа спроектирована так, что для получения результата пользователю не надо выполнять никакой предварительной («ручной») обработки информации, достаточно только ввести в формы исходные данные:

- стоимость основных фондов на начало года;
- стоимость введенных фондов;
- ликвидную стоимость;
- число месяцев введенных основных фондов;
- число месяцев функционирования выбывших основных фондов;
- стоимость основных средств на начало года;
- стоимость введенных в течение года основных средств;
- стоимость выбывших в течение года основных средств;
- число полных месяцев с момента ввода основных средств, до конца года;
- число полных месяцев с момента выбытия основных средств до конца года;
- сумму заработной платы работников;
- сумму выручки;
- стоимость реализации продукции;
- стоимость затрат материальных компонентов.

Программа имеет простой, интуитивно понятный, интерфейс (рис. 2, рис. 3).

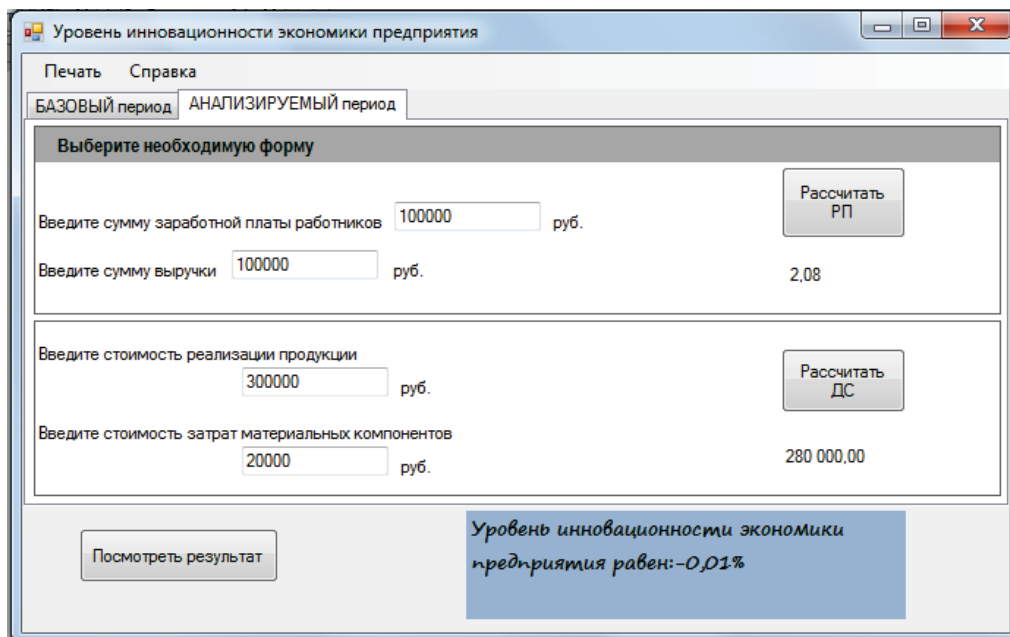


Рис. 2. Вид рабочего окна

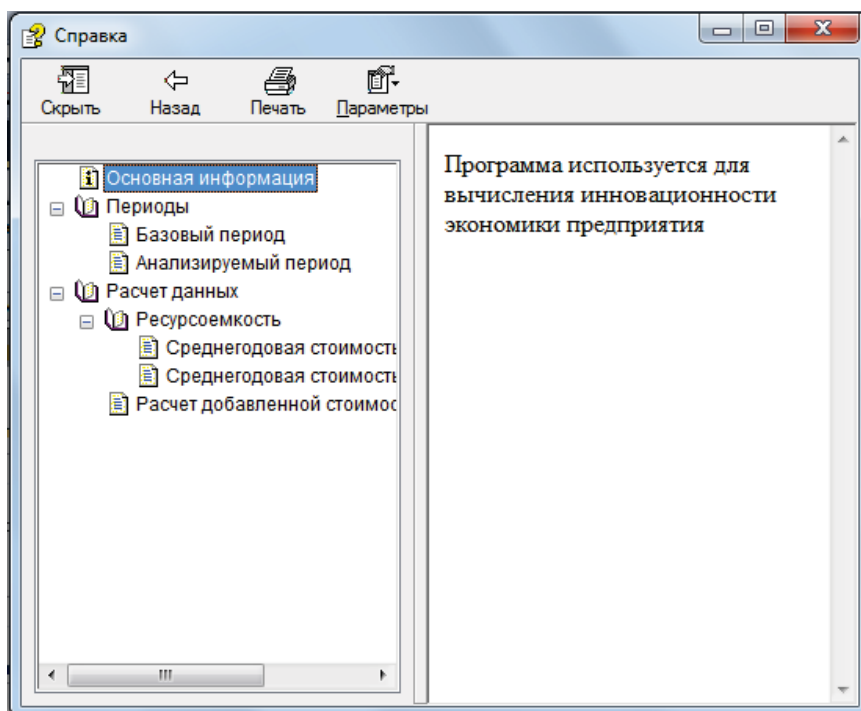


Рис. 3. Окно справочной информации

Важно отметить, что данный прототип программируется не с помощью языка C++, а с помощью более прозрачного и понятно языка C#, что обусловлено тем, что приложение призвано быть простым, формализованным и потреблять как можно меньше ресурсов. Этому же принципу отвечает и справочная система, написанная с помощью HTML-кода.

Выводы: предлагаемое решение позволит повысить эффективность управления предприятием за счет сокращения затрат на расчет показателя инновационности экономики

предприятия, а разработанный прототип показывает возможность реализации поставленной задачи. Приложение может использоваться на предприятиях, а также всеми заинтересованными лицами.

УДК 004.4

А.А. Янушевский (4 курс каф.ИУС), П.Д. Дробинцев доц. СПбГПУ

ОПЫТ ИСПОЛЬЗОВАНИЯ ВНЕШНИХ СРЕДСТВ РАЗРАБОТКИ СРЕДЫ ECLIPSE ДЛЯ СОЗДАНИЯ ПРИЛОЖЕНИЙ, РАБОТАЮЩИХ С БОЛЬШИМИ ОБЪЕМАМИ ДАННЫХ

В настоящее время огромные объемы информации с классических носителей — книг, документов, фотокарточек и т.д. хранятся в электронном формате при этом для простоты структурирования и использования данной информации её обычно разбивают по различным группам и категориям, с помощью моделей хранения данных. Проблема использования моделей данных заключается в том, что модель содержит в себе десятки, а иногда и сотни классов и огромное количество связей между ними. В силу этих обстоятельств создание и отладка такой модели, с помощью методов классического программирования, являются чрезвычайно трудоемкими задачами. Для решения этой проблемы могут быть использованы специализированные средства, входящие в состав современных пакетов разработки программного обеспечения, в данной статье будет рассмотрено средство Eclipse Modeling Framework(EMF). EMF — технология, для генерации кода модели на основе метамодели, которая может быть использована при разработке приложений в среде Eclipse. Для построения метамодели, на базе технологии EMF используется специализированный редактор Diagram Editor For Ecore, который позволяет существенно сократить трудозатраты, за счет использования графической нотации с последующей генерацией кода самой модели[1].

В силу широкого распространения технологии Eclipse не составляет труда обнаружить в открытых источниках информацию о подключении инструментария к среде Eclipse и описании действий от создания метамодели и до генерации кода модели[2]. При этом в рамках технологии также достаточно просто реализовать процессы заполнения и получения информации из полей классов. На рис. 1 приведен пример графического представления модели данных EMF:

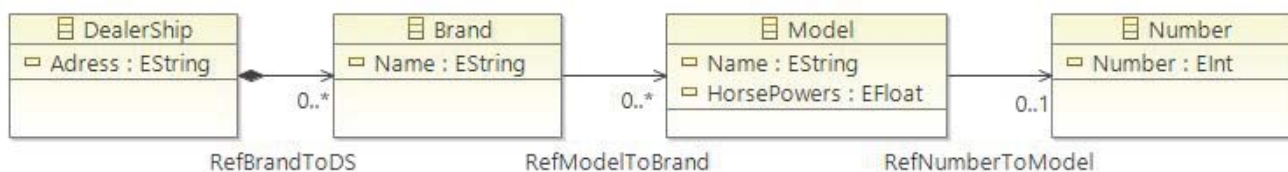


Рис. 1. Графическое представление EMF модели

На основе опыта использования технологии EMF можно сделать вывод о том, что EMF является достаточно мощным инструментом для создания моделей данных на основе метамодели, который позволяет не только сэкономить трудозатраты, но и повысить качество моделей данных, лежащих в основе разрабатываемых приложений.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. D. Steinberg, F. Budinsky, M. Paternostro, E. Merks. EMF: Eclipse Modeling Framework, 2nd Edition — USA, Addison-Wesley Professional, 2008, 744 с.
2. <http://www.vogella.com/articles/EclipseEMF/article.html>

УДК 004.4

С.А. Марченков (5 курс, каф. ИМО, ПетрГУ)
Д.Ж.Корзун, к.ф.-м.н., доцент

РАСПОЗНАВАНИЕ ПРИСУТСТВИЯ МОБИЛЬНЫХ УСТРОЙСТВ В ИНТЕЛЛЕКТУАЛЬНОМ ЗАЛЕ

Целью данной работы является разработка подсистемы распознавания присутствия персональных мобильных устройств и ее интеграция в систему интеллектуального зала (ИЗ). Технология Innorange Footfall (www.innorange.fi/) основана на пассивном радиообнаружении в беспроводной локальной сети. Выделенный сетевой сенсор выполняет мониторинг трафика мобильных устройств по MAC-адресам. В работе предложены сценарии использования этой технологии, их реализация, интеграция в систему ИЗ и экспериментальное исследование.

Система ИЗ [1] представляет многоагентное приложение, работающее на оборудовании в пределах аудитории и с внешними Интернет-сервисами. Совместная работа агентов реализует сервисы для автоматизации проведения коллективных мероприятий. Инфраструктура для агентов ИЗ — разделяемое «интеллектуальное пространство» реализована на платформе Smart-M3 (en.wikipedia.org/wiki/Smart-M3). Участники мероприятия используют свои персональные мобильные устройства (напр., смартфоны Windows Phone). Распознавание присутствия таких устройств позволяет идентифицировать конкретных участников. Рассматриваются следующие типы сценариев для системы ИЗ.

(1) Перед началом мероприятия участники собираются в аудитории, информация о присутствии используется для подготовки к мероприятию. (2) В ходе мероприятия информация о присутствии используется как контекст для сервисов ИЗ (напр., отслеживание отсутствующих и модификация программы мероприятия). (3) Анализ истории присутствия для подведения итогов мероприятия (напр., выделение «самых активных» участников). Предлагается следующая архитектура подсистемы определения присутствия (рис. 1).



Рис. 1. Архитектура подсистемы обнаружения присутствия в системе ИЗ.

Сенсор определяет присутствие устройства на основе измерения силы принимаемого сигнала (RSSI — «received signal strength indication») [2]. Агент сенсора реализует: (а) HTTP сервер для получения от сенсора MAC-адреса устройства и значение RSSI и (б) публикацию измерений в интеллектуальное пространство ИЗ. Агент определения присутствия подписывается на обновление этих данных и реализует семантическую связь мобильного устройства и его владельца, позволяя формировать в интеллектуальном пространстве актуальную информацию о присутствии участников. Эта информация далее используется сервисами для рассмотренных сценариев. Реализованы несколько демо-сценариев, на основе которых проведены эксперименты для измерения производительности.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. D. Korzun, I. Galov, S. Balandin. Smart Room Services on Top of M3 Spaces. Proc. 14th Conf. Open Innovations Framework Program FRUCT. Helsinki, Finland, 11-15 Nov. 2013. pp. 37-44.
2. S. Marchenkov, D. Korzun. User Presence Detection in SmartRoom Using Innorange Footfall Sensor. Proc. 14th Conf. Open Innovations Framework Program FRUCT. Helsinki, Finland, 11-15 Nov. 2013. pp. 199-200.

УДК 681.518

В.Э. Ковалевский (асс., каф. СТУ, СПбГПУ)

ИСПОЛЬЗОВАНИЕ СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБМЕНА ДАННЫМИ МЕЖДУ ПРОМЫШЛЕННЫМИ КОНТРОЛЛЕРАМИ SIMATIC S7 И ПЕРСОНАЛЬНЫМ КОМПЬЮТЕРОМ

Управляющими устройствами, применяемыми на производстве и в других отраслях схожих по условию применения и задачам, являются промышленные контроллеры. Основное требование, предъявляемое к промышленным контроллерам, это высокая надежность и возможность работы в жестких промышленных условиях.

Современная система промышленного производства представляет собой сложный объект управления. Зачастую для управления такими объектами принято использовать автоматизированные системы управления, которые в отличие от автоматических оставляют возможность участия человека в отдельных операциях, как в целях сохранения человеческого контроля над процессом, так и в связи со сложностью или нецелесообразностью

автоматизации отдельных операций. В качестве пользовательского интерфейса обеспечивающего взаимодействие с автоматизированной системой управления нередко применяется программное обеспечение, работающее на персональном компьютере. Соответственно, такому программному обеспечению необходим способ для обмена данными с промышленными контроллерами, реализующими систему управления.

Одними из самых массово применяемых в настоящее время промышленных контроллеров являются программируемые логические контроллеры (ПЛК) семейства Simatic S7, компании Siemens. Компания Siemens предусмотрела несколько различных способов обмена данными между своими контроллерами и персональным компьютером. Существует, как минимум, четыре способа реализации такого взаимодействия[1]. В таблице 1 приводится сравнение основных особенностей этих четырех способов.

Таблица 1. Сравнительные характеристики способов организации связи.

	С помощью SIMATIC PRODAVE	С помощью OPC-сервера	Посредством S7 API	С использованием LibNoDave
Условия использования	Лицензионный продукт Siemens	Обычно платный лицензионный продукт	Входит в состав платного пакета SIMATIC NET	Бесплатно распространяемая библиотека
Поддерживаемые сети	MPI, Profibus, Industrial Ethernet	MPI, Profibus, Industrial Ethernet	MPI, Profibus, Industrial Ethernet	MPI, Profibus, Industrial Ethernet
Ответная часть в контроллере	Для простых задач не требуется	Не требуется	Не требуется	Требуется для реализации диагностики
Параллельный опрос устройств	Не поддерживает	Поддерживается	Поддерживается	Не поддерживается
Простота использования	Прост в применении	Просты в применении	Сложен, требует высокой квалификации	Прост в применении

Часто для организации взаимодействия между пользовательскими приложениями и ПЛК используется технология OPC (OLE for Process Control). OPC взаимодействие основано на клиент-серверной схеме. Такой способ взаимодействия является унифицированным, пользователю нет необходимости привязываться к деталям реализации конкретного контроллера, поскольку за непосредственную связь с ПЛК в такой схеме отвечает OPC-сервер. Однако такой способ взаимодействия имеет и ряд существенных недостатков. Так существует необходимость в использовании дополнительного программного продукта, OPC-сервера. Такие сервера разрабатываются рядом производителей, и все они являются лицензионными платными продуктами. Кроме того такие сервера имеют существенные ограничения по объему передаваемых данных, что затрудняет их использование в реальных условиях работы.

В связи с этим было решено проверить возможности свободно распространяемой библиотеки libNoDave[2] для обмена данными между пользовательским приложением на языке высокого уровня и ПЛК. В целях сравнения с сайта разработчика была скачана последняя версия библиотеки libNoDave (версия 0.8.5) и на её основе в среде Visual Studio 2010 было разработано пользовательское приложение на языке C# для обмена данными между

персональным компьютером и ПЛК. Также было разработано приложение для обмена данными с ПЛК посредством технологии OPC. В качестве OPC-сервера было создано приложение в среде SIMATIC WinCC Flexible 2007, а пользовательское приложение, являющееся OPC-клиентом, было основано на коде, представленном на официальном сайте компании Siemens[3] и также написано на C#.

Для проверки работоспособности приложений был взят ПЛК S7-300 модели CPU 315F-2 PN/DP, имеющий встроенный Ethernet порт, и ноутбук DELL LATITUDE D620. Устройства были соединены между собой по Ethernet. Оба пользовательских приложения были запущены и производили в цикле операции чтения/записи в память контроллера. Быстродействие обеих программ приведено на рисунке 1.

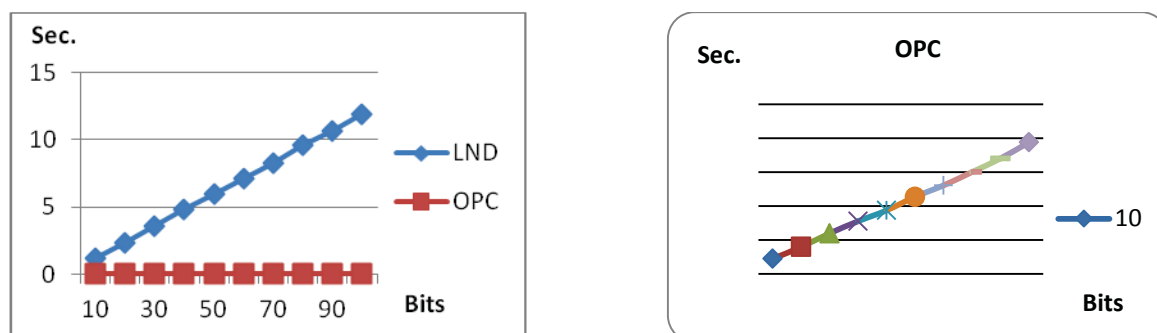


Рисунок 1. Сравнение производительности LibNoDave и OPC

В обоих случаях тесты показали практически линейную зависимость скорости работы от объема передаваемой информации, однако обмен данными по технологии OPC оказался на порядок быстрее. С другой стороны, при увеличении числа запросов за единичный временной интервал обнаружился порог (примерно 100 запросов), после которого приложение использующее технологию OPC не смогло обеспечить необходимого быстродействия и вышло из строя. Приложение же использующее для взаимодействия с ПЛК библиотеку libNoDave продолжило работать, как обычно.

Таким образом, можно сделать заключение, что в случае, когда для обмена данными между персональным компьютером и ПЛК требуется повышенное быстродействие, объем одновременно передаваемой информации невелик, и есть финансовая возможность то лучше использовать технологию OPC. В противном случае более целесообразным представляется использование библиотеки libNoDave.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. Организация обмена данными контроллеров SIMATIC S7 с пользовательскими приложениями по протоколу TCP/IP // Захарченко В.Е., Шерешевский Л.А. / Самарский центр компетенции WinCC. URL: <http://www.sms-automation.ru/about/publications/> (дата обращения: 17.12.2010)
2. LibNoDave - свободно распространяемая библиотека, для коммуникации с ПЛК Simatic S7. URL: <http://libnodave.sourceforge.net/> (дата обращения: 27.01.2014)
3. Programming of OPC DA .NET Clients with C# for the SIMATIC NET OPC Server. URL: <http://support.automation.siemens.com/WW/view/en/21043779> (дата обращения: 20.06.2013)

УДК 004.421+ 519.178

С.Ю. Когинов, 6 курс каф.ИУС ИИТУ СПбГПУ
Е.А. Каурова, 5 курс каф. ИПМ ФКТиУ НИУ ИТМО,
И.В. Кауров, ст. инженер-программист ООО "DECOSP",
Д.Ф. Дробинцев, ст.преподаватель. каф.ИУС ИИТУ СПбГПУ

РАЗРАБОТКА МОДУЛЯ «МАРШРУТИЗАТОР» СИСТЕМЫ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ИНКАССАЦИИ КОММЕРЧЕСКОГО БАНКА

Задача маршрутизации транспорта (ЗМТ) является задачей комбинаторной оптимизации, в которой для парка транспортных средств, расположенных в одном депо, требуется определить набор маршрутов, проходящих через все остальные вершины графа не более одного раза.

Целью проекта является разработка модуля «Маршрутизатор», который должен быть интегрирован в учетную систему автоматизации коммерческого банка.

В рамках проекта на основании анализа технологии работы службы инкассации банкоматов сформулированы требования к модулю, рассмотрены методы решения ЗМТ, построен алгоритм и реализовано соответствующее программное обеспечение.

В качестве примера была выбрана задача построения маршрутов инкассации банкоматов в крупном коммерческом банке.

Выбор методов решения ЗМТ проводился на основании следующих критериев:

1. Качество получаемого решения;
2. Трудоемкость алгоритма;
3. Сложность реализации.

В соответствии с данными критериями выбран и реализован алгоритм разрезания общего маршрута. Алгоритм решает задачу коммивояжера для всех вершин, затем происходит разделение маршрута в соответствии с поставленными ограничениями. После чего в каждый маршрут добавляется депо (кассовый центр), и для каждого набора точек снова решается задача коммивояжера.

Для решения задачи коммивояжера были выбраны жадный алгоритм и алгоритм Лина-Кернигана. Жадный алгоритм является одним из наиболее простых в реализации алгоритмов и работает считанные секунды. Алгоритм Лина-Кернигана имеет трудоемкость, близкую к $O(n^3)$, и известен очень высоким качеством решения. Он относится к эвристическим алгоритмам улучшения маршрута. В качестве исходных данных был использован как случайный путь, так и путь, построенный при помощи жадного алгоритма.

В докладе представлена реализация и приведено сравнение полученных результатов на следующих реальных данных:

1. Точки инкассации представляют собой 175 банкоматов, расположенных по 125 уникальным адресам
2. Маршрут инкассации не может содержать более 20 банкоматов
3. Длительность маршрута инкассации не может превышать 6 часов
4. Среднее время обслуживания банкомата – 10 минут

В качестве основного критерия сравнения результатов была выбрана суммарная длина построенных маршрутов.

В качестве архитектуры клиентского приложения был выбран тонкий клиент. Выбор был обусловлен требованием отображать результаты на картах. Приложение построено по шаблону Model-View-Controller с использованием современных технологий Microsoft Asp.Net

MVC 4 и Entity Framework, позволяющих значительно упростить и ускорить разработку пользовательского интерфейса. Справочник банкоматов и таблица расстояний лежат в базе данных под управлением Microsoft SQL Server.

УДК 621.37+621.391.8

К.С.Кириленко (4 курс ИИТУ СПбГПУ), С.А.Поддорогин (5 курс ИИТУ СПбГПУ),
рук. В.С. Тутыгин, к.т.н., доц. ИИТУ СПбГПУ

РАЗРАБОТКА ПОРТАТИВНОГО АНАЛИЗАТОРА ПАРАМЕТРОВ РАДИОИМПУЛЬСОВ

Задача определения параметров радиоимпульсов, таких как количество периодов, частота, временная задержка, возникает в системах сбора и обработки спутниковых ЛЧМ-сигналов, эхо-сигналов спектрометров, радиолокационных электромагнитно – акустических толщиномеров [1]. К таким системам могут быть отнесены также разнообразные радиотехнические, радиолокационные, гидролокационные системы, системы управления и контроля, электросвязи, телефонии.

Задача вычисления точного значения количества периодов и частоты несущей радиоимпульса или гармонического короткого сигнала весьма актуальна при спектральном анализе сигналов спинового и квадрупольного эха импульсных спектрометров ЯМР и ЯКР. Частотный спектр сигнала в этом случае характеризует состав изучаемого вещества, так как каждое вещество (химический элемент) характеризуется строго индивидуальной резонансной частотой. Частотный спектр сигнала нужно определять максимально точно, поскольку частотные линии разных веществ могут находиться весьма близко. Подобная задача (вычисления точного значения количества периодов и частоты дискретизированного сигнала) возникает в радиолокационных системах, предназначенных для определения скорости перемещения объекта на основе эффекта Доплера. Общей особенностью этих радиосигналов является малое количество периодов сигнала.

Для определения количества периодов и частоты сигнала обычно применяется быстрое преобразование Фурье (БПФ). Результатом БПФ дискретизированного сигнала определенной частоты является количество периодов сигнала. Если частота отсчетов или интервал дискретности по времени при измерении сигнала известен, то по количеству периодов можно установить и частоту измеряемого сигнала. Точность определения частоты в спектре входного сигнала вполне определена и зависит от количества p периодов сигнала [2]. Если количество периодов целое, то частота с помощью БПФ находится абсолютно точно (при отсутствии зашумленности сигнала). Если же количество периодов не является целым, то появляется погрешность определения частоты. Максимальное значение погрешности равняется $1/2p$. В упомянутых практически важных случаях количество периодов анализируемого сигнала находится в пределах от 3 до 10. В этом случае максимальная погрешность определения частоты с помощью БПФ может составлять до 17%.

В [1] теоретически обоснована возможность прецизионного измерения частоты короткого сигнала за счет принципиально другого способа определения частоты, защищенного Патентом РФ на изобретение.

Для широкого практического применения нового способа, а также при решении задачи создания портативного анализатора радиосигналов с помощью ноутбуков, планшетных компьютеров, ПЛИС или цифровых сигнальных процессоров необходима разработка программы на языке C++. Однако реализация способа требует выполнения сложных алгоритмов цифровой обработки сигналов, таких как БПФ, взаимная корреляция и сплайн-интерполяция.

Поэтому в [4] была предложена модификация алгоритма цифровой обработки в целях упрощения аппаратной или программной реализации. Вместо коэффициента корреляции в качестве признака степени сходства двух сигналов используется сумма абсолютных значений отсчетов суммарного сигнала, получаемого при сложении радиоимпульса и эталонного сигналов.

Предлагаемый способ заключается в «подстройке» частоты эталонного сигнала к несущей частоте радиоимпульса и нахождении максимальной нормы их суммарного вектора, которая имеет место при резонансе этих сигналов (рис. 1).

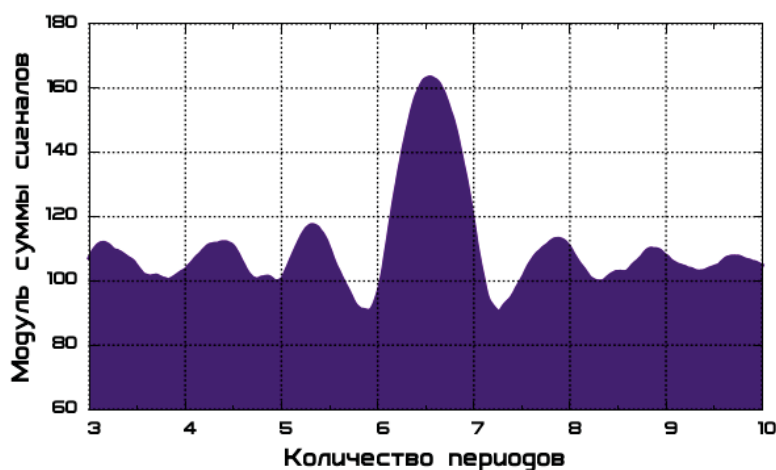


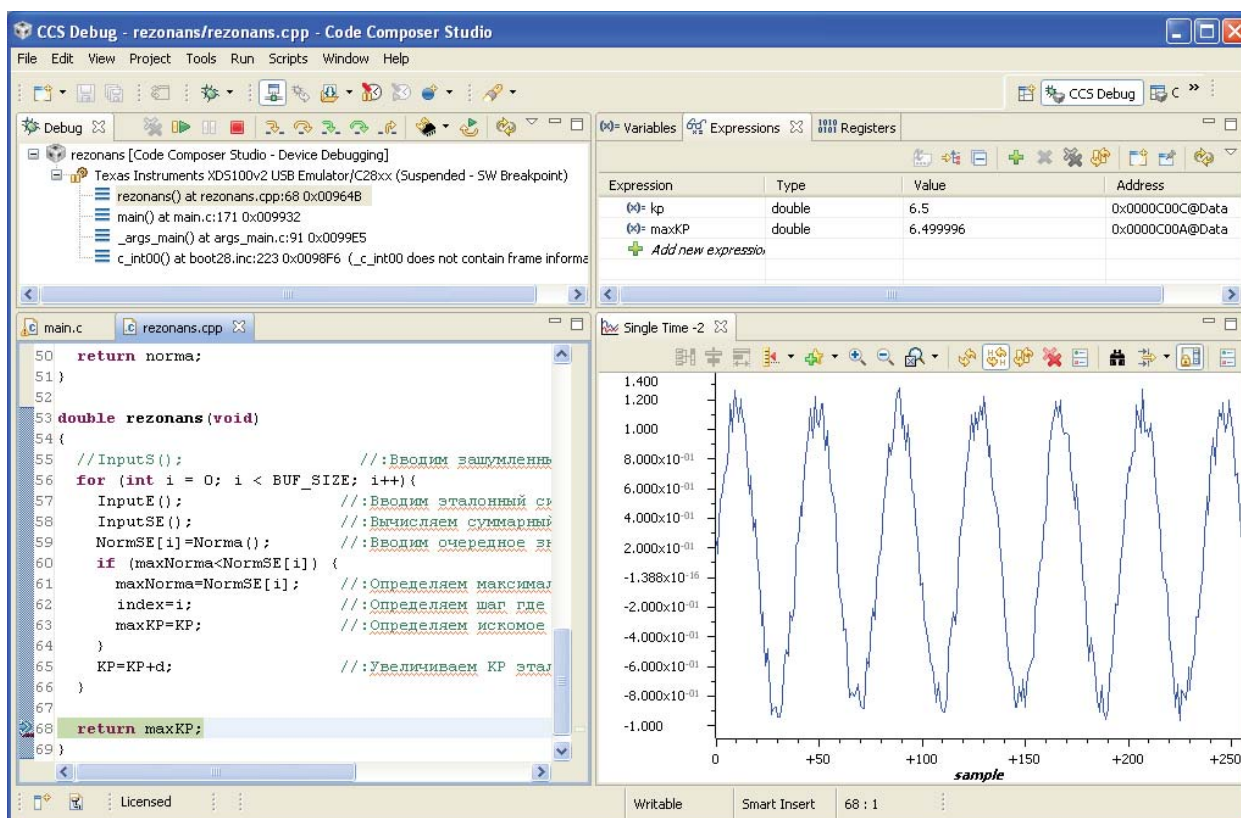
Рис. 2. Нормы суммарных сигналов. Точное значение количества периодов - 6,57, СКО шума - 0,03.

несущей частоты исследуемого радиоимпульса при этом определяется точно.

Первоначально алгоритм был реализован в виде программы на языке C++ в среде Microsoft Visual Studio 2013, а после отладки программа была перенесена в среду Code Composer Studio от Texas Instruments.

При сложении двух сигналов имеющих одинаковую частоту, то есть, находящихся в резонансе, мы получаем график, где все пики графика максимальны, в отличие от других случаев, когда резонанс отсутствует.

Так как количество периодов эталонного сигнала известно в любой момент и при достижении максимальной нормы совпадает с количеством периодов несущей частоты исследуемого сигнала, то количество периодов



В среде Code Composer Studio произведено моделирование исполнения программы с целью дальнейшего создания портативного устройства для анализа параметров радиоимпульсов на базе цифрового сигнального процессора TMS320F28335. Отладка программы произведена с помощью модуля разработчика MChip176-F28335 на основе процессора TMS320F28335.

Использование портативного анализатора параметров радиоимпульсов в импульсных спектрометрах, подобных описанному в [3], позволит значительно повысить точность анализа веществ. Использование технологий Microsoft и Texas Instruments при создании программного обеспечения позволит создать компактные недорогие промышленные системы для экспресс-анализа магнитных материалов, таких как образцы руд, содержащих элементы группы железа, полупродуктов и сырья черной и цветной металлургии.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. В.С.Тутыгин Цифровая обработка коротких сигналов. СПб., Изд-во Политехн. ун-та, 2012.-164с., ISBN 978-5-7422-3723-5.
2. С.Л.Марпл-мл. Цифровой спектральный анализ и его приложения: Пер. с англ. – М.:Мир, 1990 – 584с. ISBN 5-03-001191-9
3. В.И.Тарханов, В.С.Тутыгин Приборный комплекс для поиска и исследования сигналов ЯМР в магнитоупорядоченных веществах.//Научное приборостроение, 2003, том 13, №1, с.58-63, ISSN 0868-5886.

4. С.А. Поддорогин. Способ определения количества периодов и несущей частоты радиоимпульса.
Заявка на изобретение №2014108944.
УДК 621.37+621.391.8

И.А.Пелконен, К.С.Кириленко (3 курс ИИТУ СПбГПУ),
В.С. Тутыгин, к.т.н., доц. СПбГПУ

ПРОГРАММА ОПРЕДЕЛЕНИЯ ВРЕМЕНИ ЗАДЕРЖКИ РАДИОЛОКАЦИОННОГО СИГНАЛА

Задача цифровой обработки коротких сигналов в реальном времени возникает в системах сбора и обработки спутниковых ЛЧМ-сигналов, эхо-сигналов спектрометров, радиолокационных сигналов электромагнитно-акустических толщиномеров [1-2]. К таким системам могут быть отнесены также разнообразные радиотехнические, радиолокационные, гидролокационные, системы управления и контроля, электросвязи, телефонии.

Особенностью сигналов, получаемых в таких и подобных им системах является то, что они относятся к классу коротких и значительно зашумленных. Задача цифровой обработки заключается в обнаружении и определении параметров этих сигналов по одной реализации и в условиях значительных флуктуационных помех.

Для измерения толщины металла в труднодоступных местах, например, при измерении толщины нефтегазовых труб в процессе их эксплуатации разработаны приборы (толщиномеры), использующие различные физические явления. Наибольшую точность измерения обеспечивают электромагнитно-акустические (ЭМА) толщиномеры, использующие принцип измерения времени прохождения ультразвукового импульса, порождаемого радиоимпульсом, через металлическое изделие [1]. Источник радиоимпульса помещается у одной поверхности металлического изделия, и регистрируется сигнал, отраженный от другой поверхности (рис.1). По времени задержки первого отраженного сигнала от зондирующего или временного интервала между двумя соседними отраженными сигналами с учетом известной скорости распространения ультразвуковой волны в металле определяется толщина металла. При коррозии металлических изделий отраженный сигнал оказывается значительно зашумлен, что приводит к уменьшению точности определения временных интервалов между радиоимпульсами, если определять положение радиоимпульса по максимуму амплитуды, и, следовательно, к уменьшению точности определения толщины металла.

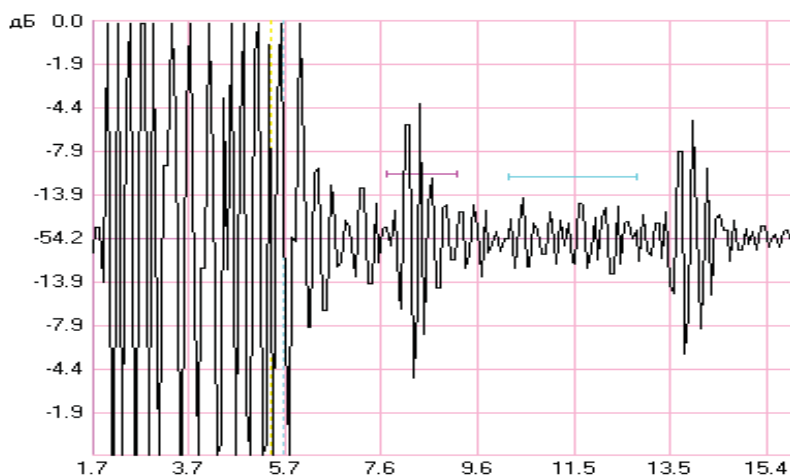


Рис. 1. Зарегистрированный с помощью АЦП с памятью отраженный сигнал ЭМА-толщиномера.

Альтернативный известный способ определения временных интервалов между радиоимпульсами, обеспечивающий более высокую точность измерения основан на использовании автокорреляционной функции (АКФ) отраженного сигнала. Использование АКФ дает возможность, как известно, значительно улучшить отношение сигнал/шум, причем АКФ от периодической функции представляет собой тоже периодическую функцию, имеющую тот же период. В этом случае измеряемым параметром будет длительность периода АКФ. Для вычисления длительности периода традиционно применяется один из двух методов: подсчет количества периодов сигнала во временном окне фиксированной длительности или быстрое преобразование Фурье (БПФ).

Использование традиционного подхода путем определения количества периодов АКФ в фиксированном временном окне определения или БПФ также не позволяет получить точное значение интервала между пиками АКФ и, следовательно, между двумя соседними отраженными сигналами ЭМА толщиномера, так как количество периодов не может быть взято достаточно большим из-за затухания отраженных эхо-сигналов по амплитуде и из-за того, что количество периодов АКФ в фиксированном временном окне может оказаться нецелым. Это порождает ошибку определения количества периодов. Дополнительная погрешность определения количества периодов возникает в связи с наличием флуктуационного шума и шума квантования.

В [2,3] предложен новый способ измерения временных интервалов между двумя радиоимпульсами и в [4] - программа, реализующая предложенный способ в среде MATLAB. Для широкого практического применения нового способа, а также при решении задачи аппаратной реализации способа с помощью ПЛИС необходима разработка программы на C++. Реализация способа требует выполнения сложных алгоритмов цифровой обработки сигналов, таких как БПФ, взаимная корреляция и сплайн-интерполяция, отсутствующих в стандартных библиотеках Visual Studio.

Алгоритм реализован в виде программы на языке C++ в Visual Studio 2012 в среде операционной системы Windows-8. Для выполнения БПФ была использована библиотека FFTW (<http://fftw.org/>, MIT, Matteo Frigo and Steven G. Johnson), распространяемая по лицензии GPL. Для выполнения сплайн-интерполяции была использована библиотека SPLINE (http://people.sc.fsu.edu/~jburkardt/cpp_src/spline/spline.html, John Burkardt, The Florida State University), распространяемая по лицензии LGPL.

Использование программы в ЭМА-толщиномерах, подобных описанному в [1], позволит значительно повысить точность измерения толщины металла. Использование технологий Microsoft при создании программного обеспечения позволит создать компактные недорогие системы для измерения толщины металла на базе планшетных компьютеров и USB – устройств сбора данных, пригодные для работы в полевых условиях.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. В.П.Себко, Г.М.Сучков, Е.А.Алексеев. Оптимизация параметров ЭМА толщиномеров для контроля тонкостенных изделий. - Дефектоскопия. 2002. № 12. С. 21-28.
2. В.С.Тутыгин Способ измерения временных интервалов между радиоимпульсами. Патент РФ на

изобретение №2456632. (приоритетная справка Роспатента от 21.03.2011 Зарегистрирован в Госреестре 20.07.2012г.).

3. В.С.Тутыгин Способ повышения точности измерений для электромагнитно-акустических толщиномеров. В сб.: Высокие технологии и фундаментальные исследования. Т. 2. Сборник трудов Двенадцатой международной научно-практической конференции «Исследование, разработка и применение высоких технологий в промышленности». 2011, Санкт-Петербург, Россия/ под ред. А.П.Кудинова.- СПб.: Изд-во Политехн. ун-та, 2011
4. В.С.Тутыгин. Программа точного определения времени задержки радиолокационного эхо-сигнала. Свидетельство Роспатента об официальной Гос. регистрации №2011612678 от 01.04.2011.

УДК 621.37+621.391.8

И.А.Пелконен, А.В.Мищенко (3 курс ИИТУ СПбГПУ),
рук. В.С. Тутыгин, к.т.н., доц. СПбГПУ

ПРОГРАММА ЧАСТОТНОГО АНАЛИЗА КОРОТКИХ СИГНАЛОВ

Задача вычисления точного значения количества периодов и частоты дискретизированного сигнала весьма актуальна при спектральном анализе сигналов спинового и квадрупольного эха импульсных спектрометров ЯМР и ЯКР. Частотный спектр сигнала в этом случае характеризует состав изучаемого вещества, так как каждое вещество (химический элемент) характеризуется строго индивидуальной резонансной частотой. Частотный спектр сигнала нужно определять максимально точно, поскольку частотные линии разных веществ могут находиться весьма близко.

Спектральный анализ сигналов спинового эха спектрометров ЯМР может быть использован, в частности, в металлургии.

В технологических процессах производства цветных металлов наличие в сырье, полупродуктах и продуктах элементов группы железа является нежелательным. Их удаление зависит от вида химического соединения, в котором они находятся. Поэтому на разных этапах технологического процесса берутся пробы полупродуктов и подвергаются химическому экспресс-анализу. По результатам экспресс-анализа производится коррекция технологического процесса за счет добавления в плавильную печь того или иного химического соединения. Обычно в результате экспресс-анализа получают данные о наличии элементов группы железа в исследуемых образцах. Но данные о том, в каких соединениях они находятся, теряются. По спектрам ЯМР можно сразу сказать, в каком именно виде (окисел, соль, чистый металл) содержится элемент группы железа.

Результатом БПФ дискретизированного сигнала определенной частоты является количество периодов сигнала. Если частота отсчетов или интервал дискретности по времени при измерении сигнала известен, то по количеству периодов можно установить и частоту измеряемого сигнала. Точность определения частоты в спектре входного сигнала вполне определена и зависит от количества p периодов сигнала [1]. Если количество периодов целое, то частота с помощью БПФ находится абсолютно точно (при отсутствии зашумленности сигнала). Если же количество периодов не является целым, то появляется погрешность определения частоты. Максимальное значение погрешности равняется $1/2p$. В некоторых, практически важных случаях, например, при обработке эхо-сигналов (Рис. 1) импульсных спектрометров ЯМР [2], количество периодов анализируемого сигнала принципиально

ограничено величиной около 10. В этом случае погрешность определения частоты с помощью БПФ достигает 1/20, т. е. 5%.

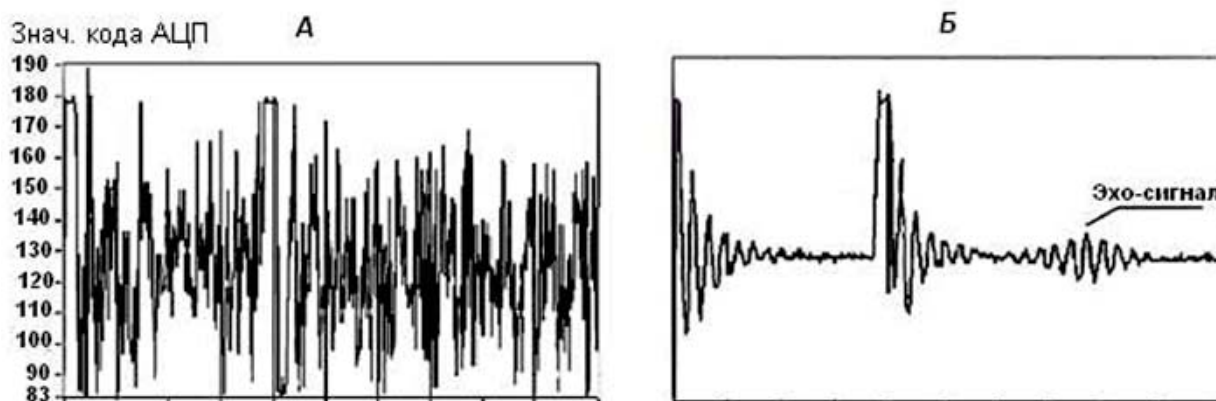


Рис. 1. Исходный (А) и накопленный (Б) эхо-сигнал в борате железа FeBO_3 .

Исследование ряда объектов методами ядерного магнитного резонанса и ядерного квадрупольного резонанса (ЯМР и ЯКР) приводит к необходимости распознавания слабых эхо-сигналов, лежащих, как правило, значительно (на 20 дБ и более) ниже уровня шумов. Общепринятый подход для распознавания наличия и идентификации эхо-сигнала в зарегистрированном сигнале основан на многократном повторении эксперимента и синхронном накоплении эхо-сигналов с последующим частотным анализом сигнала с помощью быстрого преобразования Фурье (БПФ). Предлагаемый новый способ частотного анализа зашумленных эхо-сигналов, описанный в [3,4] программно реализованный в среде MATLAB, обеспечивает значительно более высокую точность определения частоты при обработке одной реализации сигнала. Для широкого практического применения нового способа, а также при решении задачи аппаратной реализации способа с помощью ПЛИС необходима разработка программы на C++. Реализация способа требует выполнения сложных алгоритмов цифровой обработки сигналов, таких как БПФ, взаимная корреляция и сплайн-интерполяция, отсутствующих в стандартных библиотеках Visual Studio.

Алгоритм реализован в виде программы на языке C++ в Visual Studio 2012 в среде операционной системы Windows-8. Для выполнения БПФ была использована библиотека FFTW (<http://fftw.org/>, MIT, Matteo Frigo and Steven G. Johnson), распространяемая по лицензии GPL. Для выполнения сплайн-интерполяции была использована библиотека SPLINE (http://people.sc.fsu.edu/~jburkardt/cpp_src/spline/spline.html, John Burkardt, The Florida State University), распространяемая по лицензии LGPL.

Использование программы в импульсных спектрометрах, подобных описанному в [2], позволит значительно повысить точность анализа веществ. Использование технологий Microsoft при создании программного обеспечения позволит создать компактные недорогие промышленные системы для экспресс-анализа магнитных материалов, таких как образцы руд, содержащих элементы группы железа, полупродуктов и сырья черной и цветной металлургии на базе планшетных компьютеров и USB – устройств сбора данных.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. С.Л.Марпл-мл. Цифровой спектральный анализ и его приложения: Пер. с англ. – М.:Мир, 1990 – 584с. ISBN 5-03-001191-9
2. В.И.Тарханов, В.С.Тутыгин Приборный комплекс для поиска и исследования сигналов ЯМР в магнитоупорядоченных веществах.//Научное приборостроение, 2003, том 13, №1, с.58-63, ISSN 0868-5886.
3. В.С.Тутыгин Цифровая обработка коротких сигналов. СПб., Изд-во Политехн. ун-та, 2012.-164с.
4. В.С.Тутыгин. Способ измерения частоты сигнала. Заявка на изобретение №2011118449 от 17.10.2011. Положительное решение Роспатента о выдаче Патента на изобретение от 17.10.2012

УДК 004.042 + 332.8

А. Д. Солошенко, В. В. Потехин (СПбГПУ)

АВТОМАТИЗАЦИЯ СБОРА И ХРАНЕНИЯ ИНФОРМАЦИИ ОБ ЭНЕРГОПОТРЕБЛЕНИИ ПРИБОРОВ В ДОМАШНЕМ ХОЗЯЙСТВЕ

В основе энергетической стратегии России на период до 2030 г. лежит курс на кардинальную модернизацию электросетевого комплекса [1]. Одним из приоритетов энергетической стратегии является социальная ориентированность развития топливно-энергетического комплекса, то есть повышение жизненного уровня населения [2]. При этом в рыночных условиях ставка делается не на крупномасштабное наращивание производства энергоносителей, а на более эффективное их использование — энергосбережение.

Реальный потенциал экономии теплопотребления в жилых зданиях составляет 40-50%, причем половина этой экономии осуществима за счет снижения потерь тепла непосредственно в квартирах и приводит к улучшению микроклимата в них [3].

Основная цель разрабатываемой системы состоит в сборе данных о потреблении энергоресурсов пользователями системы, анализе этих данных и выдаче пользователям системы рекомендаций по наиболее оптимальному, эффективному и экономически выгодному способу использования энергоресурсов.

Задачей проекта является создание автоматизированной системы сбора и хранения данных об энергопотреблении и передачи их на более высокий уровень для мониторинга и учета.

Сбор данных. Данные должны быть полными и точными, потому что они будут использоваться для анализа и постановки цели. При сборе данных об использовании электроэнергии нужно учесть следующее:

- определение необходимой степени детализации — уровень и объем сбора данных в каждом доме будет отличаться;
- учет всех источников электроэнергии — провести инвентаризацию всей энергии, приобретенной или произведенной на объекте (электричества, газа, пара, топлива из отходов), в физических единицах и на основе стоимости;
- документирование всех фактов использования электроэнергии;
- сбор данных по объектам и эксплуатационных данных — для нормализации и оценки эффективности по контрольным параметрам может потребоваться сбор данных, не связанных с энергопотреблением, по всем объектам и видам работ, таких как размеры зданий, часы работы и т. п.

В данной работе рассматривается способ создания автоматизированной системы. Предполагается, что к каждому объекту энергопотребления присоединены датчики учета мощности или есть возможность идентификации каждого из устройств. Частота сбора информации – 300 раз/сек. Датчики расхода электроэнергии подключаются непосредственно к электрическим розеткам. Таким образом, в базу счетчиков поступают данные непосредственно от электроприборов и далее передаются в агрегированном виде в базу данных экспертной системы с указанием идентификатора каждого прибора. Агрегированная база данных служит основой для работы экспертной системы наряду с базой основных данных, которая содержит в себе профили устройств, базу тарифов и справочники. При таком подходе появляется возможность точно определить энергопотребление любого из приборов. Общая структура модели сбора данных представлена на рис. 1.

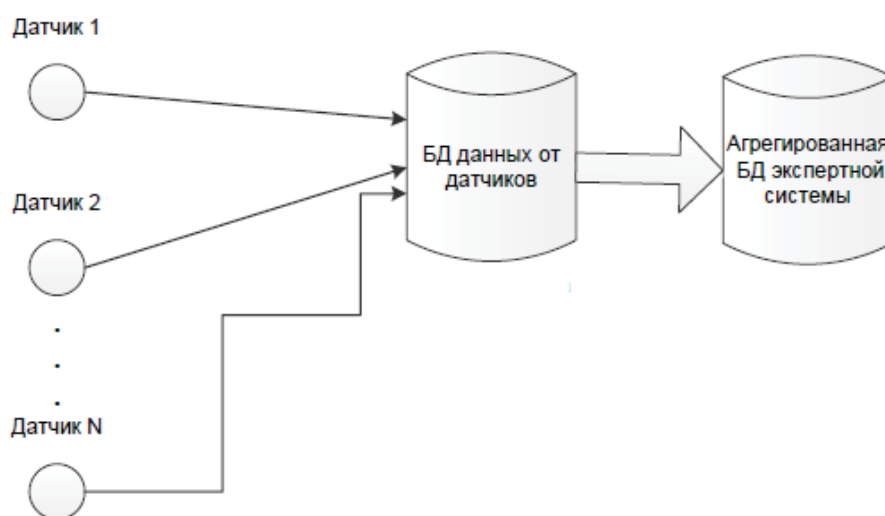


Рис. 1. Модель сбора данных

Для реализации подобной системы, автоматизирующей для реальных задач применения принятие оптимальных решений, предложено использовать стандартные инструментальные средства, основанные на системе firebird. Firebird (FirebirdSQL) — компактная, кроссплатформенная, свободно распространяемая система управления базами данных, работающая на Linux, Microsoft Windows и разнообразных Unix платформах [4].

В данной работе используется структура базы данных, которая содержит в себе периоды (временные интервалы: день-месяц-год), каждому интервалу соответствует таблица в базе данных, по которым будет осуществлен отчет для пользователя. Данная структура позволяет вести автоматический учет энергопотребления за необходимый период времени в компактной форме.

На всех этапах используются компьютерные автоматизированные информационные системы — АИС, которые служат для сбора, накопления, хранения, поиска, передачи, обработки и анализа информации с использованием вычислительной техники, компьютерных информационных сетей, средств и каналов связи [5].

Модель потребления энергии. Потребление электроэнергии представляет собой сочетание следующих факторов: потребление в системе инфраструктуры, использование энергии для общественных потребностей, использования энергии частными потребителями и потери при распределении энергии:

$$E_{sys} = E_{inf} + E_{pub} + E_{con} + L_{dis} ,$$

где:

E_{sys} – общее потребление электрической энергии в системе;

E_{inf} – потребление энергии на инфраструктуру системы;

E_{pub} – потребление энергии для общественных нужд;

E_{con} – использование энергии частными потребителями;

L_{dis} – потеря энергии при распределении [6].

Модель энергетической оптимизации потребления. Можно подойти к проблеме оптимизации энергопотребления двумя способами: сведение к минимуму потребления энергии или за счет минимизации стоимости электрической энергии.

При минимизации потребления энергии, эта проблема может быть формализована следующим образом:

$$E_{sys} = E_{inf} + E_{pub} + E_{con} + L_{dis} \rightarrow \min .$$

При минимизации затрат на энергию, задача формулируется следующим образом:

$$\sum_{i=1}^m \int_0^T E_{(sum)_i}(t) C_i(t) dt + \sum_{j=1}^n \int_0^T L_j(t) C_j(t) dt \rightarrow \min$$

где:

m – количество различных источников электрической энергии в системе;

$E_{(sum)_i}(t)$ – временной профиль энергопотребления объектов, подключенных к определенному источнику энергии;

$C_i(t)$ – время профиля стоимости энергии, который подается для частных потребителей системы;

n – количество различных источников потерь в системе;

$L_j(t)$ – временной профиль потребления энергии на конкретный объект потери энергии в системе;

$C_j(t)$ – временной профиль стоимость энергии, потребляемой объектами потерь энергии [6].

Временной профиль указывает на моменты времени получения информации (например, каждый день).

Выводы. Учет электроэнергии обеспечивает обратную связь и информацию о том, как много энергии было использовано и какова стоимость потраченной энергии. Также такой учет предоставляет понятные данные, которыми могут оперировать обслуживающий персонал, жильцы и управляющие компании для управления затратами и оптимизации энергопотребления. Создав систему учета электроэнергии и понимая принципы ценообразования на электроэнергию, можно эффективнее оптимизировать свои расходы в данной области. Чем больше знаний о потреблении электроэнергии накоплено, тем больше возможность снизить расходы на энергопотребление.

ЛИТЕРАТУРА:

1. Информационно – правовой портал ГАРАНТ. Прогноз долгосрочного социально-экономического развития Российской Федерации на период до 2030 года. Дата обращения: 8 февраля 2013. Электронный документ: <http://www.garant.ru/products/ipo/prime/doc/70209010/>.
2. Официальный сайт Администрации Санкт-Петербурга. Энергетическая стратегия России на период до 2030 года. Правительства Российской Федерации от 13 ноября 2009 г . No 1715. Дата обращения 09

- ноября 2013. Электронный документ: <http://gov.spb.ru/static/writable/ckeditor/uploads/2012/08/05/Energostrategiya-2030.pdf>.
3. Свидерская О. В. Курс лекций. Основы энергосбережения / 4-е издание, стереотипное. - Минск: 2006. - С.37.
4. Firebird Project. About Firebird. DQ Team. Дата обращения: 14 марта 2013. Электронный документ: <http://www.firebirdsql.org/>
5. Фуфаев Д. Э., Фуфаев Э. В. Разработка и эксплуатация автоматизированных информационных систем. – М.: Академия, 2008. - С. 8.
6. Potekhin V. V., Kovalevsky V. E., Ambrajei A. N. and Leming M. J. Intelligent information measuring system for monitoring and optimization of energy consumption, Distributed Intelligent Systems and Technologies Workshop, St. Petersburg, Russia, 1-4 July 2013. - P. 81–88.

УДК 004.67

С. Н. Здробилко , П. Д. Дробинцев (СПбГПУ)

СИСТЕМА ВЫЯВЛЕНИЯ АНОМАЛИЙ И ШУМОВ В НЕДЕТЕРМИНИРОВАННЫХ СИГНАЛАХ ПОДВОДНЫХ МАГИСТРАЛЕЙ ПРОТЯЖЕННОСТЬЮ 200 КМ И БОЛЕЕ

Проблемная область. Подводные магистрали можно считать подклассом гидроакустических систем, относящихся к классу сложных технических систем сбора и обработки информации, функционирующих в условиях неопределенности при изменяющихся параметрах системы и характеристиках среды эксплуатации.

Сложность получения достоверных данных от подводных магистралей связана со следующими их особенностями: большая протяженность, большое количество элементов, случайное пространственное расположение элементов на морском дне, наличие до нескольких десятков разветвленных антенных подсистем, разнесенных на несколько десятков километров друг от друга. Немаловажным фактором является изменчивость подводной морской среды эксплуатации, внешней по отношению к магистральной системе, он оказывает случайное возмущающее воздействие на работу отдельных элементов и каналы передачи информации.

Обзор существующих типов аномалий и шумов. Среди гидродинамических шумов различают турбулентные и вихревые шумы. Сплошная среда может плавно обтекать тела, а может отрываться от обтекаемых поверхностей, сворачиваясь в вихри. Картина течения прежде всего зависит от формы обтекаемого тела, а так же от скорости и характера его движения. Если течение плавно обтекает тело, то такие тела называют хорошо обтекаемыми [1]. Если течение отрывается от поверхности тел, то такие тела называют плохо обтекаемыми. Плавное обтекание при достаточно больших скоростях течения сопровождается турбулентным шумом. Турбулентный шум на границе турбулентных течений называют пристеночными пульсациями давления [2]. Основная энергия пристеночных пульсаций давления сосредоточена в области больших волновых чисел, которую часто называют псевдозвуком. При отрыве течения возникает вихревой шум. Турбулентный шум имеет чрезвычайно широкий пространственно-временной спектр. Интенсивность вихревого шума главным образом сосредоточена в сравнительно узком частотном диапазоне и часто превышает интенсивность турбулентного шума.

Кроме того следует обратить внимание на механические шумы. Вращающиеся лопасти обеспечивают силу тяги для кораблей приводят к возникновению значительного

акустического шума, а так же вызывают движение сплошной среды в целях вентиляции и теплообмена [3,4]. Помимо движения лопастей подводный шум судов складывается из главной энергической установки и вспомогательных механизмов, гидродинамического шума. Шумы судов проникают в водную среду разными путями: через корпус, через воздушную среду отсеков, через крепления, фундаменты, гребные валы, трубопроводы и кабели, через потоки воды в системах охлаждения и др. Для боевых кораблей, особенно для подводных лодок, подводный шум является демаскирующим фактором, мешающим скрытности их действий [5].

Помимо техногенных шумов следует так же рассматривать и природные шумы, такие как поверхностные шумы моря, обусловленные волнением моря в результате взаимодействия атмосферы с поверхностью моря, шумы связанные с тепловыми и кавитационными процессами в толще морской воды, динамикой ледового покрова, биологическими явлениями и тектонической деятельностью земной коры на дне морей [6].

В работе перед автором *ставилась задача* разработки системы автоматического мониторинга гидроакустической среды, *целью* которой является автоматическое выявление аномалий и шумов во входящих сигналах, а так же определение вышедших из строя узлов системы.

Для *решения поставленной задачи в работе* предлагается использовать комбинацию из методов выявления аномалий и шумов

Для определения работоспособности всех узлов системы используется метод запроса авто-повторений.

Метод запроса авто-повторений. Запрос авто-повторений (ARQ) используется, чтобы обнаружить ошибочные данные в процессе передачи по каналу связи и в случае их обнаружения осуществить повторную передачу пакетов, содержащих ошибки. Самая простая схема запроса авто-повторений, которая может быть использована в подводной магистрали это stop and wait ARQ, где источник пакета ждет квитанции от узла адресата для подтверждения передачи пакета без ошибок. Так как канал не используется в течение времени между передачей пакета и приемом квитанции, эта ARQ схема имеет низкую производительность. В go back-N и селективном методах повторении ARQ, узлы передают пакеты и получают квитанции непосредственно в процессе передачи пакетов, поэтому требуют канал, поддерживающий полный дуплекс. Деление ограниченной пропускной способности подводной магистрали на два канала реализации полного дуплекса может значительно уменьшать скорость передачи данных по физическому каналу. [7]

Схема селективного метода может быть модернизирована, чтобы работать в симплексных подводных магистралях. Вместо подтверждения каждого пакета в процессе приема, получатель будет ждать пока не будут переданы N пакетов и после этого посылать квитанцию с номерами пакетов, полученных без ошибок. Соответственно, источник пакетов определит ошибочные пакеты по принятой квитанции и pošлет другую группу N пакетов, которая будет содержать повторно передающиеся и новые пакеты.

Подтверждения могут быть обработаны двумя возможными способами. В первом подходе, который называется “положительное подтверждение”, в случае приема пакета, не содержащего ошибок, узел адресата pošлет квитанцию исходному узлу. Если источник не получает квитанцию исходному узлу. Если источник не получает квитанцию за заданный промежуток времени, то он повторно передаст этот пакет данных. В случае отрицательного подтверждения, адресат посылает квитанцию, если получает поврежденный пакет или не получает его вообще. Отрицательное подтверждение может помогать сохранять энергию,

устраняя потребность посылать квитанции каждый раз и повторно высылать пакеты данных в случае потери квитанции [8, 9].

Для анализа входящих сигналов используется комбинация из описанного выше метода запроса авто-повторений и метода фильтрации Калмана.

Метод фильтрации Калмана. Динамический анализ стохастических сигналов имеет важное значение при решении широкого круга научно-технических задач. Важным видом задач обработки является идентификация или определение параметров исследуемых процессов на основе анализа поступающей акустической информации. Для анализа сигналов, как правило, используются методы спектрального и стохастического анализа [10, 11]. Особенностью спектрального подхода является использование всех отсчетов данных, зарегистрированных в анализируемой выборке. Многие типичные исследуемые процессы имеют специфический спектральный состав и занимают характерные спектральные области, что позволяет легко выделять информативные составляющие сигналов. К недостаткам классического спектрального анализа относятся малая чувствительность к локальным свойствам сигналов, недостаточно высокое спектральное разрешение и сравнительно большие вычислительные затраты. Как правило, спектральный анализ проводится по выборке данных достаточно большой протяженности, и короткие локальные изменения не вносят значительного вклада в результирующий спектр сигнала. Вместе с тем, часто именно локальные изменения свойств сигналов, например, при ударных процессах, могут содержать полезную информацию. Большой объем обрабатываемых данных и сложные алгоритмы обработки требуют значительных вычислительных затрат при анализе спектров даже в случае использования быстрых алгоритмов типа быстрого преобразования Фурье (БПФ).

Методы стохастического анализа [12, 13] позволяют анализировать как глобальные, так и локальные свойства сигналов в динамическом режиме и требуют меньших вычислительных затрат. Такие методы основываются на параметрических моделях, в структуре которых учитывается априорная информация об общих свойствах анализируемых сигналов. При использовании методов стохастической фильтрации ведется обработка только той информации, которая необходима в рамках решаемой задачи. Например, если известно, что следует анализировать данные только в определенной полосе частот, можно обойтись без спектрального преобразования и последующего анализа во всей спектральной области, используя фильтр Калмана для выделения в исходных данных необходимых спектральных областей.

Фильтры Калмана низких порядков позволяют осуществлять динамическую обработку стохастических сигналов при незначительных вычислительных затратах, что особенно важно при анализе акустических сигналов в реальном времени. Фильтр Калмана первого порядка удобен для выделения низких частот в заданном диапазоне спектра мощности, фильтр второго порядка обеспечивает выделение узкополосных стохастических сигналов в заданной полосе частот.

В работе *реализован прототип системы* мониторинга аномалий в реальном времени. Получая на вход данные с датчиков расположенных на подводной магистрали, система используя предложенные методы выявляет аномальные пики в потоке данных. Гибкая настройка с помощью конфигурационного xml файла позволяет задавать необходимые пороговые значения аномалий и возможные, допустимые, погрешности в обработке сигналов. Кроме того с помощью метода запроса авто-повторений система позволяет наблюдать за работоспособностью устройств, запрашивая их состояния и по результату ARQ ответа делающая выводы о работоспособности того или иного устройства.

В системе мониторинга предусмотрена визуализация в виде графиков, показывающих качество сигнала в зависимости от времени, а так же в виде схемы подводной магистрали с датчиками, с указанием на каких из датчиков появляются аномалии и шумы.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. С. И. Девнин. Аэродинамический расчет плохообтекаемых судовых конструкций. - Л: Судостроение, 1967.
2. А. В. Смольяков, В. М. Ткаченко. Измерение турбулентных пульсаций. - Л: Энергия, 1980
3. Д. И. Блохинцев. Акустика неоднородной движущейся среды. - М: Наука, 1981
4. И. Я. Миниович, А. Л. Перник, В. С. Петровский. Гидродинамические источники звука. - Л: Судостроение, 1972
5. Справочник по гидроакустике/ Под ред. А. Е. Колесникова. - Л.: Судостроение 1982
6. А. В. Фурдуев. Шумы моря. В кн. под ред. Л. М. Бреховских. Акустика океана. -М: Наука, 1974
7. J. G. Proakis, J. A. Rice, E. M. Sozer, M. Stojanovic. Shallow Water Acoustic Networks // IEEE Communications Magazine. – 2001. November
8. В. Г. Олифер, Н. А. Олифер / Компьютерные сети. Принципы, технологии, протоколы. – Санкт-Петербург: Питер, 2001. – 672с.
9. С. Н. Новиков. Методы маршрутизации на цифровых широкополосных сетях связи: Ч.1 / Учебное пособие. – Новосибирск: 2000. – 84с.
10. С. Л. Марпл. Цифровой спектральный анализ и его приложения. М.: Мир, 1990.
11. Дж. Бендат, А. Пирсол. Применения корреляционного и спектрального анализа. М.: Мир, 1983.
12. В. Н. Васильев, И. П. Гуров. Компьютерная обработка сигналов в приложении к интерферометрическим системам. СПб: БХВ - Санкт-Петербург, 1998
13. Справочник по прикладной статистике / Под ред. Э. Ллойда, У. Ледермана. М.: Мир, 1990. Т. 2.

УДК 621.311.24

А. В. Васильева, Д. А. Тыжненко, В. В. Потехин

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО АЛГОРИТМА И СОЗДАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ АВТОНОМНОЙ ВЕТРОЭЛЕКТРОСТАНЦИЕЙ

В настоящее время во всём мире в качестве основного источника энергии используется органическое топливо: газ, нефть, уголь. Однако, эти ресурсы исчерпаемы, что, в дальнейшем, может привести к необратимым последствиям. Альтернативой данным источникам энергии являются природные неисчерпаемые ресурсы, такие как, например: энергия ветра, солнца, приливов и отливов рек и озёр. Особое внимание уделяется такому источнику, как ветер, так как он обладает наиболее высоким потенциалом. Одна из главных и основных проблем ветроэнергетики – скорость и непостоянство направления ветра. Для стабилизации мощности автономных ВЭС в их состав включаются электроаккумулирующие устройства, которые накапливают избыточную энергию и отдают её в моменты пиковых нагрузок. Максимальная вырабатываемая мощность ВЭС не обязательно будет получена при максимуме выработки каждой ВЭУ в отдельности. При расчете мест стоящих рядом ветряков необходимо учитывать образование больших безветренных пространств, таким образом, при объединении ветроустановок в единую сеть возникает вопрос выбора топологии объединения.

Требования к ветроустановкам и качеству вырабатываемой электроэнергии регламентируются нормативным документом [1]. Мощность, вырабатываемая следующими друг за другом ВЭУ, зависит от коэффициента торможения потока ветра первой установки. Если максимальной мощности первой соответствует оптимальное значение торможения потока, то, выработка второй падает. [2] Этот факт указывает на то, что необходимо оптимизировать работу ВЭС в целом. Главной задачей интеллектуальной системы управления автономной ВЭС, является в первую очередь обеспечение надежного электроснабжения с высоким качеством электроэнергии, с соблюдением условия в каждый момент времени. Основными производителями ветроустановок являются: Vestas[3], Gamesa [4].

В качестве способа управления системой ВЭС предложено использовать мультиагентный подход. Разработана мультиагентная система для организации взаимодействия составных частей ВЭС, сбора данных, передачи и обработки управляющих воздействий [6].

В процессе разработки системы управления автономной ветроэлектростанции рассматривались два направления:

1. разработка алгоритма комплексного управления ветроэлектростанцией на верхнем уровне;
2. разработка алгоритма управления отдельной ВЭУ в составе ВЭС.

За основу в схеме работы агента взята структура сети адаптивной критики, представленная на рисунке 1. Структура строится на двух основных компонентах:

- Адаптивный критик;
- Исполнитель.

Цель критика в данном случае на основе состояния системы (событий сети обработки событий) предоставить грубую внутреннюю оценку/награду, полученную на основе состояния среды (сети обработки событий). В целом это представляет приближение функции оценки при текущей стратегии, транслируемой исполнителю. Исполнитель стремится выработать t -оптимальную стратегию (оптимальную стратегию в момент времени t) на основе значения, полученного от критика. В то же время критик стремится получить функцию оценки при текущей стратегии, выработанной в исполнителе.

Проблема управления режимами работы автономной ветроэлектростанции может быть представлена в виде Марковского процесса принятия решений [5]. Марковский процесс является стохастической системой со множеством состояний, изменения состояний описываются вероятностями состояний. При разработке алгоритма предусматривалась его расширяемость за счет знаний полученных от экспертов и эксплуатирующего персонала.

В связи с этим на этапе испытаний и промышленной эксплуатации существует возможность изменения набора состояний и действий, которые могут выполняться агентами системы.

Выводы. Основной целью реализованного проекта была разработка прототипа системы управления автономной ВЭС, реализующей интеллектуальный алгоритм управления ВЭС с целью повышения эффективности работы автономной ВЭС путем обеспечения максимального использования энергии ветра. Разработанная система может быть применена к широкому спектру автономных ВЭС обеспечивающих энергоснабжение населенных пунктов, исследовательских станций и иных потребителей с характером нагрузки преимущественно активным, не имеющих возможности подключения к общей системе энергоснабжения.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.



Рис. 1. Схема взаимодействия агента с сетью обработки событий

ЛИТЕРАТУРА:

1. ГОСТ Р. 51991-2002 «Нетрадиционная энергетика. Ветроэнергетика. Установки ветроэнергетические. Общие технические требования, МЭК 61400-1:2010 «Ветроэнергетические турбины. Часть 1. Требования к конструкции».
2. K. Pandey, A. Tiwari, - Maximum power point tracking of wind energy conversion system with permanent magnet synchronous generator, International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 5, July – 2012.
3. Официальный сайт компании «Vestas» (Дания). – Режим доступа: <http://www.vestas.com/en/wind-power-plants.aspx#/vestas-univers>
4. Официальный сайт компании «Gamesa» (Испания). – Режим доступа: <http://www.gamesacorp.com>
5. М. Я. Кельберг, Ю. М. Сухов - Марковские цепи как отправная точка теории случайных процессов и их приложения, МЦНМО, 2010 г.
6. A. L. Loginov, V. V. Potekhin, D. V. Akulov, A. K. Khritonenkov, I. G. Poletaev . - INTELLIGENT CONTROL SYSTEM OF AUTONOMOUS WIND FARM, SPBSPU, - 2013.

УДК 004.9

Л. И. Кожаяев, С. А. Фёдоров (СПбГПУ)

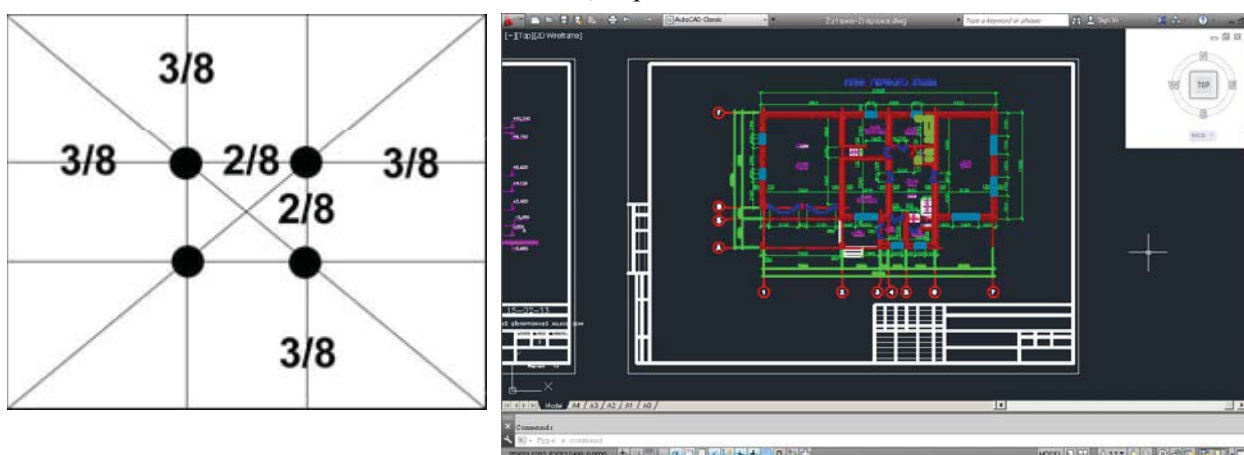
РАЗРАБОТКА МОДУЛЯ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ЗДАНИЙ И СООРУЖЕНИЙ В СООТВЕТСТВИИ С НОРМАТИВНОЙ ДОКУМЕНТАЦИЕЙ

В работе был разработан модуль для автоматизированного проектирования зданий и сооружений в соответствии с нормативной документацией, отвечающий следующим требованиям:

- автоматическое выполнение некоторых рутинных задач при проектировании ИЖС;
- автоматическая настройка рабочей среды инженера в полном соответствии с требованиями стандарта предприятия;
- реализовано взаимодействие с системами управления предприятием.
- автоматизированная генерация планировки помещения индивидуального жилого дома в рамках нормативной документации с возможностью дальнейшей модернизации и коррекции полученной модели;

На основе золотого сечения можно определить зрительные центры и использовать для компоновки элементов на листе и корректном выравнивании форматной рамки относительно чертежа. Локус внимания в данном случае будет центрирован, относительно листа следующим образом: см. рис. 1.

Рис.1. Центр внимания



Золотое сечение также может использоваться при генерации помещений здания, которая осуществляется в несколько этапов:

1. Определяем площадь минимального помещения в здании на основании нормативов.
2. Определяем минимальные длины стен для данного помещения исходя из нормативов и здравого смысла. Очевидно, что минимальную допустимую площадь в 6м² образует помещение со сторонами стен примерно 2,5м.

Площади остальных помещений (спальни и холла) можно вычислить по формулам:

$$S_2 = S_{\text{мин. помещ.}} * 1.6181;$$

$$S_3 = S_2 * 1.6181;$$

Далее можно модифицировать алгоритм в отношении компоновки комнат на чертеже (см. рис. 1, 2 и 3).

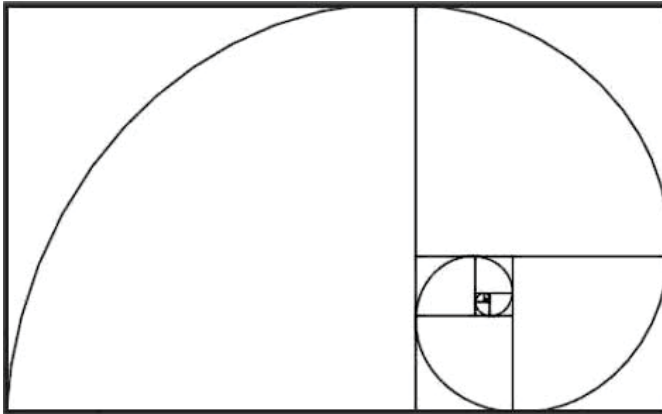


Рис. 2. Схематическая реализация алгоритма

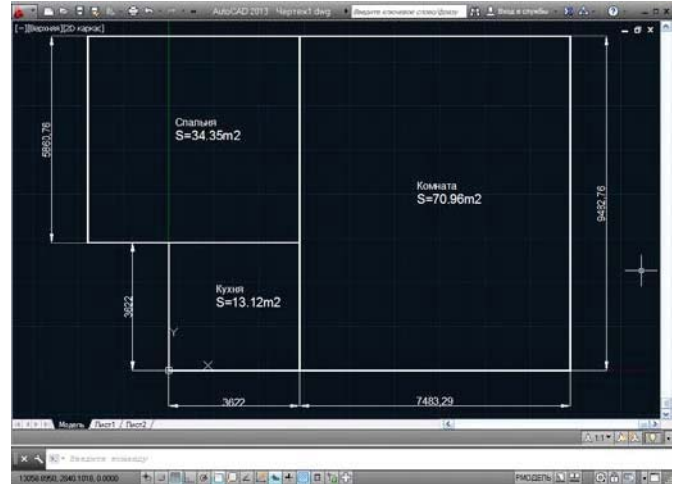


Рис. 3. Генерация планировки, вычисление размеров (мм) и площадей в AutoCAD.

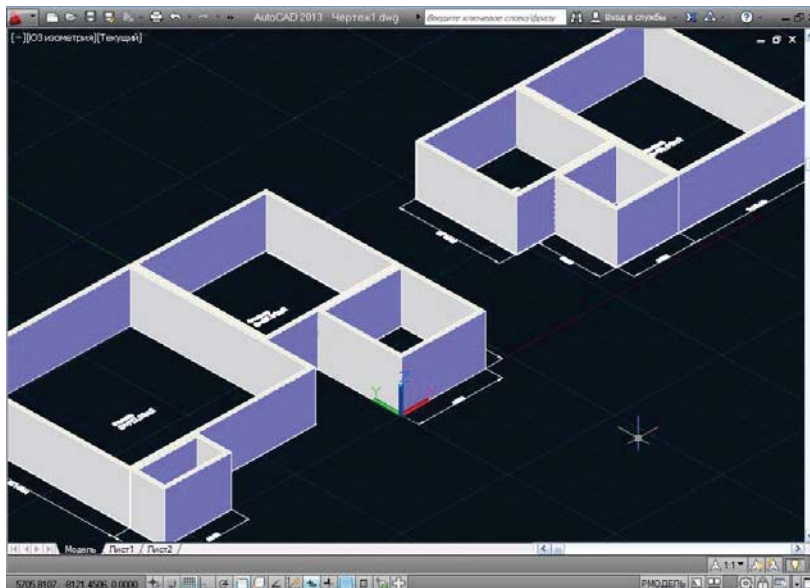


Рис. 4. 3D модель на основе планировки внутренних помещений. Вся графика, размеры, площади помещений

Были получены результаты по направлению автоматизации проектирования зданий и сооружений.

При разработке модуля были решены следующие задачи:

1. Обзор инструментов по производству моделей зданий.
2. Сравнительный анализ BIM- и CAD- технологий проектирования.
3. Обзор нормативной документации.
4. Разработка методов проектирования внутренней планировки здания.
5. Разработка алгоритма формирования рамок в соответствие со стандартом.
6. Разработка алгоритма формирования штампа по стандарту.
7. Разработка алгоритма формирования слоёв в соответствие со стандартом предприятия.
8. Реализация разработанных методов и алгоритмов.

Данная работа была посвящена изучению современных технологий проектирования и программирования и применению данных технологий при разработке программного модуля, для автоматизации процесса проектирования инженерами промышленного и гражданского строительства. Также возможно применение данной разработки и в других отраслях. Разработанные алгоритмы могут быть применены практически к любой САПР системе.

Основным перспективным направлением данной работы является получение расширяемого многофункционального модуля с интуитивным интерфейсом для повышения автоматизации в проектных организациях. Интеграция с наиболее популярными САПР платформами: AutoCAD, NanoCAD, Revit.

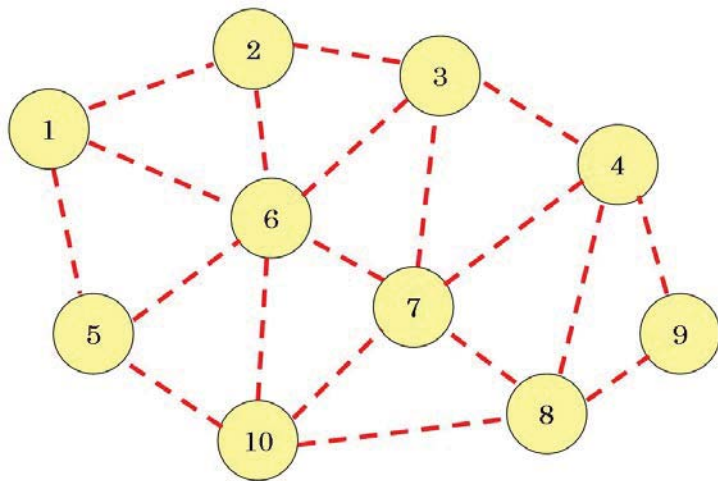
Выводы. В работе реализована попытка полностью автоматизировать процесс создания внутренней планировки индивидуального жилого дома в рамках граничных значений из нормативной документации. Для более сложной генерации проектной документации требуется создание расширенной экспертной системы на основе базы знаний о проектировании. На основе этих знаний теоретически возможно сгенерировать проектную документацию на тот или иной тип здания со всеми расчётами, архитектурой и инженерией.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 004.722.2

Н. С. Голубев, И. А. Селин, Н. В. Воинов, П. Д. Дробинцев (СПбГПУ)

ОРГАНИЗАЦИЯ СЕТИ С ЯЧЕЙСТОЙ ТОПОЛОГИЕЙ



Краткий обзор: Ячеистая топология — сетевая топология компьютерной сети, в которой каждая рабочая станция сети соединяется с несколькими другими рабочими станциями этой же сети. Данный тип топологии характеризуется высокой отказоустойчивостью, необходимой при решении достаточно широкого круга задач, однако при этом резко возрастает сложность настройки и расход кабеля. В рамках сети каждый компьютер имеет множество возможных путей соединения с другими компьютерами, часть из которых должна

быть реализована, при этом предполагается что обрыв кабеля в одном из соединений не приведет к потере работоспособности сети. Такая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных промышленных сетей. Пример сети с ячеистой топологией представлен на рис. 1.

Для беспроводных сетей преимуществом использования этой топологии является мобильность. Нет необходимости иметь заранее организованную инфраструктуру для работы сети [1].

Целью данного исследования было выбрать подходящий программно-аппаратный комплекс и создать сеть с ячеистой (mesh) топологией, провести начальную конфигурацию, проверку работоспособности и измерение различных характеристик сети в процессе ее работы.

Построение сети на основе беспроводных устройств: Для реализации был выбран программно-аппаратный комплекс, состоящий из компьютеров на базе ОС Ubuntu 12.04 [2] и устройств, использующих драйвера с поддержкой mesh на основе стандарта 802.11s [3]. При использовании старой версии Ubuntu или иного дистрибутива Linux нужно загрузить драйвера устройств (они организованы как backports, т. е. портированы с новой версии ядра). Списки устройств, драйвера и информация о их возможностях представлены на сайте проекта Linux Wireless [4].

Все необходимые операции производились в командой строке (терминал) от суперпользователя.

Для конфигурирования узла необходимо провести следующие действия:

1. Перевести устройство в режим unmanaged. Для этого необходимо либо выключить network-manager, либо добавить mac устройства [5] в список исключений network-manager
2. Определить индекс настраиваемого устройства, для этого надо ввести команду iw dev
3. Установить интерфейс mesh-сети с помощью команды iw dev wlan# interface add \$MESH_IFACE type mp mesh join \$MESH_ID, где \$MESH_IFACE — имя интерфейса, а \$MESH_ID — название mesh сети.
4. Установить IP-адрес устройства для его дальнейшего использования: ifconfig \$MESH_IFACE \$NODE_IP, где \$NODE_IP — IP-адрес узла
5. Проверить список подключенных устройств с помощью команды iw dev \$MESH_IFACE station dump

И посмотреть таблицу маршрутизации у узла:

```
iw dev $MESH_IFACE mpath dump
```

После выполнения приведенных выше настроек, сеть начинает использовать адресацию на уровне IP. При этом маршруты до других устройств будут строиться автоматически. В случае отказа одного из узлов, маршруты будут перестроены.

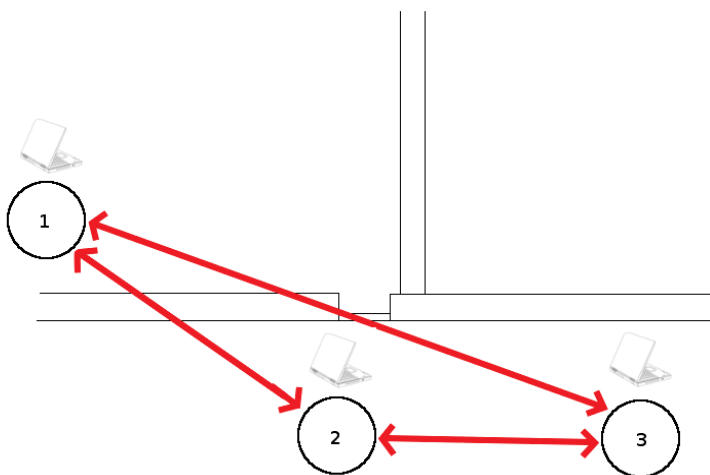


Рис. 2 Полученная топология

В результате настройки была получена сеть с топологией, приведенной на рис. 2.

Результаты: В результате проделанной работы была получена mesh-сеть и были проведены замеры скоростей и задержек. Обозначения узлов будут соответствовать рис. 2. Данные приведены в табл. 1.

Узлы передачи	Скорость (Mb/s)	Задержка (мс)
1 → 2	3.2	2.4
2 → 3	9.3	1.7
1 → 2 → 3	0.5	6.3

По результатам проведенных замеров можно отметить, что один промежуточный узел вносит значительную задержку и также снижает скорость. Так же при передаче через один узел происходили потери информации (до 46% пакетов).

Вывод. Полученная сеть позволяет передавать данные небольших объемов с относительно небольшими задержками. При отключении одного узла происходит перестроение карты сети, в соответствии с оставшимися, что позволяет не прерывать передачу данных.

На основании проведенного исследования можно сделать вывод о возможности использования mesh-сети для передачи голосовых сообщений с некоторыми ограничениями, обусловленными низкой скоростью передачи данных.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Steve Methley. Essentials of Wireless Mesh Networking: Cambridge University Press, 2009, 228с.
2. Ubuntu desktop: URL: <http://www.ubuntu.com/download/desktop>
3. IEEE Xplore - IEEE 802.11s: The WLAN Mesh Standard:
URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5416357>
4. Users — Linux Wireless: URL: <http://wireless.kernel.org/en/>
5. Definition of mac address: URL: http://en.wikipedia.org/wiki/MAC_address

УДК 004.4

А. В. Федоров, В. В. Потехин (СПбГПУ)

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО АЛГОРИТМА ОБРАБОТКИ ИЗОБРАЖЕНИЙ УЗИ В БИОМЕДИЦИНЕ

Ультразвуковая эластография - это метод, который измеряет деформацию тканей в ответ на приложенное усилие, тем самым показывая их жесткость. Анализ жесткости тканей дает новое измерение в анализе ультразвуковых изображений, которые включают в себя такие свойства, как качественную и количественную оценку эластичности тканей. Многие программные комплексы в области диагностических инструментов могут обеспечить более высокую точность диагностики, чем просто отображение исходного изображения с аппарата.[1] Далее описан интеллектуальный алгоритм получения эластограммы при помощи обычного ультразвукового устройства. Было разработано приложение, которое принимает

видео-поток и помогает врачам решать диагностические задачи. Для выполнения алгоритма в режиме реального времени в приложении был использован многоядерный подход.

За счет ультразвуковой эластографии врачами не нужно использовать метод пальпации для обнаружения твердых тканей, программа или устройство с функцией эластографии показывает, какая ткань более жесткая. Основываясь на данных о жесткости врачи-диагносты могут судить является ли данный участок ткани ненормальным, например, злокачественные новообразования жестче доброкачественных. Большинство существующих устройств с функцией эластографии, использует свойство ультразвуковой волны - поперечные волны распространяются быстрее в жесткой ткани, чем в мягких тканях, следовательно жесткость ткани можно вывести путем измерения поперечных волн, зная их длину и интенсивность.[2] Однако стоимость таких устройств очень высока. Главная цель данной работы - получить ультразвуковые эластографию без каких-либо модификаций ультразвуковых устройств, в которых нет функции эластографии. Все вычисления в данном случае выполняются на компьютере, подключенном к ультразвуковому устройству.

Подход без использования квазистатического метода, волн сдвига или акустического излучения силового импульса (ARFI) является менее точным и требует дополнительных действий со стороны врачей.[3] Система, разработанная автором, использует только видео - поток, ничего не зная о скорости волны или других ее свойствах. С учетом данных ограничений, система работает только в динамике - врач должен оказывать давление при помощи ультразвукового устройства на органы, для регистрации изменений.

Общая схема алгоритма в программе следующая: вначале видео - поток через VGA - порт передаются в компьютер от ультразвукового устройства, затем программа принимает изображения из этого потока и отображает их в главном окне. Врач выбирает область интереса(ROI) , и программа накапливает изображения в специальную структуру данных. Изображения поступающие с ультразвукового устройства являются черно-белыми. Для отображения эластограммы программа принимает кадры в режиме реального времени и специальный алгоритм вычисляет относительную жесткость тканей в области интереса. Алгоритм похож по схеме действия на нейронные сети. Сначала по сохраненным изображениям вычисляются веса для каждого пикселя или зоны пикселей изображения в области интереса. Затем текущее изображение поступающее с ультразвукового устройства сравнивается с полученными весовыми функциями, а пороговая функция дает нам цвет каждого нового пикселя или зоны пикселей. Пороговая функция на выходе выдает четыре цвета - синий (для мягких тканей), зеленый (для тканей менее мягких), желтый (для жестких тканей) и красный (самые жесткие ткани). Другой алгоритм позволяет обновлять структуру хранения данных, для отображения относительных смещений ткани после приложенного к ним давления. Результат работы программы на примере лимфомы человека показана на рис.1.:

Многоядерный подход был реализован для распараллеливания вычисления весов из структуры хранения изображений. Вычисления в зоне интереса могут быть распараллелены по строкам и по столбцам.[4] Такой подход дает обновление эластограммы исходного изображения практически в реальном времени. На ПК с процессором Core 2DUO частота обновления кадров программы составляла всего 15 кадров в секунду, с процессором Core i7 с восемью ядрами, частота обновления кадров повысилась до 45, что позволяет врачу смотреть в режиме реального времени видео - поток от ультразвуковых устройств и осуществлять анализ эластограммы.

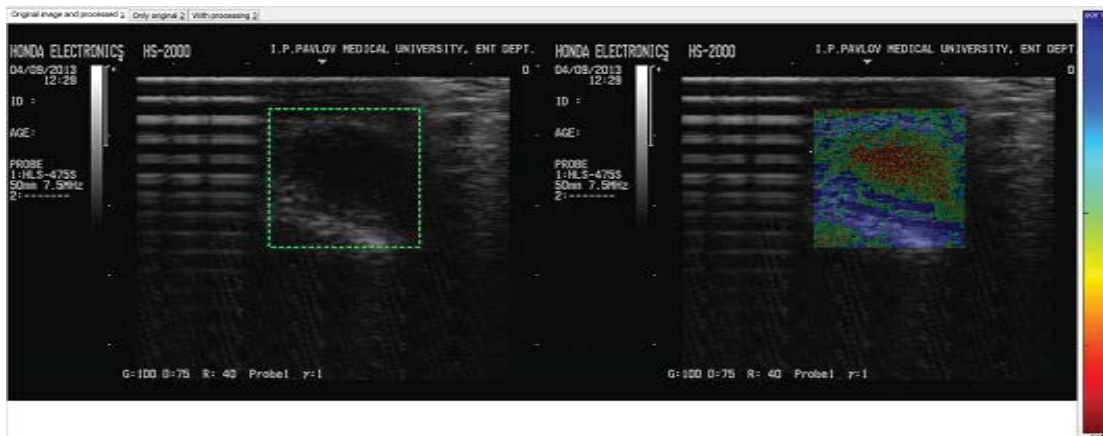


Рис. 1. Ультразвуковое изображение эластографии жесткости ткани.

В будущем планируется добавить больше функциональных возможностей программы путем реализации кластеризации жесткости тканей и добавив некоторые аспекты обучения. [5] Обучение программы будет включать в себя изучение тестовых примеров, где должна быть определены злокачественные и доброкачественные новообразования.[6]

Выводы. В лаборатории были использованы ультразвуковое устройство General Electric 200 PRO и ПК с процессором Core-i7 с 8 ядрами. Программа была успешно протестирована в лаборатории, и в данный момент проводятся клинические испытания квалифицированным медицинским персоналом для получения статистических показателей. Важно учитывать что данный метод имеет ограничения, например, некоторых тканях внутренних тканей с тем же цветом внешней ткани и потому, что программа работает только с изображениями, он не может найти этот скрытый слой. Но с большим количеством тестов в лаборатории метод показал хорошие результаты.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. John Benson, Liexiang Fan, "A Complete Ultrasound Solution for Elastography", Tissue Strain Analytics, Siemens Medical Solutions USA, Inc., 2012.
2. Si Luo, Cheoljin Lee, Kerem Karadayi and Yongmin Kim, Real-Time Ultrasound Elastography on a Multi-core DSP, Texas Instruments Incorporated, 2011.
3. A. Saftoiu, P. Vilmann, F. Gorunescu, D.-I. Gheonea, M. Gorunescu, S. Belciug, Endoscopic Ultrasound Elastography in the Diagnosis of Pancreatic Cancer, Annals of gastroenterology 2010, 23(3):200-201 p.
4. Jonathan F. Carlsen, Caroline Ewertsen, Lars Lönn and Michael B. Nielsen, Strain Elastography Ultrasound: An Overview with Emphasis on Breast Cancer Diagnosis, Diagnostics, 2013.
5. T. Varghese, J. A. Zagzebski, G. Frank and E. L. Madsen, Elastographic imaging using a handheld compressor, Ultrasonic Imaging 24,2002, 25- 35 p.
6. Brian S. Garra, Tissue elasticity imaging using ultrasound, APPLIED RADIOLOGY, April 2011.

УДК 004.4'2

Н.Д. Борисов (4 курс, каф. ИУС), И.В. Никифоров , (асп., ИУС)

ГЕНЕРАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ПРЕДСТАВЛЕНИЯ НА ОСНОВЕ XML-СПЕЦИФИКАЦИЙ ДЛЯ ТЕХНОЛОГИИ SAP DYNPAGE

На сегодняшний день актуальной является задача разработки автоматизированных систем управления внутренними бизнес процессами предприятий, к которым относятся: бухгалтерский учет, торговля, производство, финансы, управление персоналом, управление складами и т. д. Одной из таких систем, существующих на рынке программного обеспечения, является система SAP NetWeaver [1,2].

В SAP NetWeaver входит компонент, который представляет собой корпоративный веб-портал SAP NetWeaver Portal. Он является единой точкой доступа к корпоративным сервисам, приложениям и данным. И помимо этого является платформой для выполнения приложений реализованных с помощью технологий Web Dynpro[3] или Dyn Page[4].

Особенностью DynPage-приложений является использование множества компонентов HTML-b[5] для языка Java. Сочетание двух технологий позволяет создавать гибкие и функциональные порталные приложения, способные решать бизнес задачи промышленного уровня.

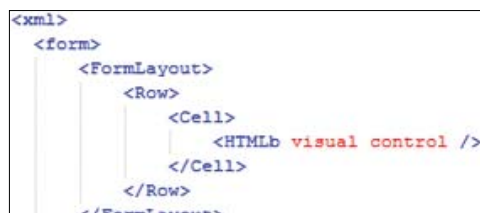
На сегодняшний день является актуальной задача автоматизации процесса проектирования пользовательского интерфейса (ПИ) для связки технологий DynPage и HTML-b. В современных условиях процесс создания пользовательского интерфейса трудоемок и требует значительных временных затрат. Такая тенденция связана с:

1. отсутствием средств автоматизации проектирования ПИ;
2. с невозможностью отделения описания ПИ от бизнес-логики приложения;
3. с многократным использованием повторяющегося кода.

В работе предлагается реализовать прототип программного продукта, который позволит перейти к декларативному подходу создания ПИ для технологии DynPage, и автоматизировать часть работы с целью уменьшения временных издержек. Предлагаемый подход основан на генерации java-кода, описывающего ПИ, на основе xml-спецификации (рис. 1).

Свойства объектов ПИ задаются через атрибуты xml-спецификаций. Важной особенностью такой реализации является тег «HTMLb visual control», под которым понимается визуальный компонент с атрибутами или же дополнительный *FormLayout*.

Генерации кода основана на использование программного интерфейса DOM (Document Object Model)



```
<xml>
<form>
  <FormLayout>
    <Row>
      <Cell>
        <HTMLb visual control />
      </Cell>
    </Row>
  </FormLayout>
</form>
```

Рис. 1 структура xml-спецификации

Предлагаемая в работе xml-спецификация и подход к автоматизации генерации кода позволяет задать и конфигурировать ПИ в более простом и понятном виде. Позволяет избежать ручной работы по созданию ПИ, тем самым сокращает трудоемкость разработки порталных приложений.

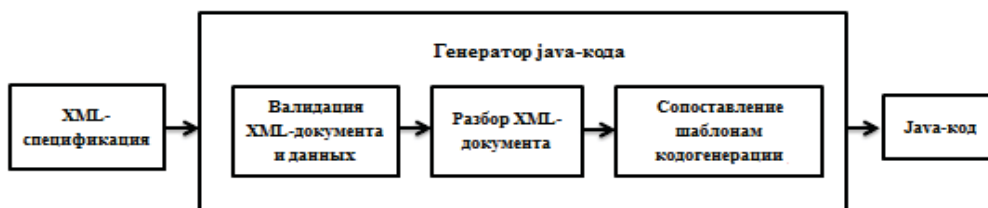


Рис. 2 Схема генерации java-кода

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. SAP NetWeaver. [Электронный ресурс]. Режим доступа: <http://scn.sap.com/community/netweaver>
2. Marcus Banner, Berthold Latka, Michael Spee, Roland Schroth. Developer's Guide to SAP NetWeaver Portal Applications. 2009. -P. 243
3. Web Dynpro [Электронный ресурс]. Режим доступа: <http://scn.sap.com/community/web-dynpro-java>
4. Dyn Page [Электронный ресурс]. Режим доступа: <http://scn.sap.com/docs/DOC-22191>
5. HTML-b [Электронный ресурс]. Режим доступа: http://sapdesignguild.org/resources/htmlb_guidance/index.html

УДК 519.685.3

М.И. Полубелова (3 курс, каф.СП, СПбГУ)
С.В.Григорьев (научный сотрудник JetBrains)

ГЕНЕРАТОР АБСТРАКТНЫХ ЛЕКСИЧЕСКИХ АНАЛИЗАТОРОВ.

Современные языки программирования позволяют во время выполнения программы формировать выражения на других языках и выполнять их. Языки, которые используются для написания таких динамически формируемых выражений, называются встроенными. В качестве примера встроенного языка можно привести динамический SQL: запросы к базе данным формируются при участии пользователя во время выполнения программы. Одной из трудностей применения разнообразных встроенных языков является отсутствие какой-либо информации о правильности составленного выражения до этапа выполнения основной программы. Как следствие, ошибки выявляются лишь во время выполнения программы, что существенно увеличивает затраты на разработку, отладку и сопровождение приложений, в которых используются встроенные языки. Кроме того, встроенные языки являются серьезным источником уязвимостей информационных систем. Например, атаки, основанные на SQL-инъекциях, возможны благодаря использованию в приложениях встроенного SQL.

С другой стороны, для обычных языков программирования существуют интегрированные среды разработки, которые предоставляют разработчику набор различной функциональности, упрощающей процесс разработки, например, автодополнение, рефакторинг и дополнительные статические проверки. Для встроенных языков такая функциональность также была бы полезной.

Несмотря на то, что уже существуют инструменты для работы с динамически формируемыми выражениями, например Alvor [4], есть необходимость в разработке новых подобных инструментов, в которых рассмотрен вопрос обобщения лексического анализа и синтаксического разбора.

Обработка встроенных языков состоит из нескольких этапов, одним из которых является абстрактный лексический анализ [1], [2]. Главная особенность абстрактного анализа – не вычислять все возможные значения динамически формируемой строки, а работать с компактным представлением множества значений. В данном случае это будет граф, являющийся представлением динамически формируемого выражения. Таким образом каждому пути в графе будет соответствовать некоторое возможное значение динамического запроса. Результатом работы анализатора также является граф, ребро которого содержит токен (лексическая единица). Абстрактный лексический анализатор основан на конечном преобразователе (finite-state transducer [3], [5]), который, основываясь на сгенерированном описании автомата, переводит входной граф в граф, содержащий соответствующие спецификации токены. В случае ошибки этого процесса, необходимо указать место ее возникновения. Для этого необходимо реализовать сохранение привязки частей динамически формируемого выражения к исходному коду и привязку лексических единиц внутри каждой части.

Решение поставленной задачи тестировалось на примере разработки генератора абстрактных лексических анализаторов, реализованный на базе инструмента FsLex в рамках проекта YaccConstructor [6], для обработки встроенного языка Transact-SQL. Также в рамках этого проекта ведется разработка плагина к ReSharper с целью предоставления платформы для поддержки встроенных языков в Microsoft Visual Studio IDE. Одновременно с этим проводились эксперименты по ускорению работы генератора — делались замеры времени на разных типах коллекций F#, и происходило сравнение лексических анализаторов YaccConstructor и Alvor.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. Aivar Annamaa, Andrey Breslav, and Varmo Vene. Using abstract lexical analysis and parsing to detect errors in string-embedded dsl statement. Marina Walden and Luigia Petre (Eds.): Proceedings of the 22nd Nordic Workshop on Programming Theory, TUCS General Publication, No 57, November 2010, pp.20-22.
2. Andrey Breslav. Parsing Abstract Strings, University of Tartu // STACC, 2010.
3. Mohri Mehryar. Finite-State Transducers in Language and Speech Processing. Association for Computational Linguistics. — 1997. — <http://www.cs.nyu.edu/mohri/pub/cl1.pdf>
4. Сайт проекта Alvor. — <http://code.google.com/p/alvor/>
5. Мартыненко Б.К. Языки и трансляции. Издательство С.-Петербургского университета, 2002.
6. Сайт проекта YaccConstructor. — <https://code.google.com/p/recursive-ascent/>

УДК 519.681.3

Р.Хаберланд (асп., каф. МО ЭВМ, СПбГЭТУ)
С.А.Ивановский, к.т.н., доцент

ВЕРИФИКАЦИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММ, ИСПОЛЬЗУЮЩИХ ДИНАМИЧЕСКУЮ ПАМЯТЬ, С ПРИМЕНЕНИЕМ ЯЗЫКА ЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Целью работы является создание прототипа системы верификации программ, работающих с динамической памятью. Программы представлены на специальном языке программирования, который близок к языку *C* с объектно-ориентированным расширением и без автоматического сбора мусора. Исследуется комплекс вопросов, связанных с утечкой памяти, с проверкой отсутствия висячих ссылок, неверных обращений к ячейкам памяти, а также непредусмотренных циклических зависимостей при ссылочном связывании ячеек памяти. Для этих целей используется формальное представление динамической памяти. При разработке программной системы верификации в качестве языка реализации используется язык логического программирования Пролог.

С точки зрения использования в программной инженерии существенными требованиями для системы верификации являются гибкость используемой модели памяти в связи с выразительными возможностями, а также локальность спецификаций, возможно, неполных. Высокий уровень абстракции при написании правил на логическом языке в виде дизъюнктов позволяет иметь простую реализацию и дает пользователю при эксплуатации системы возможность добавления новых правил в отдельную от правил верификации часть интерпретируемой Пролог-программы. Свойство локальности спецификаций позволяет специфицировать не всю программу, а лишь некоторые её части, представляющие интерес. Ожидается также, что реализация на языке Пролог обеспечит возможность дальнейшего функционального расширения системы верификации на основе генерации контр-примеров во время логического вывода. Задание правил на логическом языке в виде дизъюнктов соответствует выбранному уровню абстракции и, в том числе, может найти применение в учебных целях.

Грамматика входного объектно-ориентированного языка при синтаксическом анализе задается шаблоном, описывающим правила для системы ANTLR. Статически проверяются семантические ошибки, и затем данная программа преобразуется в Пролог-термы. Процессы синтаксического анализа и этапы верификации разделены и допускают обмен лишь общими Пролог-термами. Предусмотрена возможность расширения этапов перевода и преобразования, например, для графического представления верификации или промежуточных представлений. Проект также дает возможность исследования улучшения откомпилированного кода после статической верификации, например, на стадии сбора мусора, на основе ранее полученных знаний о динамическом расположении и зависимости ячеек памяти.

В качестве спецификации памяти планируется разрешать добавление пользователем индивидуальных предикатов, описывающих состояние памяти в отдельной программе. Для преобразования в нормальную форму, которая позволит избежать приостановки и способствовать преодолению тупиковых состояний, планируется использовать SAT-решатель (например, Why или YNot). Для нахождения новых лемм целесообразно использование абдуктивного подхода к выводу. Рассмотрена возможность применимости предлагаемого подхода в технологиях Microsoft.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

АДАПТАЦИЯ СЕТЕВЫХ ПРИЛОЖЕНИЙ ДЛЯ РАБОТЫ В СЕТИ С ЯЧЕИСТОЙ ТОПОЛОГИЕЙ

Одним из наиболее активно развивающихся в настоящее время направлений связанных с созданием компьютерных сетей, является разработка и построение сетей с ячеистой топологией на основе различных устройств. Ячеистая топология (англ. *Mesh Topology*) — сетевая топология компьютерной сети, основанная на принципе ячеек, в которой каждая рабочая станция сети соединяется с несколькими другими рабочими станциями этой же сети с возможным принятием на себя функций коммутатора для других рабочих станций. Характеризуется высокой отказоустойчивостью, однако это приводит к сложностям настройки и, для проводных сетей, преизбыточному расходу кабеля, так как в подобных сетях каждый компьютер имеет множество возможных путей соединения с другими компьютерами.

Данная работа посвящена разработке программного модуля позволяющего адаптировать существующее сетевое приложение к использованию в mesh сетях путем упрощения настройки сети.

Перед началом общения между устройствами, должен быть сконфигурирован интерфейс mesh-сети, в который входит:

1. Имя сетевого интерфейса, на котором будет работать сеть (например wlan0)
2. Имя сетевого интерфейса mesh-сети, которое отображается в списке сетевых интерфейсов
3. ID mesh-сети, которое необходимо задавать одинаковым для всех участников одной mesh-сети (это как имя точки доступа Wi-Fi)

Далее необходимо сконфигурировать имя сетевого интерфейса ядра, которое должно иметь имя интерфейса mesh-сети и свободный IP адрес.

После создания интерфейса mesh-сети, сетевое приложение может работать в обычном режиме, но используя интерфейс и IP тот, что был назначен при конфигурировании сети.

Созданный в рамках работы программный модуль позволяет частично автоматизировать создание и конфигурацию mesh-сети.

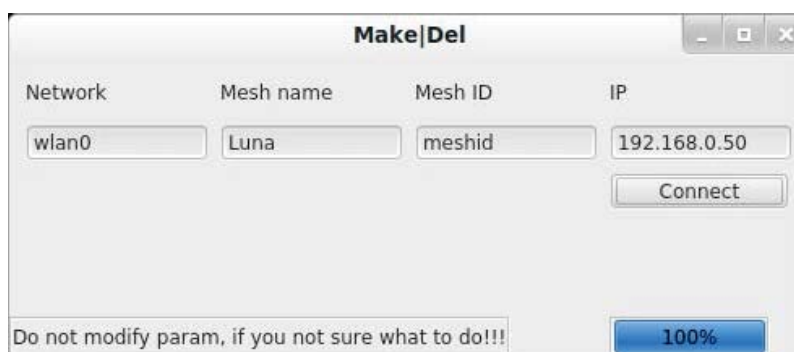


Рис. 3. Пользовательский интерфейс

На рисунке представлен пользовательский интерфейс программного модуля. Для построения mesh сети с его использованием необходимо ввести имя сетевого интерфейса, на котором будет работать сеть (поле Network), имя сетевого интерфейса mesh-сети, которое отображается в списке сетевых интерфейсов (поле Mesh name), ID mesh-сети, которое необходимо задавать одинаковым для всех участников одной mesh-сети (поле Mesh ID) и свободный IP адрес. На данный момент необходимо знать свободные IP адреса, но планируется автоматизировать поиск свободного адреса.

Данную программу можно встроить практически в любую сетевую программу, что поможет ускорить создание mesh-сети. При закрытии программы происходит удаление интерфейса mesh-сети.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Steve Methley. Essentials of Wireless Mesh Networking: Cambridge University Press, 2009, 228с.
2. IEEE Xplore - IEEE 802.11s: The WLAN Mesh Standard: [Электронный ресурс]. URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5416357> (Дата обращения 13.11.2013)

УДК 004.052.42

А.Д. Янкович (4 курс, каф. ИУС, СПбГПУ),
И.В. Никифоров (асп. каф. ИУС, СПбГПУ),
Н.В. Воинов (доц. каф. ИУС, СПбГПУ)

ИНСТРУМЕНТАЛЬНАЯ ПОДДЕРЖКА РЕДАКТИРОВАНИЯ ПАРАМЕТРОВ КОНКРЕТИЗАЦИИ СИМВОЛЬНЫХ ТЕСТОВЫХ СЦЕНАРИЕВ

Тестирование программного обеспечения (ПО) — один из наиболее важных этапов жизненного цикла разработки программного продукта. Современные стандарты создания ПО, такие как, например, Rational Unified Process [1], основаны на применении моделей систем. На основе моделей проводится их верификация и генерация тестовых сценариев. Технология VRS/TAT [2] также основана на применении подхода формальных моделей программ и их верификации. Отличительной особенностью технологии является генерация символьных тестовых сценариев [3], которые позволяют значительно сократить конечное множество тестов.

При формировании конечных тестов, исполняемых на модели, необходимо осуществить подстановку вместо символьных параметров модели их конкретных значений. Для этого используется таблица конкретизации параметров, которая представляет из себя текстовый файл, содержащий соответствие параметров принимаемым в тесте значениям. Каждому тестовому сценарию соответствует свой текстовый файл с параметрами конкретизации.

Ручное редактирование или заполнение таких таблиц очень трудоемкий процесс для разработчика, требующий особой скрупулезности и внимательности. Решение в виде инструмента-редактора, предлагаемое в работе, направлено на то, чтобы облегчить разработчику процесс формирования таблицы соответствий.

Технически предлагаемый редактор реализован в виде окна (view), интегрированного в среду Eclipse (Рис.1), который автоматически загружает содержимое текстового файла, привязанного к тестовому сценарию.

ID	Parameter name	Signal name	Serial number	Parameter indication	Type of parameter	Range	Rule	Range	Value
0	UserPC	u.ackmode...	unchanged_...	type	ACK_MODE	rang...	M	FLIP...	RCIP
1	Receiver	r.ackmode...	unchanged_...	type	ACK_MODE	rang...	M	FLIP...	RCIP
2	Receiver	r.ack_check	unchanged_...	type	BOOL_TYPE	rang...	M	true,...	true
3	ack1	CONFIGURE	0	0	ACK_MODE	rang...	M	FLM...	FL...
4	ack2	CONFIGURE	0	0	ACK_MODE	rang...			
5	ack1	CONFIGURE	1	0	ACK_MODE	rang...			
6	ack2	CONFIGURE	1	0	ACK_MODE	rang...			
7	changed	CNI_MEAS...	2	0	changed	rang...			
8	loc_cni	CNI_CHAN...	3	0	MODCOD	rang...	M	sam...	cha...
9	loc_ackmpeg2	CONFIGURE	4	0	ACK_MODE	rang...			
10	loc_ackmultic...	CONFIGURE	4	0	ACK_MODE	rang...			
11	changed	CNI_CHAN...	5	0	changed	rang...			
12	code	ACM_CAP...	6	0	MODCOD	rang...			
13	loc_reccod	ACM_CAP...	7	0	MODCOD	rang...	M	sam...	cha...
14	loc_reccod	ACM_CAP...	8	0	MODCOD	rang...			
15	ack	ACM_ACK...	12	0	BOOL_TYPE	rang...			
16	ack	ACM_ACK...	13	0	BOOL_TYPE	rang...			
17	loc_ack_payload	ACM_ACK...	14	0	BOOL_TYPE	rang...			
18	code	ACM_CAP...	16	0	MODCOD	rang...	M	sam...	same

Рис. 1. Окно редактирования таблицы конкретизации

Содержимое таблицы связывается с актуальным тестовым сценарием, для которого осуществляется конкретизация. Таблица содержит следующие колонки: ID - «Идентификатор параметра в трассе», Parameter name - «Имя параметра», Signal name - «Имя сигнала с этим параметром», Serial number - «Порядковый номер сигнала в трассе», Parameter indication - «Признак того, что это параметр сигнала», Type of parameter - «Тип параметра», Range - «Диапазон возможных значений параметра», Rule - «Правило подстановки значения для параметра», Range - «Диапазон возможных значений параметра», Value - «Выбранное значение для параметра».

Полученное решение значительно (примерно в 3 раза) уменьшает время редактирования таблиц конкретизации по отношению к тому времени, которое пользователь затрачивал на формирование параметров конкретизации без использования редактора.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Крачтен Ф., Введение в Rational Unified Process : переводное издание / [Пер. с англ. Назаренко А.В.]. - 2-е изд. - М. : Вильямс, 2002. - 239 с.
2. Веселов А.О., Котляров В.П. Автоматизация тестирования в области телекоммуникаций // Научно-технические ведомости СПбГПУ. №4(103). СПб.: Изд-во Политехнического ун-та, 2010. С. 180-185.
3. Никифоров И.В., Петров А.В., Юсупов Ю.В., Генерация формальной модели системы по требованиям, заданным в нотации USE CASE MAP // Научно-технические ведомости СПбГПУ. № 4(103). СПб.: Изд-во Политехнического ун-та, -2010. -С. 191-195.

ПРИМЕР ОПТИМИЗАЦИИ ТЕСТИРОВАНИЯ ПРИ ВНЕДРЕНИИ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ

В любом процессе разработки программного продукта в том или ином виде присутствует фаза контроля качества. Чаще всего она представлена тестированием, на которое может приходиться до 40% всех затрат [1]. Наиболее важна задача снижения времени тестирования в условиях коротких итераций производственного цикла, например, при использовании подхода непрерывной интеграции (continuous integration, CI).

В рамках создания системы тестирования ТАТ сборка и тестирование очередной версии программного комплекса автоматизированы с помощью инструментов, разработанных согласно нуждам проекта. С ростом объёма тестового набора и увеличением сложности тестируемого поведения время, необходимое для выполнения всех тестов, увеличилось. Принятый в проекте процесс разработки подразумевал частый выпуск промежуточных версий и обязательное тестирование перед созданием очередной ревизии кода. Поэтому была поставлена задача оптимизировать средства запуска тестов так, чтобы при дальнейшем увеличении их количества иметь возможность сохранять время тестирования на приемлемом уровне.

В качестве решения было предложено выполнять параллельный запуск тестов. Для этого был разработан и реализован соответствующий параллельный алгоритм. В условиях отсутствия зависимостей между тестами они могут быть выполнены одновременно, и потому возможно эффективное масштабирование их запуска. Для равномерного распределения нагрузки на вычислительные ресурсы было решено выполнять одинаковое количество тестов в каждом потоке исполнения [2]. Такой подход определяет правило масштабирования выполнения тестового набора: при наличии m тестов каждый из n запущенных потоков выполнит m/n тестов. При этом максимально возможное число потоков выполнения равно числу тестов в наборе, минимальное — одному потоку, что соответствует ранее применявшемуся методу автоматизированного запуска тестов.

Изменяя число параллельных процессов исполнения тестового набора, можно определить наиболее эффективное значение для конкретной среды исполнения. Экспериментальная проверка нового подхода на компьютере с четырьмя вычислительными ядрами показала прирост производительности в 3,86 раза по сравнению с последовательным автоматизированным запуском. С увеличением числа потоков выполнения ускорение выполнения сохранялось с поправкой на дополнительные издержки, связанные с реализацией многопоточности.

В результате выполнения работы было обеспечено приемлемое время запуска тестов, а также возможность его масштабирования. Дальнейшим направлением разработки является реализация планирования выполнения тестов на основе информации об их структуре и времени предыдущего запуска.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Veselov A., Kotlyarov V. Testing Automation of Projects in Telecommunication Domain // Proceedings of the 4th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2010), June 1-2, 2010 – Nizhny Novgorod, Russia
2. Тютин Б.В., Веселов А.О., Котляров В.П. Масштабирование выполнения тестового набора при автоматизированном тестировании // Научно-технические ведомости СПбГПУ. № 3 (174). СПб.: Изд-во Политехнического ун-та, 2013

УДК 621.319

В. А. Мухин (асп. 1 курс, каф. информатики, СПбГУ),
В. О. Сафонов, д.т.н., проф., каф. информатики, мат.-мех. фак-т, СПбГУ

ИСПОЛЬЗОВАНИЕ C# В КАЧЕСТВЕ СКРИПТОВОГО ЯЗЫКА

Целью работы является исследование возможности использования языка C# для создания сценариев работы ОС - *скриптов*. В настоящее время для этой цели используются *bash*-скрипты в *nix-системах, *batch*-скрипты в Windows-системах, а также специализированные языки, например, Perl, Python или Ruby.

C# обладает выразительным и достаточно простым синтаксисом, а также обширными возможностями создания лаконичного и емкого кода (например, лямбда-выражения и анонимные типы [1, 2]), что делает его весьма привлекательным для использования в контексте разработки скриптов.

Данную задачу планируется решить путем разработки *shell*-скрипта, получающего на вход при запуске конструкции языка C# (не заключенные в класс). Задачей скрипта будет генерация текста C#-класса, добавление в него полученной логики, передача сгенерированного кода компилятору .NET или Mono и запуск выходного бинарного файла.

Конечно, при данном подходе производительность будет ощутимо ниже, чем при использовании скриптовых языков, но в прикладных задачах, не включающих в себя работу с большими объемами данных (например, перемещение нескольких десятков файлов или разбор нескольких строк в тексте) производительность не критична, а так как современные ЭВМ зачастую обладают несколькими гигабайтами оперативной памяти и имеют многоядерные процессоры, то использовании промежуточной компиляции не является существенной преградой.

Из преимуществ использования C# как скриптового инструмента стоит отметить то, что разработчикам не придется осваивать дополнительный интерпретируемый язык.

Следующим этапом планируется разработка мультиплатформного консольного C#-приложения, работающего по тому же принципу.

Возможным третьим этапом будет оптимизация приложения путем добавления многопоточности и опциональности.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4. – М.: Вильямс, 2011.
2. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. – СПб.: Питер, 2012.

УДК 004.052.42

А.П.Маслаков (5 курс, ИУС), И.В. Никифоров (асс, ИУС)
Л.П.Котлярова (проф., ИУС СПбГПУ)

ПОИСК И ИСПРАВЛЕНИЕ ОШИБОК ПРИ ФОРМАЛИЗАЦИИ ТРЕБОВАНИЙ

Большинство ведущих мировых организаций, занимающихся разработкой программно-аппаратных комплексов, используют развитые технологии и стандарты создания программного обеспечения. Одним из таких стандартов является стандарт - Rational Unified Process (RUP)[1]. В основе подхода RUP лежит инкрементальный и итерационный подход, в соответствии с которым разработка системы осуществляется в несколько этапов и включает в себя: разработку требований, создание моделей проектируемой системы различной степени детальности с целью их дальнейшего анализа и верификации, а также реализацию, тестирование и развертывание системы.

Несмотря на чётко формализованные фазы RUP, основные аспекты описанного в нем жизненного цикла разработки можно увидеть и в других технологиях. Одной из технологий, в которых наблюдается отражение итерационных этапов проектирования требований, верификации и тестирования, является технология VRS/TAT [2]. Она основана на использовании высокоуровневых UCM-спецификаций [3] для создания формальных функциональных моделей программных продуктов, которые в дальнейшем анализируются, верифицируются и относительно которых генерируются тестовые сценарии. Среди всех этапов технологии VRS/TAT особый интерес представляет этап создания формальной модели системы на языке UCM по неформализованным требованиям. Созданная функциональная модель в дальнейшем преобразуются в модель на языке VRS-верификатора.

Формализацией требований для промышленных проектов обычно занимается команда разработчиков, каждому из членов которой предоставляется часть системы для формализации её в виде UCM-модели. Затем, все элементы и части системы интегрируются в единую формальную модель, с которой работают доказательные средства верификации и тестирования. В процессе интеграции в обобщенной UCM-модели могут появиться ошибки, которые препятствуют дальнейшему анализу и проектированию системы.

Для каждой разрабатываемой части обобщенной UCM-модели можно выделить следующие подэтапы: создание поведения и структуры модели[4], наполнение модели потоком данных [5], анализ синтаксических и семантических ошибок. Даже если для каждой из частей в отдельности была создана такая модель, в которой при анализе не будет

обнаружено ошибок, то после интеграции всех компонентов в единую модель могут возникнуть ошибки, связанные с пересечением областей видимости имен, с несоответствием интерфейсов или наличием структурных ошибок интеграции.

Для того чтобы использовать обобщенную модель, необходимо провести ее двухуровневый анализ: на уровне UCM-модели выявляются ошибки в структуре модели, ошибки проектирования многопоточных конструкций и синтаксические ошибки разработки, а на втором уровне осуществляется проверка правильности преобразования UCM-модели в модель на языке VRS-верификатора.

Если анализ, проводимый на уровне UCM-модели, дает разработчику информацию о точном местоположении ошибок, то при анализе на уровне модели VRS-верификатора, необходимым фактором является отображение ошибок на уровень UCM.

В работе была решена задача поиска и отображения на уровень UCM-модели ошибок, выявленных на стадии анализа модели VRS-верификатора, что значительно упростило отладку обобщенной UCM-модели и позволило использовать ее для дальнейшей генерации тестовых сценариев.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Крачтен Ф., Введение в Rational Unified Process : переводное издание / [Пер. с англ. Назаренко А.В.]. - 2-е изд. - М. : Вильямс, 2002. - 239 с.
2. Веселов А.О., Котляров В.П. Автоматизация тестирования в области телекоммуникаций // Научно-технические ведомости СПбГПУ. №4(103). СПб.: Изд-во Политехнического ун-та, 2010. С. 180-185.
3. ITU Recommendation Z.151. User Requirement Notation (URN), 2008.
4. Дробинцев П.Д., Никифоров И.В., Котляров В.П. Методика проектирования тестов сложных программных комплексов на основе структурированных UCM моделей // Научно-технические ведомости СПбГПУ. № 3(174). СПб.: Изд-во Политехнического ун-та, -2013. -С. 99-105.
5. Никифоров И.В., Петров А.В., Юсупов Ю.В., Генерация формальной модели системы по требованиям, заданным в нотации USE CASE MAP // Научно-технические ведомости СПбГПУ. № 4(103). СПб.: Изд-во Политехнического ун-та, -2010. -С. 191-195

УДК 004.42

Н. В. Крупный, Е. П. Назарьев, А. О. Веселов, Б. В. Тютин (СПбГПУ)

АВТОМАТИЗАЦИЯ РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ НА ПРИМЕРЕ СИСТЕМЫ ТАТ

Тестирование — неотъемлемый этап любого современного процесса промышленной разработки программного обеспечения. Затраты на тестирование могут достигать 40% всех затрат, связанных с разработкой [1]. Поэтому вопросы автоматизации и эффективности тестирования являются весьма актуальными в наше время.

Данная статья содержит описание решения по автоматизации процесса предрелизного регрессионного тестирования инструмента ТАТ [2,3,4]. Процесс, применяемый при разработке данного инструмента, подразумевает короткие итерации и частый выпуск

промежуточных версий программы. Выпуск каждой новой версии, согласно процессу, требует проведения регрессионного тестирования с целью удостовериться, что внесенные изменения не привели к регрессии качества. Выполнение этого требования привело к созданию соответствующего тестового набора. В начале он содержал достаточно небольшое количество тестов, но число их быстро росло по мере развития проекта. Изначально предполагался запуск каждого теста вручную в отдельности, что было приемлемо на начальной стадии, когда количество тестов было небольшим. Однако, с ростом их числа пред релизный прогон тестового набора стал занимать гораздо больше времени. Издержки на поддержание процесса разработки значительно возросли. Была поставлена задача автоматизировать пред релизное регрессионное тестирование с целью минимизации временных затрат.

Для достижения поставленной цели было принято решение разработать инструмент, осуществляющий автоматический запуск тестового набора и отображение информации о результатах тестирования с диагностикой ошибок. Также данный инструмент, по завершении выполнения, должен выдавать статистику о количестве пройденных и проваленных тестов. Для реализации инструмента был выбран скриптовый язык Python.

В результате создания и внедрения такого инструмента запуска было получено значительное преимущество по времени прогона тестового набора: при ручном подходе, в среднем на прогон одного теста и анализ результатов уходит приблизительно 3 минуты, автоматизация же позволила сократить это время до 2 минут 20 секунд (в среднем). Таким образом, было получено повышение производительности процесса тестирования приблизительно на 22%.

Однако, дальнейший рост числа тестов в регрессионном тестовом наборе привел к тому, что увеличение производительности, достигнутое при помощи автоматизации, оказалось недостаточным. В качестве решения было предложено реализовать параллельный запуск тестов. Инструмент запуска был модифицирован для выполнения данной задачи. Архитектура итоговой реализации инструмента представлена на рис. 1.

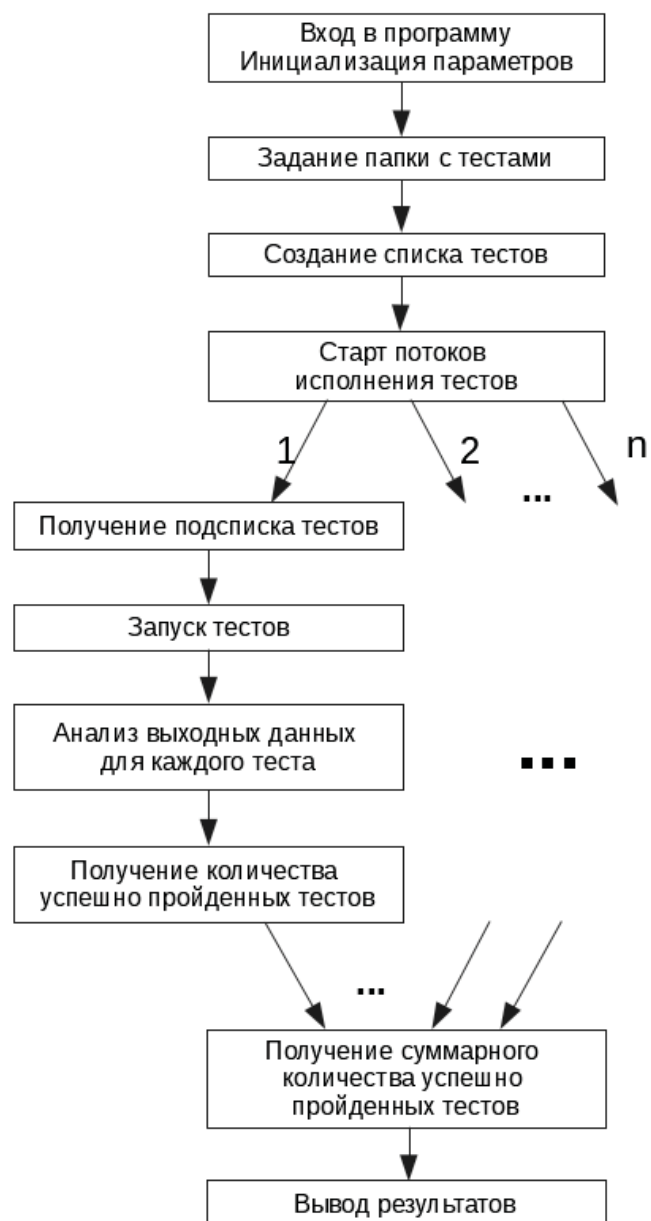


Рис. 1. Схема работы инструмента

Поскольку все тесты независимы, то могут быть выполнены одновременно. Для равномерного распределения нагрузки на вычислительные ресурсы при параллельном запуске тестов требуется выполнить одинаковое их количество в каждом потоке исполнения [5]. Такой подход определяет правило масштабирования выполнения тестового набора: при наличии m тестов каждый из n запущенных потоков выполнит m/n тестов. При этом максимально возможное число потоков выполнения равно числу тестов в наборе, минимальное — одному потоку, что соответствует ранее применявшемуся методу автоматизированного запуска тестов.

Управление инструментом осуществляется при помощи интерфейса командной строки. Поддерживаются следующие опции:

- `file` - печать результатов в файл;
- `ignoretest` - список тестов, которые не должны быть включены в тестовый набор;
- `help` - вывод справочной информации;

- tests - список тестов, которые подлежат запуску;
- threads - число параллельных процессов исполнения тестового набора;
- verbose - флаг вывода вспомогательной информации.

Опция --tests задает список тестов, которые должны быть включены в тестовый набор, при этом все другие тесты игнорируются. Если эта опция не указана, то берутся все тесты в текущей директории, не входящие в список, заданный опцией --ignoretest. При пересечении множеств тестов, заданных с помощью --tests и --ignoretest, предпочтение отдается опции --tests.

По умолчанию (при отсутствии опций --tests и --ignoretest) инструмент запускает все тесты из текущей директории. Стоит также отметить, что значение опции --threads по умолчанию принимается равным числу вычислительных ядер общего назначения на данной рабочей машине.

Изменяя число параллельных процессов исполнения тестового набора (варьируя значение опции --threads), можно определить наиболее эффективное значение для конкретной среды исполнения. Проведенные эксперименты показали, что на рабочей машине с четырьмя ядрами, как и ожидалось, наибольшая производительность достигается при значении --threads равном четырем: при увеличении значения от одного до четырех производительность растёт практически линейно, а с пяти начинает падать так как количество процессов превышает число ядер, и процессы начинают конкурировать за процессорное время между собой.

Применение рассматриваемого подхода к параллельному запуску тестов на рабочей машине с четырьмя вычислительными ядрами дало прирост производительности в 3,86 раза по сравнению с последовательным автоматизированным запуском и практически в четыре раза по сравнению с ручным запуском.

В результате проведенной работы был создан инструмент параллельного запуска тестового набора для системы ТАТ, а также разработана методика определения оптимального количества процессов для конкретной среды исполнения. Внедрение данного инструмента в Процесс разработки позволило в значительной мере сократить издержки на предрелизное регрессионное тестирование, что, в свою очередь, позитивно отразилось на качестве тестируемого программного продукта так как позволило существенно расширить набор регрессионных тестов без потери скорости тестирования.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. A. Veselov, V. Kotlyarov. Testing Automation of Projects in Telecommunication Domain // Proceedings of the 4th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2010), June 1-2, 2010 – Nizhny Novgorod, Russia.
2. А. О. Веселов, А. С. Иванов, Б. В. Тютин, В. П. Котляров. Автоматическая настройка тестового окружения телекоммуникационных проектов // Научно-технические ведомости СПбГПУ. № 4(128). СПб.: Изд-во Политехнического ун-та, 2011, с.149–152.
3. Б. В. Тютин, И. В. Никифоров, В. П. Котляров. Построение системы автоматизации статической и динамической проверки требований к программному продукту // Научно-технические ведомости СПбГПУ. № 4 (152) 2012. СПб.: Изд-во Политехнического ун-та. – 2012. – С 119-123.
4. А. О. Веселов, В. П. Котляров. Автоматизация тестирования в области телекоммуникаций // Научно-технические ведомости СПбГПУ. № 4(103). СПб.: Изд-во Политехнического ун-та, 2010, с.180–185.

5. Б. В. Тютин, А. О. Веселов, В. П. Котляров. Масштабирование выполнения тестового набора при автоматизированном тестировании // Научно-технические ведомости СПбГПУ. № 3 (174). СПб.: Изд-во Политехнического ун-та, 2013.

УДК 004.415.5

А. П. Маслаков, И. В. Никифоров, В. П. Котляров (СПбГПУ)

АВТОМАТИЗАЦИЯ ЭТАПОВ ПРОЕКТИРОВАНИЯ ТЕСТОВЫХ СЦЕНАРИЕВ

На сегодняшний день одной из наиболее эффективных технологий повышения качества программного продукта является технология VRS/TAT[1]. Она позволяет выявить ошибки в спецификациях на самых ранних этапах создания системы, а так же произвести генерацию тестового набора по формальной модели системы на языке UCM[2], который обеспечивает покрытие по выбранному интегральному критерию[3] и обеспечивает 100% проверку требований.

Обзор существующей технологии и ее недостатки. На рис.1 представлена обобщенная схема существующей технологии.

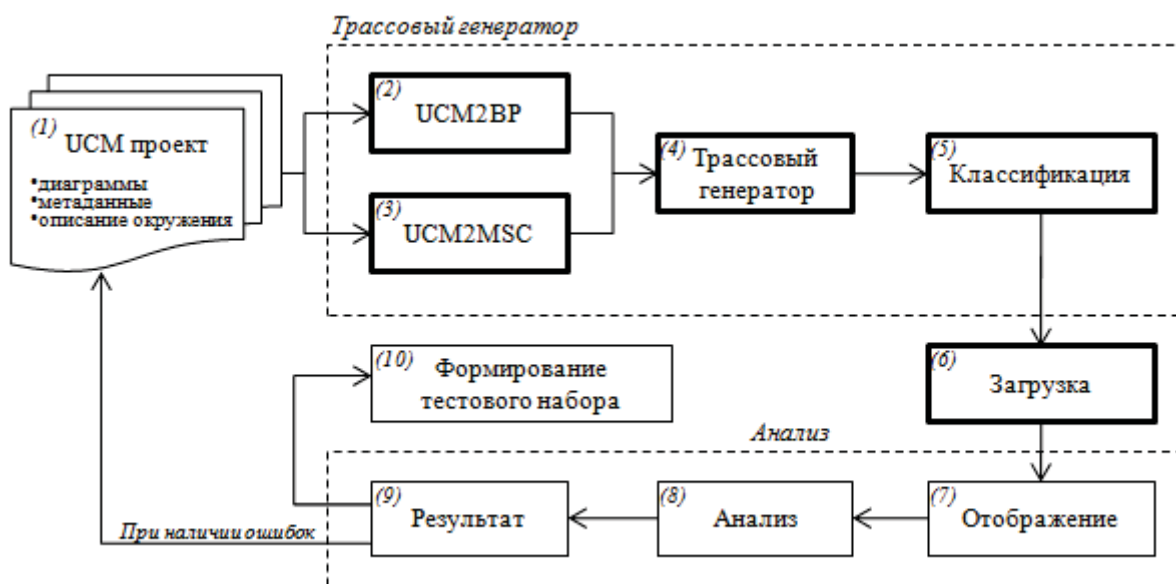


Рис. 1.

Обобщенная схема технологии VRS/TAT.

В технологии требования на систему представляются в виде высокоуровневых UCM спецификаций(1), которые содержат:

1. набор UCM диаграмм, описывающие поведение системы;
2. метаданные, выражающие поток данных в системе;
3. описание окружения, содержащее информацию о взаимодействующих сущностях и атрибутах системы.

По UCM проекту, с помощью преобразователя UCM2BP(2)[4] генерируется формальная модель на языке базовых протоколов, с которой работают доказательные средства. Так же инструментом UCM2MSC(3)[5] генерируется множество гидов, обеспечивающее покрытие по выбранному интегральному критерию покрытия. На вход трассового генератора(4) поступает

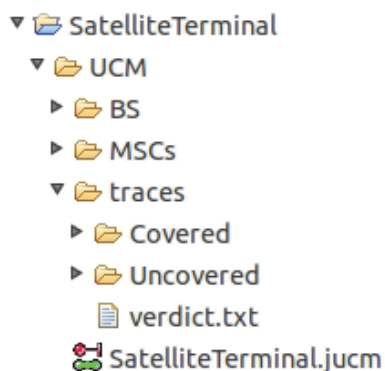
модель системы на языке базовых протоколов и множество гидов, после чего в дереве поведения ищутся те пути, которые описаны гидами. Итогом трассовой генерации является файл verdict.txt, содержащий информацию о покрытии гидов трассами и информацию о классах трасс, а так же само множество трасс(5).

Для анализа трасс пользователю приходится вручную, на основе файла verdict.txt, классифицировать трассы по директориям и осуществлять их загрузку(6) в инструмент анализа UCM EVA[6]. В ходе изучения классов тестов производится визуальное отображение(7) трасс на UCM диаграммах, анализ(8) покрытия или выявление причин не покрытия. После чего формируется вердикт и результат анализа(9), и пользователь, либо вносит изменения в исходный UCM проект (при наличии ошибок), либо формирует тестовый набор(10).

Несмотря на все преимущества данной технологии, она обладает недостатками, связанными с отсутствием автоматизации запусков инструментов (шаги (2), (3), (4) на рис.1), отсутствием автоматической классификации трасс (шаг (5) на рис.1) и отсутствием автоматического обмена данными между инструментами (шаг (6) на рис.1). А как следствие данных недостатков увеличивается время создания эффективного тестового набора, а так же от пользователя требуется внимательность и соблюдение строгой последовательности запуска и настройки инструментов.

В работе перед автором *ставилась задача* автоматизации процесса генерации тестовых сценариев, *целью* которой является сокращение трудоемкости разработки эффективного тестового набора.

Полной автоматизации процесса генерации тестовых сценариев препятствует отсутствие возможности автоматического запуска инструментов UCM2BP, UCM2MSC и EVA. Такая ситуация была связана с необходимостью ручной настройки инструментов, а именно: ручной настройки расположения входной и выходной информации для каждого из инструментов.



Для автоматизации запуска в работе были *предложены и реализованы следующие подходы*: строго заданная структура директорий проекта (рис.2), автоматическая классификация тестовых сценариев по директориям и автоматическая загрузка тестового набора в инструмент анализа.

Для инструмента UCM2BP входной информацией является *.jucm файл, располагающийся в корне директории “UCM”. А выходной информацией является директория “BSs”. Для инструмента UCM2MSC входной информацией является директория “BSs” и *.jucm файл. Выходной информацией является директория “MSCs”.

На вход трассового генератора поступает модель системы на языке базовых протоколов, хранимых в директории “BSs”, и множество гидов из директории “MSCs”. Выходом с трассового генератора является директория “trace”, с автоматически классифицированными по директориям трассам.

Классификация трасс по директориям осуществляется автоматически на основе разбора файла verdict.txt: трассы, обеспечивающие полное покрытие гидов помещаются в директорию “Covered”, а трассы, которые не покрывают гиды до конечной точки, помещаются в директорию “Uncovered”. В случае отсутствия гидов при генерации, трассы классифицируются по следующим критериям:

- дедлок трассы — трассы, которые достигли тупикового состояния в системе;
- циклы — трассы, которые обнаружили циклы в системе;
- целевые трассы — трассы, которые достигли особого состояния UCM диаграммы. Под особым состоянием понимается отмеченный пользователем UCM элемент маркером «GOAL».

Инструмент анализа UCM EVA[7] автоматически определяет появление новых ресурсов в структуре директорий проекта и загружает их для отображения покрытия на UCM и выявления причин расхождения трассы и гида, в случае наличия тестовых сценариев в директории “traces/Uncovered”.

Выводы: Используя подходы к автоматизации процесса генерации тестовых сценариев, реализованные в работе, технологию VRS/TAT можно рассматривать как “push-button”.

Автоматизация этапов проектирования тестовых сценариев позволяет сократить время получения эффективного тестового набора и сделать использование инструментов более удобным для пользователя. Данные подходы интегрированы в технологию VRS/TAT и применены в двух телекоммуникационных проектах: SMTP и CDMA.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. И. С. Ануреев, С. Н. Баранов, Д. М. Белоглазов. Средства поддержки интегрированной технологии для анализа и верификации спецификаций телекоммуникационных приложений. // Труды СПИИРАН-2013-№3(26). 2013. С. 349-384.
2. ITU Recommendation Z.151. User Requirement Notation (URN), 2008.
3. И. В. Никифоров, П. Д. Дробинцев, В. П. Котляров. Интегральные критерии проверки требований к программному обеспечению // Научно-технические ведомости СПбГПУ. № 3(174). – СПб.: Изд-во Политехнического ун-та, -2013. -С. 111-118.
3. И. В. Никифоров, А. В. Петров, Ю. В. Юсупов. Генерация формальной модели системы по требованиям, заданным в нотации USE CASE MAP // Научно-технические ведомости СПбГПУ. № 4(103). – СПб.: Изд-во Политехнического ун-та, -2010. -С. 191-195.
4. П. Д. Дробинцев, В. П. Котляров, И. Г. Черноруцкий. Автоматизация тестирования на основе покрытия пользовательских сценариев. // Научно-технические ведомости СПбГПУ. № 4 (152). – СПб.: Изд-во Политехнического ун-та. – 2012. – С. 123-126.
5. А. П. Маслаков, И. В. Никифоров, В. П. Котляров. Методы поиска расхождений гайдов и трасс в анализаторе последовательностей событий // Технологии Microsoft в теории и практике программирования. Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада. 26 марта 2013 г. – СПб.: Изд-во Политехнического ун-та. – 2013. – С. 72-73.

6. И. В. Никифоров, А. В. Петров, В. П. Котляров. Статический метод отладки тестовых сценариев, сгенерированных с помощью эвристик // Научно-технические ведомости СПбГПУ. № 4(152). – СПб.: Изд-во Политехнического ун-та, -2012. -С. 114-119.

УДК 004.056

Д. В. Иванов, Д. А. Москвин (СПбГПУ)

ИССЛЕДОВАНИЕ И АНАЛИЗ МЕХАНИЗМА РАЗРЕШЕНИЙ ОПЕРАЦИОННОЙ СИСТЕМЫ GOOGLE ANDROID С ТОЧКИ ЗРЕНИЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ СИСТЕМЫ

Операционная система (ОС) Google Android использует множество защитных механизмов для обеспечения безопасности пользовательских данных, хранящихся на устройстве. Одним из таких механизмов является механизм разрешений, который позволяет ОС обеспечить и контролировать распределение прав и привилегий приложений. Точная и надежная работа механизма разрешений критически важна как для защиты пользовательских данных, так и для безопасности всей системы в целом. В работе получены новые научные знания о подходах, с помощью которых могут осуществляться нарушения механизма разрешений, тем самым снижая общий уровень безопасности системы, также описаны меры, которые помогут нейтрализовать описанные угрозы.

Разрешением в ОС Google Android называется предоставление пользователем полномочий приложению на выполнение различных функций и видов доступа на устройстве [1]. Существует большое количество различных разрешений. Приложение при установке запрашивает у пользователя полномочия на использование этих разрешений. При этом, если пользователь мобильного устройства принял решение, что приложение запрашивает избыточный список прав, он не может его корректировать, а может лишь отказаться от установки данного приложения. Проверить разрешения можно не только перед установкой новых программ, но и у уже установленных приложений во вкладке «Настройки - Приложения - (Название приложения) – Разрешения» [2]. Самыми популярными разрешениями являются:

- осуществление телефонных звонков;
- отправка СМС;
- доступ к сведениям об аккаунтах пользователя;
- доступ к информации о местоположении пользователя;
- доступ к информации о геолокации устройства;
- доступ к списку контактов.

Как и любой другой механизм, механизм разрешений имеет свои слабые стороны, которые могут быть использованы злоумышленником для достижения своих целей. Среди недостатков механизма разрешений можно перечислить следующие:

1. Одной из важнейших проблем механизма разрешений является их низкая четкость, детерминированность и понятность пользователю. Например, рассмотрим разрешение доступа к сети Интернет [3]. Пусть устанавливаемое приложение является ИС-клиентом, тогда доступ в сеть Интернет для него обязателен и запрос такого разрешения выглядит вполне оправданным. Однако такому приложению вовсе необязательно использовать для своей работы другие протоколы, но механизм разрешений не позволяет запретить ему делать это.

2. Механизм разрешений позволяет решить большое количество проблем, связанных с безопасностью устройства, но все же не спасает от угроз, связанных с человеческим фактором – невнимательностью и низкой грамотностью пользователя в области компьютерной безопасности. Например, это выражается в том, что они не обращают внимание на запрашиваемые устанавливаемым приложением разрешения. Это может привести к тому, что приложение получит избыточный список привилегий и станет способно нарушить безопасность устройства.
3. Механизм разрешений не позволяет защитить данные пользователя от использования устанавливаемым приложениям уязвимостей для получения разрешений неуказанных в списке при установке [4].
4. Использование злоумышленником методов социальной инженерии. На этапе проектирования злоумышленник в состоянии, основываясь на том какие разрешения требуются для выполнения зловредной нагрузки, подобрать подходящий тип приложения, таким образом «оправдывая» список запрашиваемых привилегий. Более того злоумышленнику вовсе необязательно реализовывать заявленный функционал, ему достаточно всего лишь создать «заглушки», делающие вид легитимной работы приложения [5]. Понять, что приложение не выполняет заявленных функциональных возможностей, пользователь сможет лишь после установки приложения. К этому моменту можно закрепиться в системе таким образом, что удаление приложения не нарушит работы зловредной нагрузки.

Изучение слабых сторон механизма разрешений дает возможность разработать меры по увеличению его эффективности. Перечисленные проблемы можно решить используя следующий комплекс мер:

1. Введение более точной детализации разрешений принятых в ОС Google Android;
2. Повышение квалификации пользователя устройства, а также более внимательное отношение к устанавливаемым приложениям;
3. Постоянное отслеживание и своевременная установка пользователем устройства патчей и обновлений, закрывающих уязвимости в системе;
4. Максимально тщательная проверка приложений еще до момента их установки на устройство пользователя, способная распознать приложения-заглушки, работающие «вхолостую».

Выводы. Таким образом, в работе рассмотрен ряд подходов по обходу механизма разрешений ОС Google Android. Полученные результаты позволили сформулировать комплекс мер по усовершенствованию данного механизма безопасности. Механизм разрешений ОС Google Android в какой-то степени позволяет повысить уровень безопасности пользовательских данных, но делает это недостаточно эффективно. Использование мер по усовершенствованию механизма разрешений способно повысить его эффективность, а значит и укрепить безопасность всей системы в целом.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

Результаты исследований позволяют начать проведение более глубоких и детальных исследований безопасности как рассматриваемого механизма разрешений, так и всей системы ОС Google Android.

ЛИТЕРАТУРА:

1. Р. Л. Смелянский. Проблемы современных компьютерных сетей. – СПб.: Материалы XIX всероссийской научно-методической конференции «Телематика», 2012.
2. N. McKeown. OpenFlow: Enabling Innovation in Campus Networks. – Стэнфорд: Stanford University, 2008.
3. I. Yamahata. Software Defined Networking, openflow protocol and its controllers. – Чикаго.: LinuxCon, 2010
4. H. Uppal. OpenFlow Based Load Balancing . – Вашингтон.: University of Washington, 2012.
5. P. Porras. A Security Enforcement Kernel for OpenFlow Networks. – Техас.: Texas A&M University, 2012
6. Bypassing Android Permissions: What You Need to Know [электронный ресурс] // Сайт компании Trend Micro – Электрон. дан. – 2012. – Режим доступа: <http://blog.trendmicro.com/trendlabs-security-intelligence/bypassing-android-permissions-what-you-need-to-know/>, свободный. – Загл. с экрана. Дата обращения: 10.11.2013.

УДК 004.056

А. Ю. Кухта, В. В. Платонов (СПбГПУ)

ОБНАРУЖЕНИЕ ВТОРЖЕНИЙ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Генетические алгоритмы – это адаптивные методы поиска, которые в последнее время используются для решения задач оптимизации [1]. Данный класс алгоритмов относится к эволюционным. Идея генетического алгоритма основывается на концепции естественного отбора Ч. Дарвина. С его помощью решаются оптимизационные задачи с использованием методов естественной эволюции, таких как селекция, скрещивание и мутация. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

Генетические алгоритмы можно применить для создания простых правил поиска атак в сетевом трафике. Эти правила используются для того, чтобы отличить нормальные соединения от аномальных, причем аномальные соединения рассматриваются как атаки или вторжения. Функции такого обнаружения реализуются системами обнаружения вторжений [2].

Система обнаружения вторжений (СОВ) — программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет [3].

Основными задачами использования генетических алгоритмов для обнаружения вторжений являются следующие: определение качества правила, каким образом нужно проводить селекцию и какой метод скрещивания применять.

Целью данной работы являлось исследование возможностей реализации прототипа СОВ с использованием генетических алгоритмов. Основная задача заключалась в том, чтобы формировать правила, которые бы находили все атаки данного вида и сводили к минимуму ошибки первого рода.

Реализация системы обнаружения вторжений с применением генетического алгоритма состоит из следующих этапов:

1. Обработка файлов аудита, которые включают в себя нормальные и аномальные соединения и представляют собой файл с перечислением значений параметров протокола.
2. Следующий этап заключается в определении набора условий, которые будут задействованы при создании правил, с помощью которых будут определяться заданные атаки.
3. Следующий этап состоит в создании правил, используя полученный набор условий. Начальный набор условий для правил задается вручную и каждое правило состоит из четырех условий. Допускается введение нескольких условий для отдельных параметров.
4. После того как популяция правил задана, они применяются на определенной части дампа (обучающее множество). Полученные данные об их работе применяются для определения значений функции приспособленности (fitness function) для каждого правила [4].
5. Основываясь на полученных значениях функции приспособленности, правила сортируются в порядке возрастания этих значений и полученная иерархия делится на две части для скрещивания [5].
6. Правила из первой части скрещиваются со случайно выбранным правилом из всей популяции. Таким образом, получается новая популяция.
7. Этапы с 4 по 6 повторяются, пока не найдется правило со значением функции приспособленности равном 1 или данный цикл не повторится 20 раз.

Было реализовано программное средство, формирующее правила для обнаружения. Для этого средства выбраны методы селекции, способы скрещивания и функция приспособленности для определения того, насколько хорошо получившиеся правила осуществили обнаружение. Разработанный прототип COB должен обнаруживать следующие виды атак (использовались дампы DARPA 2000):

1. Yaga (относится к классу атак U2R);
2. PortswEEP (принадлежит классу атак Probe);
3. Secret (относится к классу атак Data);
4. Snmpget (принадлежит классу атак).

Проведенные эксперименты показали хорошие результаты. Для трех атак удалось получить правила, которые находили все атаки и лишь для одного примера вредоносного программного обеспечения данный результат не был достигнут.

В дальнейшем планируется расширить базу атак для увеличения эффективности данного средства обнаружения вторжений. Также необходимо оптимизировать программу и снизить количество ошибок первого рода для исключения случаев, когда нормальные пакеты принимаются за аномальные. Дальнейшее исследование направлено на выбор функций приспособленности, чтобы улучшить результаты работы программного прототипа.

Выводы. Разработанное программное средство дает хорошие результаты при обнаружении вторжений, если правила охватывают отличительные признаки атаки. В других случаях могут возникать ложные срабатывания. Данный подход можно применять как для защиты отдельного персонального компьютера, так и сети компьютеров.

Результаты работы программы:

- Для атаки Yaga удалось получить правило, которое находит 100% атак и не имеет ошибок первого рода.
- Для вредоносной программы portswEEP удалось получить правило, которое определяет все 100% атак без ошибок 1-го рода.

- Атака Secret имеет наихудшие результаты. Правило обнаруживает все ошибки, но имеет 25% процентов ошибок первого рода от общего количества пакетов.
- Для атаки Snmpget удалось получить правило, которое обнаруживает 100% атак и не имеет ошибок первого рода.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Т. В. Панченко. Генетические алгоритмы. - Астрахань: Издательский дом «Астраханский университет», – 2007. – 87 с.
2. D. F. Hougen. Improved off-line intrusion detection using a genetic algorithm.- School of computer science, University of Oklahoma Norman, OK, USA. - 2000. - 8 с.
3. В. В. Платонов. Программно-аппаратные средства обеспечения информационной безопасности автоматизированных систем. – М.: Издательский центр «Академия» - 2006. – 240 с
4. P. A. Diaz-Gomez. Analysis and Mathematical Justification of a Fitness Function used in an Intrusion Detection System. - Ingenieria de Sistemas Universidad El Bosque Bogota, Columbia – 2005. - 3 с
5. R. H. Gong. A software implementation of a genetic algorithm based approach to network intrusion detection. - School of computing, Queen's University, Kingston, Ontario, Canada. - 2005. - 8 с.

УДК 004.451

П. А. Опарин, П. Д. Дробинцев, Л. П. Котлярова

РАЗРАБОТКА МЕТОДИКИ ОБУЧЕНИЯ ОПЕРАЦИОННЫМ СИСТЕМАМ РЕАЛЬНОГО ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ ЯЗЫКОВ МОДЕЛИРОВАНИЯ.

В настоящее время разработка программного обеспечения (ПО) невозможна без использования средств, предоставляемых операционной системой. Это обусловлено, в первую очередь, сложностью работы с современным аппаратным обеспечением, которое предоставляет пользователю огромное количество возможностей, но в то же время, использование этих возможностей без абстрагирования от аппаратного уровня привносит в процесс разработки ПО существенные сложности, связанные с переносимостью, надежностью, скоростью работы и другими характеристиками ПО. Операционная система (ОС) предоставляет программистам удобную абстракцию для использования аппаратного обеспечения компьютера, а также является менеджером ресурсов [1,2].

Обучение специалистов в области операционных систем без использования абстрактных моделей на которых студенты смогли бы получить начальное представление об операционных системах требует больших трудозатрат. Данная работа основывается на модели операционной системы реального времени(ОСРВ) стандарта OSEK [3].

ОСРВ предназначены для обеспечения интерфейса к критическим по времени ресурсам систем реального времени. Основной задачей в таких системах является время выполнения обработки данных.

Стандарт OSEK изначально позиционировался для автомобильной промышленности, с целью повышения уровня безопасности и предсказуемости при управлении автомобилями. Однако, он получился универсальным и его использование не ограничилось этой областью. Стандарт

OSEK/VDX состоит из трёх частей – стандарт для операционной системы (OS), коммуникационный стандарт (COM) и стандарт для сетевого менеджера (NM). В области данной работе был использован стандарт OS.

Для разработки модели ОСВР необходимо было выбрать язык графического моделирования. Среди множества языков графического моделирования были выбраны 3 наиболее распространенных – MSC, SDL и UML[4]. Ниже приведена таблица сравнения этих языков в соответствии с необходимыми критериями.

Таблица 1. Сравнение языков моделирования

Критерий	MSC	SDL	UML
1.Наличие графической нотации	+	+	+
2.Наличие современных кодогенераторов	-	±	+
3.Построение модели поведения	-	+	+
4.Построение протокола взаимодействия	+	-	+
5. Наличие инструкции	+	+	+
6.Удобство описания ОС	±	+	+
7.Распространенность стандарта	+	±	+

Как видно из таблицы язык UML удовлетворяет всем критериям, именно поэтому он был выбран в качестве языка моделирования системы.

Среди множества сред разработки для создания модели был выбран инструмент разработки Rational TAU Telelogic 4.3 [5]. Основным критерием при выборе средств разработки была его распространенность в промышленной среде.

Целью данной работы является разработка методики обучения студентов кафедры ИУС курсу операционных систем на основе модели операционной системы реального времени написанной на языке UML.

Для достижения этой цели необходимо решить следующие задачи:

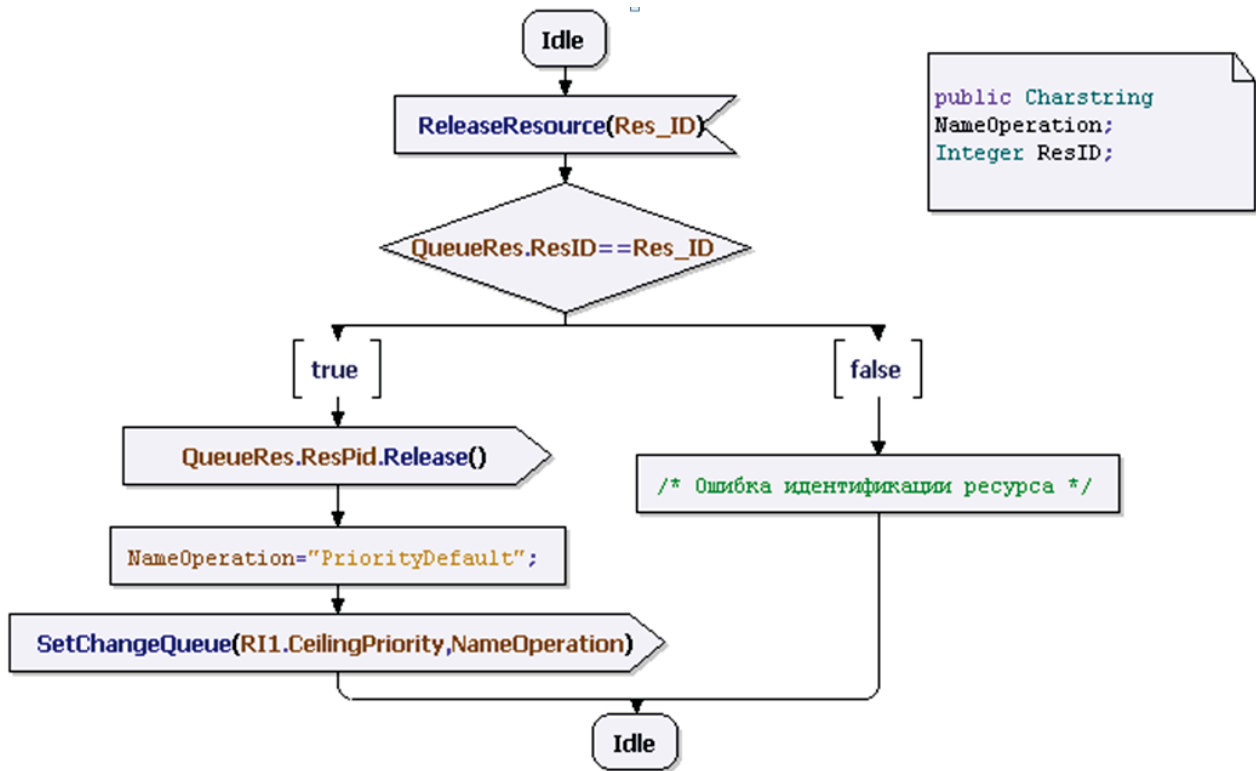
1. Создать описание имеющейся модели.
2. Провести лабораторные работы со студентами.
3. Написать методическое пособие на основании описания и результатов лабораторных работ.

Для получения начальных навыков работы с Rational TAU используется руководство от фирмы IBM которое описывает создание простой модели кофе машины. Это руководство демонстрирует основные возможности среды, которые необходимы для дальнейшей работы с моделью [3].

Для того, чтобы студенты могли работать с моделью им необходимо знать:

- элементы, входящие в состав модели;
- принцип работы каждого элемента.

На основе разработанной модели было создано руководство, которое включает в себя перечень элементов системы с особенностями их работы, характеристиками и общую архитектуру модели. В качестве примера на рис. 1 показан конечный автомат ResourceManagement, который описывает освобождение ресурса.



Пилотное исследование проводилось со студентами 3-го курса. В рамках работы студентам предлагалось выбрать один из вариантов представленных в таблице 2 и изменить систему в соответствии с предложенным вариантом.

Таблица 2. Задания по лабораторным работам

Свойства ОСВР	Вариант 1	Вариант 2	Вариант 3	Вариант 4
Тип планировщика	Плоский	POSIX	Битовый	
Алгоритм Планирования	EDF	LL	PD	RMA
Управление ресурсами	Простые семафоры	PIP	HLP	
Управление событиями	События принадлежат задаче	Системные события		

Работы проводились группами по 2-3 человека. Студентам предлагалось выбрать одно из свойств системы, которое они хотят поменять и вариант изменения. В результате выполнения заданий качество освоения студентами материала по курсу операционных систем повысилось, что подтверждается результатами экзаменов по дисциплине ОС.

На основе созданного описания и результатов выполнения лабораторных работ создается методическое пособие. Оно включает в себя:

- материалы из области ОСВР, которые будут необходимые студентам для понимания модели;
- описание модели;

- список заданий;
- инструкция по выполнению заданий;

Таким образом в рамках работы были решены следующие задачи:

1. Создано руководство по работе с моделью ОСВР.
2. Проведены лабораторные работы, которые показали что студенты качественнее усваивали материал по курсу операционных систем.

В будущем возможно расширение модели за счет обработки системных прерываний

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Э. Таненбаум. Современные операционные системы. 3-е изд. - СПб.: 2010. — 1120 с.
2. С. Золотарёв. Операционные системы реального времени для 32-разрядных микропроцессоров. Современная электроника 7/2006.
3. OSEK/VDX Operating System version 2.2.3 February 17th.
4. Гради Буч, Джеймс Рамбо, Ивар Якобсон. Язык UML. Руководство пользователя, 2-е изд. ДМК Пресс, 2007.
5. BM Rational Modeler Tau Edition 4.3 UML Tutorial.

UDC 004.272

A. Y. Kashcheev, S. A. Fyodorov, L. K. Lebedev
(SPbSTU, Federal State Unitary Enterprise "Research-Development Institute "Kvant")

GRAPHICS PROCESSING UNITS COMPARATIVE TESTING

One of the key features of modern supercomputers is their hybridity [1], i.e. the use of microprocessors with different types of cores in the compute nodes. In supercomputers, created from commercially available components, the hybridity is usually implemented by sharing universal microprocessors with superscalar architecture and graphics processing units (GPUs) in compute nodes. Modern GPUs have high computational capabilities and, as a consequence, are widely used in high performance computing (HPC). The main vendor of GPUs are NVIDIA and AMD. In 2013, a new type of co-processor Intel Xeon Phi appeared on the market, which, unlike GPUs, was built on the basis of computing cores with x86 microarchitecture.

Many works deal with the problem of various GPUs evaluation testing. Thus, in [2] the testing of different NVIDIA GPUs using SHOC suite [3] was performed. However, a specific batch of tests targeting exclusively GPUs does not allow to compare them with superscalar processors, and consequently, to determine the prospects of the use of GPUs in a specific field of application. In contrast, in [4] the comparison of the computing devices of different architectures was performed. However, the comparison was carried out on one target, albeit not very important from the point of view of the actual performance assessment - on sparse matrix multiplication by a vector.

Using the compute cores with support of x86 instructions set allowed to carry out in [5] the comparison of Intel Xeon Phi coprocessor with general purpose processor Intel Sandy Bridge using multilevel techniques of evaluation testing [6]. The adaptation of complex multilevel testing technique applied in the computer systems performance analysis for the study of GPUs was performed in this work. When testing, the ready SHOC test suite [3] were partially used and the NAS Parallel

BenchmarkS, implemented using OpenCL [7], supplemented with some of the specially prepared tests.

In accordance with the levels of a complex technique the following tests were used in the study (levels are indicated in accordance with the multi-level technique [6]):

Level 2 - tests of the boundary points of APEX-map: (GEMM, FFT, SpMV, Triad - SHOC suite)

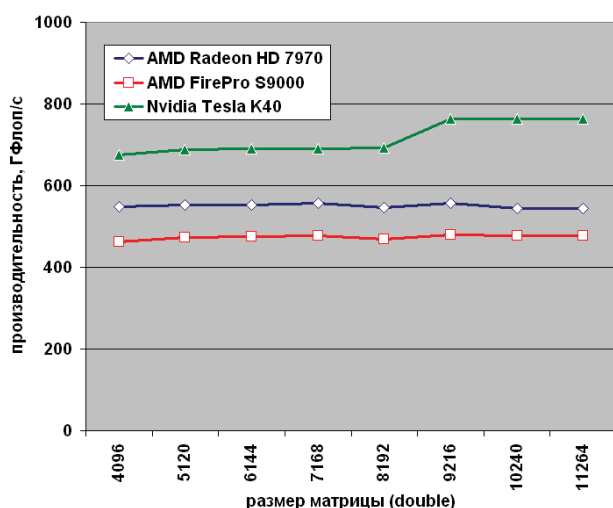
Level 3 - components special system tests: computational tests with various patterns of memory access as well as with a variable ratio of the number of arithmetic operations and memory access operations.

Level 5 - application core tests (NAS Parallel Benchmark package).

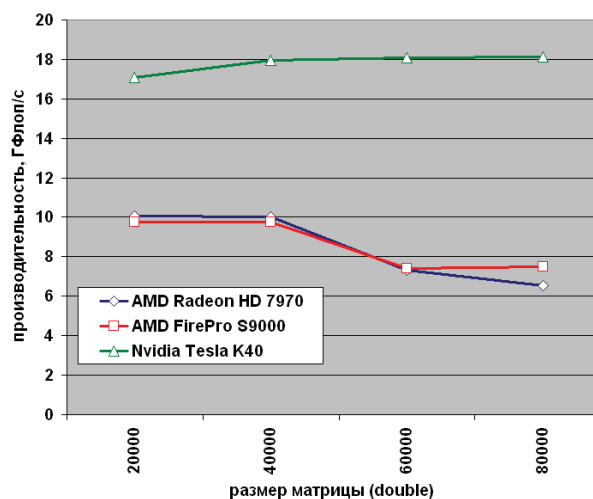
The testing of graphics cards of NVIDIA (NVIDIA Tesla K40) and AMD (FirePro S9000 and Radeon HD 7970) were performed in the framework of the studies. Performance assessment results were complemented by the results of comparing the Intel Xeon Phi coprocessor with general purpose processor Intel Sandy Bridge, given in [5].

The graphs (see Fig. 1) show the results of measuring the performance of GPUs AMD and NVIDIA on the tests of the two boundary points of APEX-map: test GEMM - multiplication of dense matrices, the point of good spatial and temporal locality, test SpMV - the sparse-matrix vector multiplication - the point of bad spatial and temporal locality. When running the SpMV test, randomly generated matrices with the number of non-zero elements equal to 1% of the number of matrix elements were used.

Test results show that the NVIDIA Tesla K40 card has better performance, and on SpMV the difference reaches two times. It should be noted that for GPUs the difference in performance between the points of good spatial and temporal locality (see Figure 1 (a)) and the point of bad spatial and temporal locality (see Figure 1 (b)) is 40-50 times.



(a) performance, GEMM, Gflop/s



(b) performance, SpMV, Gflop/s

Fig. 1. The results of performance assessment using tests GEMM and SpMV for graphics accelerators

The graph (see Fig. 2) shows the results of measuring the performance on the NAS Parallel benchmarkS. The results are given in relative units. The results of performance measurement of the test on a single processor Intel Xeon E5-2660 Sandy Bridge were taken as the unit of measurement.

Test results show that for tasks with the high level of execution parallelism (EP) GPUs significantly outperform the general purpose processor Sandy Bridge. The key determinant of performance for these tasks is the number of compute cores, which is confirmed by the results of measurements.

In CG test, comprising the stage of multiplying the sparse matrix by the vector, the best performance is demonstrated by the accelerator AMD FirePro S9000, which is 2 times the performance of the processor Intel Sandy Bridge. It should be noted that in general the performance gap between the computing accelerators and general-purpose microprocessor reduced to 1.4-2 times compared to the EP test.

SP test results show that Intel Xeon Phi has the best performance of all the accelerators. The presence of complex computing, finding a finite-difference solution of 3-dimensional Navier-Stokes equations for a compressible fluid or gas with an implicit algorithm leads to degradation of the achieved computing accelerators performance. The measurement results show that despite the significantly higher peak performance, actual performance achieved in all accelerators is lower than that of general purpose processor Intel Sandy Bridge.

It should be noted that while measuring performance for graphics accelerators AMD and NVIDIA the implementation of NAS Parallel BenchmarkS using OpenCL was applied and when testing Intel Xeon Phi coprocessor and general purpose processor Intel Sandy Bridge - the implementation of NAS Parallel BenchmarkS using OpenMP was applied.

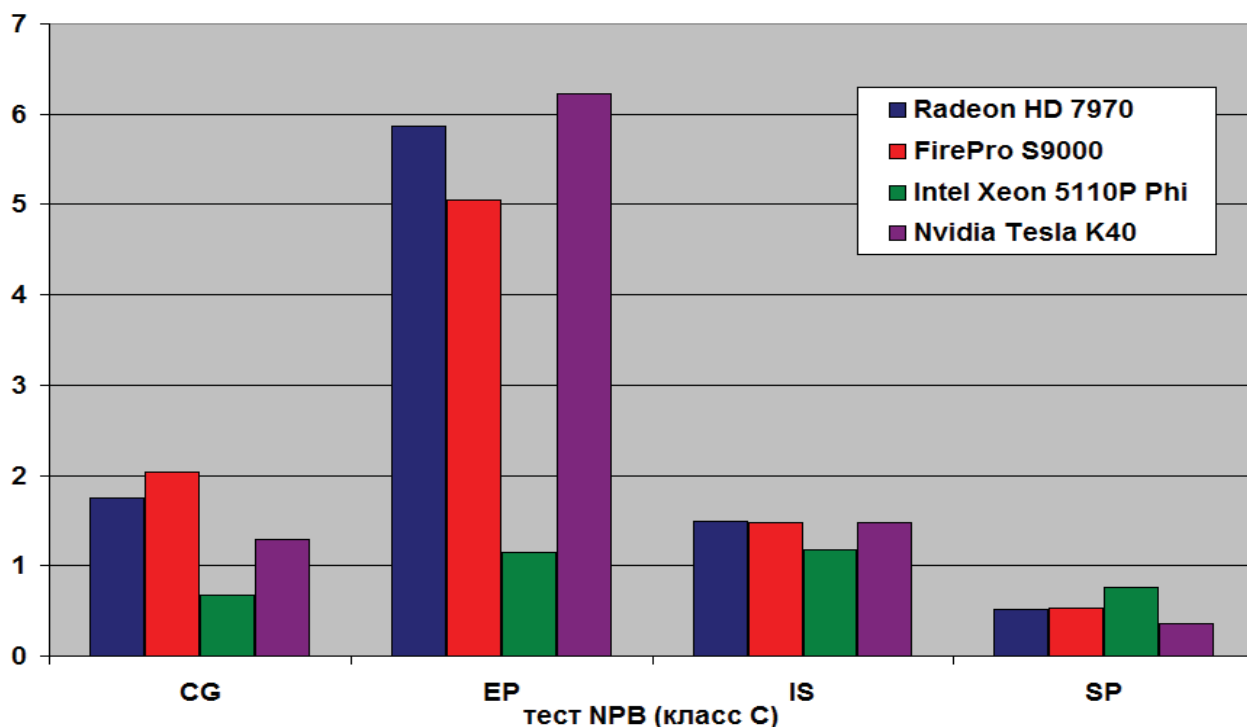


Fig. 2. Relative performance of the accelerators using tests NAS Parallel benchmark

Documents development and research results representation are used MS Office.technology

REFERENCES

1. Д. Андрюшин, В. Горбунов, Л.Эйсымонт, Перспективные особенности Tianhe-2. // Открытые системы, №8, 2013

2. Кривов М.А., Казеннов А.М. Сравнение вычислительных возможностей графических ускорителей при решении различных классов задач // Труды Всероссийской научно практической конференции "Применение гибридных высокопроизводительных вычислительных систем для решения научных и инженерных задач". Н.Н., 2011, с. 18-24.
3. Danalis et al., The Scalable Heterogeneous Computing (SHOC) Benchmark Suite // The Third Workshop on General-Purpose Computation on Graphics Processors (GPGPU 2010).
4. Saule E. et al., Performance Evaluation of Sparse Matrix Multiplication Kernels on Intel Xeon Phi. 5 Feb 2013, 19 p
5. А.В.Мищенко, С.А.Фёдоров, Д.В.Андрюшин., Оценочное тестирование Intel Xeon Phi как массово-многоядерной SMP-системы. Труды молодежной научной конференции «Студенты и молодые ученые инновационной России», СПбГПУ, Санкт-Петербург, 2013
6. В. Горбунов, Л. Эйсымонт, Комплексная методика тестирования производительности суперкомпьютеров. Профессиональный подход. // Труды Международной суперкомпьютерной конференции «Научный сервис в сети Интернет - 2013», Абрау-Дюрсо 2013.
7. http://aces.snu.ac.kr/Center_for_Manycore_Programming/SNU_NPB_Suite.html (access date 18.11.2013)

UDC 004.9

N. V. Garkusha, S. A. Fyodorov (SPbSTU)

OBJECT FORTRAN OBJECT SYSTEM FOR LANGUAGE INTEROPERABILITY

The aim of this work is the approach to automation of Fortran code interoperability and its documentation. Conventional techniques require hand-written bindings to another language. Also, bindings portability and creation of automatic documentation to them don't exist.

There is a huge body of work and a large amount of modern Fortran source code (66/77/90/95/03/08/12), which allows solving crucial applied problems. Fortran is the main language for scientific, technical and high performance computing. One of the examples of open-ended libraries of numerical analysis are LAPACK, Scalable LAPACK, and Parallel LAPACK, which are characterized by high reliability and performance, allowing to compute on high-performance systems of different architectures.

Language interoperability is becoming increasingly important in using the Fortran libraries in other industrial processes for software development. For example, unlike Fortran, Java platform is portable at the level of interpreted code, supports rich graphical user interface, and enterprise services. Through cross-language interaction it is possible to develop software in which, for example, graphical interface and enterprise interaction are implemented using Java, and basic computing – using Fortran. In other words, you can create a Java interface for the Fortran program. Consequently, there is a need to be able to call the methods implemented on Fortran from Java code.

The techniques on calling third-party code already exist in Java technology:

1. Java Native Interface (JNI), calling the code through C [1];
2. Java Native Architecture (JNA) [2].

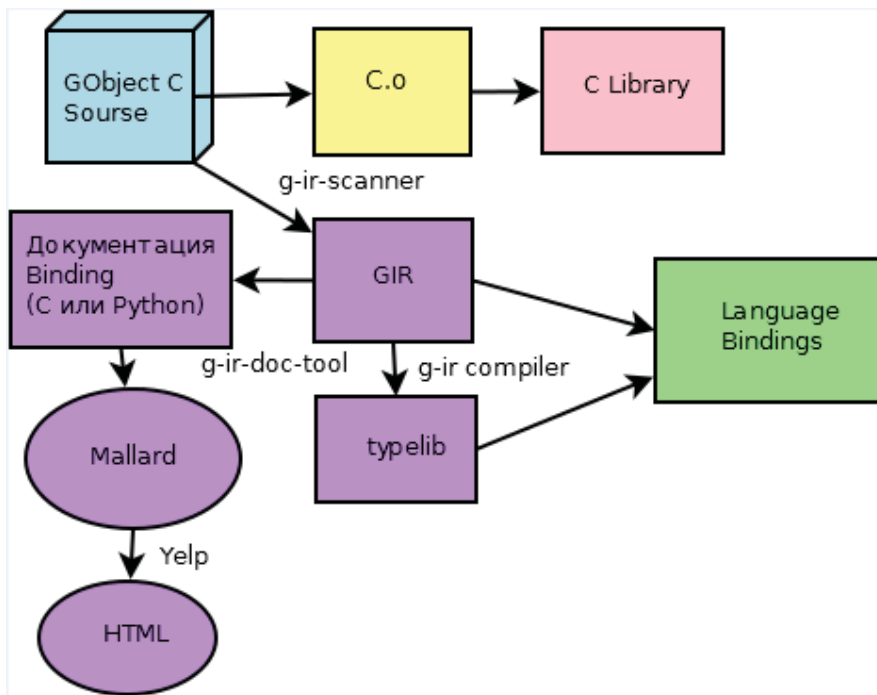


Fig. 1. The existing techniques of language interoperability Gobject
(violet entities are generated automatically)

Such technologies, unfortunately, require manual coding or bindings, and semi-automatically produce the code through which as through an intermediary the Fortran code is called and it has its deficiencies, reducing the effectiveness of working with it.

In this work two methods of the necessary automation are suggested based on the GObject object system [3, 4]. This object system was created for the industrial development of system software on C using object-oriented programming, memory management and other modern programming means.

To automate language interoperability of GObject C code a special method of GObject Introspection [5] was developed on the basis of GIR repository introspection (GObject Introspection Repository) [6]. It involves introspection of the GObject C code and basing on it, building this code interface description in the GIR XML format or binary format typelib [9]. Later, special tools can process these files statically at compile time or dynamically and then automatically generate bindings to specific languages, as well as documentation to bindings for these languages (Fig. 1).

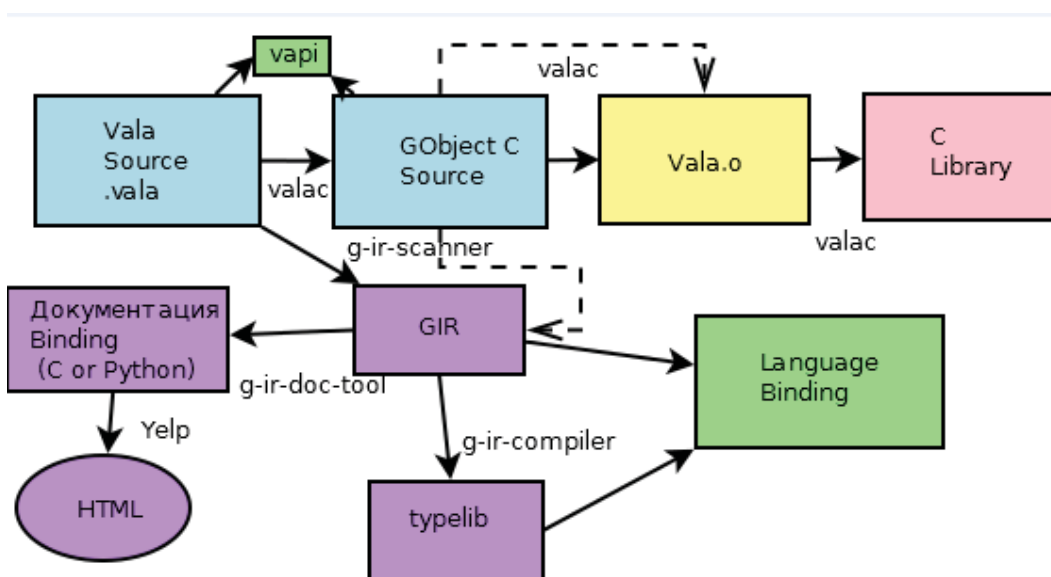


Fig.2. The existing techniques of language interoperability Vala

(violet entities are generated automatically)

One of the drawbacks GObject object system is the increase of the service code, providing the work with objects in C. To solve this problem language Vala [11] is developed, which was originally built on the object system GObject and library GLib and is translated into GObject C code, keeping all its advantages of being cross-platform and portable. Also, for Vala language the support of introspection is being developed both for the automation of language interoperability, and for the automation of code documentation [7, 8, 10].

The main advantage of the Vala language is the original objects support, as well as all tools of GLib library (lists, trees, hash tables, etc.) (Fig. 2).

In this work two techniques for the organization of language interoperability GObject Fortran and Vala-Fortran are suggested. Implementation schemes are presented below. The techniques suggested in the work contain the inclusion of Fortran characters at compile time to the object system and the GObject C code namespace (GObject C-Fortran) or Vala code (Vala-Fortran). It's not Fortran code wrapping with special intermediate code, but writing the Fortran methods as symbols that implement GObject conventions. As a result such binary code looks as if it was written in GObject C with C calling conventions. After such an inclusion for this Fortran code already developed techniques for language interoperability automation of GObject C code and Vala become available.

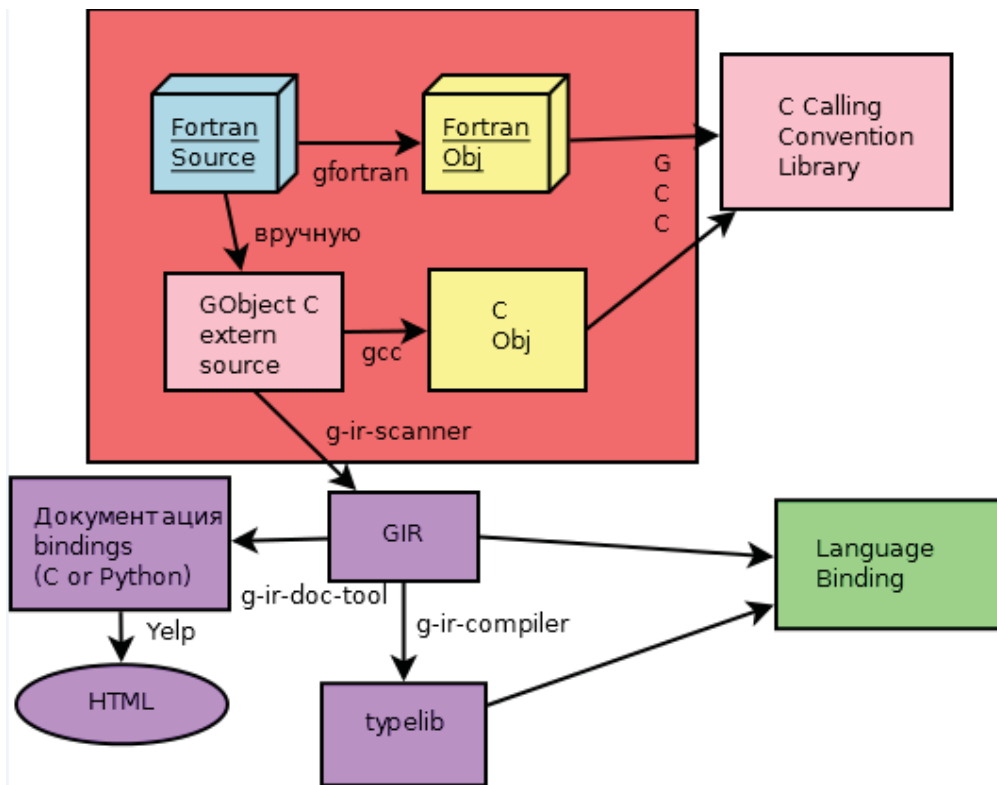


Fig.3. Gobject Fortran developed technique

The main difference in these techniques is the language in which the external symbols are declared. In the case of GObject Fortran interaction it was necessary to describe the techniques on C using the GObject structures. It was necessary not only to describe the functions called, but also to create special objects to them (Fig. 3).

In the case of Vala-Fortran interaction the code was written in the Vala language. Here it was only necessary to describe the called functions (Fig. 4).

The presence of such techniques in the industry will allow to develop new high-tech applications requiring the use of various languages and technologies, and to use modern code and optimizations of Fortran 20xx for effective support of multi-core architectures without using an intermediate code.

Documents development and research results representation are used MS Office.technology

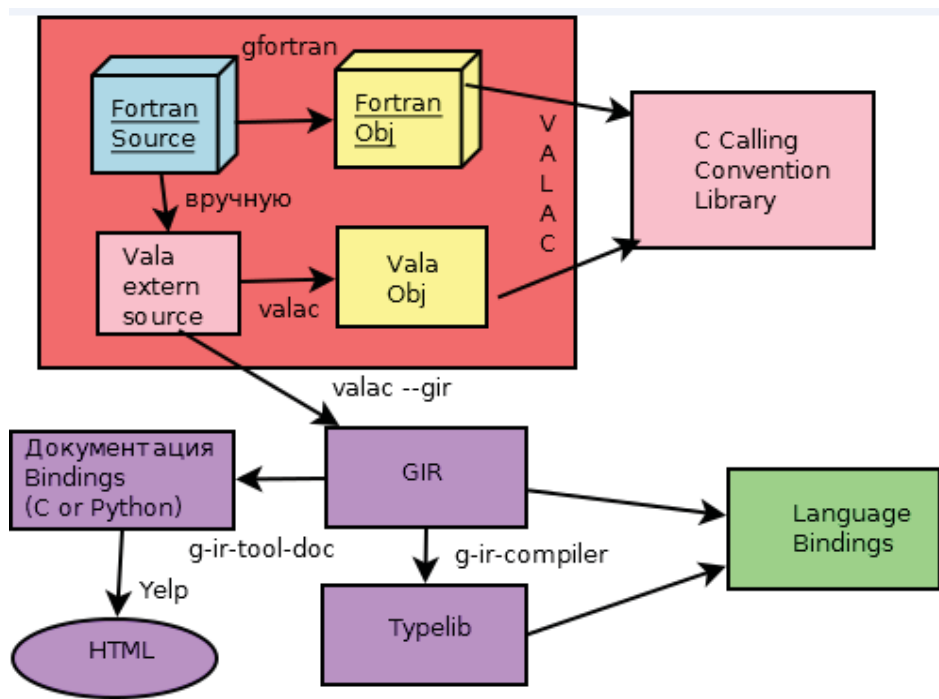


Fig. 4. Developed technique of Vala-Fortran interaction(violet entities are generated automatically)

REFERENCES:

1. Oracle JNI - <http://docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/jniTOC.html> .
2. JNA API doc - <http://twall.github.io/jna/3.5.2/javadoc/> .
3. GObject reference manual - <https://developer.gnome.org/gobject/stable/>
4. Glib reference manual - <https://developer.gnome.org/glib/stable/>
5. GObject Introspection Reference Manual - <https://developer.gnome.org/gi/1.36/>
6. GIR description - <https://live.gnome.org/GObjectIntrospection>
7. Description of the Vala introspection techniques - <https://wiki.gnome.org/Vala/Bindings>
8. Valadoc official site - valadoc.org
9. GObject Introspection overview - <http://static.antono.info/lvee/2012/gobject-introspection/gobject-introspectionslides.html?full#languages>
10. Description of the techniques to call the external libraries in Vala - <https://github.com/antono/vala-object>
11. Description of Vala syntax - <https://wiki.gnome.org/Vala> .

РАЗРАБОТКА ТЕХНОЛОГИИ ПЛАНИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ ПРИ ОРГАНИЗАЦИИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ КОНВЕЙЕРНО-ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ОБЛАЧНЫХ СРЕДАХ

В работе получены новые знания о диспетчеризации вычислительных процессов на основе текущего состояния использования вычислительных ресурсов и типов процессов при организации высокопроизводительных вычислений в облачных средах с использованием конвейерно-параллельного механизма.

Конвейерно-параллельные вычисления – одна из современных технологий построения высокопроизводительных систем, в которой выполнение операций на параллельных вычислительных узлах представляется в виде исполнения последовательностей операционных подкоманд [1]. Преимущества, достигаемые за счет использования подхода, обеспечивают увеличение коэффициента полезного использования вычислительных мощностей и повышение энергоэффективности. Основная задача работы - создание методов организации конвейерно-параллельных вычислений на платформе облачных вычислений, а также программного комплекса управления вычислительной нагрузкой, внедрение которого в реальную облачную систему гарантирует улучшение ее эксплуатационных характеристик, таких как эффективность распределения ресурсов, совокупная производительность виртуального кластера, энергопотребление.

В ходе работы проведены масштабные аналитические исследования, в рамках которых рассмотрены пути создания алгоритмов планирования вычислительных процессов в облачной среде, архитектура систем распределенных облачных вычислений. Решение задачи эффективного использования ресурсов высокопроизводительных систем при выполнении прикладных процессов решается на уровне планировщика процессов. Известны решения данной задачи в грид-системах. Компонентом планирования процессов в грид-системах является менеджер ресурсов, используемый для перераспределения пользовательских задач между узлами вычислительной сети Globus Toolkit, gLite и пр. Однако данный менеджер не учитывает условия работы, характерные для облачных систем (наличие виртуальных сред, разделение физических ресурсов между виртуальными, динамическая миграция виртуальных машин и т.п.). В этой связи принципы планирования, используемые в грид-системах, являются бинарными (выбирается для загрузки физический узел сети, свободный в данный момент). Кроме того, менеджерами ресурсов грид-систем не учитывается текущее состояние распределения ресурсов (процессорного времени, объемов памяти и нагруженности подсистемы ввода/вывода) данного узла, которое необходимо учитывать при распараллеливании выполнения вычислительных процессов. Другим решением задачи планирования является подход, используемый в подсистемах диспетчеризации процессов в суперкомпьютерах (например, Hadoop [2], Dryad [3], MapReduce [4] и пр.). Однако данные суперкомпьютерные технологии не учитывают особенностей функционирования виртуальных машин в облачных системах, в частности они не обеспечивают консолидацию виртуальных вычислительных сред и перераспределение нагрузки с учетом возможной конвейеризации процессов. Стандартные планировщики облачных платформ используют алгоритмы FIFO, RoundRobin и т.п., в которых не учитывается, в частности, динамика распределения нагрузки, многомерность пространства параметров распределения

вычислительных ресурсов и др. Проблему отсутствия планировщика процессов, работающего в условиях облачной среды и учитывающего текущее распределение ресурсов при планировании конвейерно-параллельных процессов, предлагается устранить в ходе выполнения работы.

Подход, основанный на конвейерно-параллельных вычислениях, подразумевает выполнение операций на вычислителях в виде исполнения последовательности составляющих операцию подкоманд. При вычислениях на таких устройствах одновременно несколько различных элементов данных или несколько процессов могут находиться на разных стадиях обработки [5]. Применение данной технологии к облачным вычислительным системам обеспечивает значительное технологическое преимущество за счет агрегации преимуществ виртуализации и гибкого планирования. Теоретическая база, представленная существующими работами в данной области, предоставляет достаточно данных для разработки эффективной модели для выполнения задач на вычислительных кластерах в облачной среде.

В ходе теоретических исследований разработаны модели и алгоритмы планирования вычислительных процессов и распределения виртуальных машин, обеспечивающие динамическое распределение нагрузки на базе применения принципов конвейерно-параллельной обработки в облачной среде. С целью повышения производительности предлагаемая модель осуществляет обработку процесса на этапе компиляции: множество набора команд разделяется на независимые подмножества, которые предназначены для параллельной обработки. Внутри каждого подмножества выполняется разбиение на инструкции для последующей конвейерной обработки. Для каждого из подмножеств производится подсчет типов команд, предъявляющих особые требования к процессорному времени, оперативной памяти или пропускной способности ввода-вывода.

Разработанный алгоритм планирования позволяет распределять команды по виртуальным машинам в облачной среде, тогда как распределение виртуальных машин по физическим серверам обеспечивается за счет эксплуатации технологии миграции виртуальных машин в облачной среде. Виртуальные машины и физические сервера образуют два множества. Каждая виртуальная машина и каждый физический сервер характеризуются параметрами использования ресурсов (процессорное время, объем оперативной памяти, пропускная способность ввода-вывода). Предложенная модель предусматривает определение оптимальных значений, которые делают возможным функционирование виртуальной машины из множества виртуальных машин на физическом сервере из множества физических серверов.

Разработан планировщик вычислительных процессов с интеллектуальными функциями, ориентированный на применении в облачной среде. Такой планировщик обладает значительными преимуществами в сравнении с существующими аналогами, в том числе, обеспечивает повышенный уровень автоматизации, поскольку основан на динамическом мониторинге нагрузки и на ее балансировке между узлами виртуальной среды облака, а не управляется пользователем посредством задания параметров в конфигурационных файлах запуска вычислительных задач. Планировщик выполняет автоматическое распределение ресурсов вычислительной системы, учитывая при этом многомерность пространства факторов, соотношение процессорного времени, объема оперативной памяти, пропускной способности подсистемы ввода-вывода, необходимых виртуальной машине, с ресурсами, предоставляемыми физическими серверами. В результате, такой подход позволяет повысить энергоэффективность эксплуатируемого оборудования, а оптимально распределять нагрузку на виртуальные машины облачной среды.

На этапе экспериментальных исследований полученных результатов выполнена программная реализация разработанных алгоритмов для платформы OpenStack. Эксперименты показали, что благодаря оптимальному распределению виртуальных машин по физическим серверам облачной среды решается задача повышения не только производительности обработки информации свыше 10%, но и энергоэффективности до 10%, что позволяет увеличить экономическую эффективность высокопроизводительных вычислений.

Предлагаемое решение позволяет оптимизировать распределение ресурсов по нескольким критериям (например, нагрузок на дисковую подсистему, процессор и память), вести оперативный учет фактически потребляемых ресурсов виртуальными машинами за счет опроса гипервизоров облачной системы, а также позволяет настраивать планировщик для приложений разных классов, запускаемых в виртуальных машинах, а не ориентироваться только на прикладные задачи одного типа.

Внедрение вычислительной концепции конвейерно-параллельных вычислений в облачную среду позволило создать новый механизм распределения ресурсов с повышенной гибкостью, оперативно реагирующей на изменения в системе и перераспределяющей ресурсы. Модель вычислений на базе тандема конвейеризации и распараллеливания задач обеспечивает повышенную динамичность решения задачи распределения нагрузки

Результаты НИР применимы в следующих областях:

- исследование и разработка сетевой инфраструктуры облачных сред, предназначенных для организации конвейерно-параллельных вычислений;
- исследование и разработка систем управления конвейерно-параллельными вычислениями в облачных средах;
- исследование и разработка методов и средств управления сетевой инфраструктурой облачных сред, используемых для организации конвейерно-параллельных вычислений.

Проведенные исследования позволяют расширить возможности по обеспечению управления ресурсами облачных сред, а также возможности использования технологии конвейерно-параллельных вычислений на базе виртуализированного вычислительного кластера. Внедрение конвейерно-параллельных вычислений в облачную среду позволяет решить ряд проблем, возникающих при организации крупномасштабных кластерных систем, главными из которых являются частые простои и неэффективное использование ресурсов физических серверов.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. А. А. Талалаев. Организация конвейерно-параллельных вычислений для обработки потоков данных // А. А. Талалаев // Журнал «Информационные технологии и вычислительные системы». – Электрон. журн. – 2011. Режим доступа: http://www.jitcs.ru/images/documents/2011-01/8_13.pdf, свободный.
2. Спецификация: Apache Hadoop. Режим доступа: <http://hadoop.apache.org>, свободный. Дата обращения: 07.10.2013
3. Microsoft Dryad // Microsoft, Inc. – Электрон.дан. – 2007. Режим доступа: <http://research.microsoft.com/research/sv/dryad>, свободный. – Загл. с экрана. Дата обращения: 07.10.2013
4. J. Dean, S. Ghemawat. MapReduce: simplified data processing on large clusters / J. Dean, S. Ghemawat // Google, Inc. – Электрон. дан. – 2008. Режим доступа: <http://research.google.com/archive/>

mapreduce.html, свободный. Дата обращения: 04.10.2013

5. В. П. Гергель. Основы параллельных вычислений для многопроцессорных вычислительных систем.
– Н. Новгород: Изд-во ННГУ им. Н. И. Лобачевского, 2010. 420 с.

УДК 004.932.2

М.В. Виденева (асп., кафедра Информатики, СПбГУ),
И.П.Соловьев, к. ф.-м.н., доцент.

СРАВНЕНИЕ ФРАКТАЛЬНЫХ ХАРАКТЕРИСТИК ИЗОБРАЖЕНИЙ И ИХ ВЕЙВЛЕТ-ОБРАЗОВ В ЗАДАЧЕ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

Задача анализа и классификации цифровых изображений приводит нас к необходимости выбора подходящего метода исследования из целого ряда таких методов, ориентированных на специфические области применения и опирающихся на различные математические и компьютерные модели. Так, для анализа текстовых изображений, часто применяют морфологические методы, для анализа изображений биомедицинских препаратов — статистические текстурные [6, 9].

Мы рассматриваем два подхода к анализу и классификации изображений на основе фрактальных характеристик этих изображений. Оба подхода основаны на вычислении фрактальной сигнатуры монохромных изображений, представляемых двумерной функцией градации серого [2].

Первый подход опирается на непосредственное получение фрактальных характеристик исходного изображения, в нашем случае это метод фрактальной сигнатуры (МФС), второй основан на применении МФС к вейвлет-преобразованиям исходных изображений. Метод фрактальной сигнатуры заключается в вычислении площади поверхности так называемого δ -параллельного тела для поверхности графика функции градации серого (G). Под δ -параллельным телом понимают множество точек, находящихся на расстоянии не более чем δ от поверхности графика функции G ; такие точки образуют «покрывало» толщиной 2δ . Вычисление объема «покрывала» позволяет нам получить приближенное значение площади поверхности. Отношение $\log_2 A_\delta / \log_2 \delta$ называется фрактальной сигатурой, где A_δ — площадь поверхности графика функции G . Путем последовательного изменения δ , начиная с 1, вычисляется вектор фрактальных сигатур — характеристику изображения [5,7].

Принципиальным отличием второго подхода является то, что он проводится в два этапа. Сначала выполняется вейлет-преобразование исходного изображения [1]. На втором этапе вычисляются фрактальные характеристики полученного образа с помощью МФС [3]. Отметим, что наиболее распространенные вещественные базисы вейвлетов конструируются на основе производных функции Гаусса: $g(t) = \exp(-t^2/2)$. Это обусловлено тем обстоятельством, что функция Гаусса и ее производные имеют хорошие показатели локализации, как во временной, так и в частотной областях [4]. Таким образом, вейвлет, построенный на основе производных функции Гаусса, имеет узкий энергетический спектр, что

делает его наиболее предпочтительным при анализе особенностей высокого порядка. В данном случае мы применяем к исходному цифровому изображению вейвлет-преобразование, построенное с помощью второй производной функции Гаусса. Полученный вейвлет-образ мы считаем новым изображением, для которого вычисляем значение фрактальной сигнатуры с помощью МФС для $\delta = 1, 2$. Последовательно изменяя значение параметра масштаба в эмпирически определенном диапазоне при вейвлет-преобразовании для каждого изображения, мы получаем набор связанных с ним образов. Фрактальные сигнатуры, вычисленные для каждого из таких образов, образуют характеристический вектор исходного цифрового изображения.

Таким образом, каждому исходному изображению сопоставляется два вектора фрактальных сигнатур. Эмпирическим путем было выбрано, что в первом подходе δ разумно изменять в диапазоне от 1 до 50. Соответственно, масштаб при вейвлет-преобразовании выбирается в сопоставимом диапазоне. Вычисление декартова расстояния между векторами изображений, полученных с помощью одного и того же подхода, позволяет разделять исходное множество на классы с некоторой погрешностью: чем меньше расстояние между векторами, тем больше вероятность принадлежности изображений одному классу [8].

Мы сравниваем результаты применения описанных подходов для множества изображений, заранее разбитого на несколько классов — «обучающей выборке». Также мы проводим анализ полученных результатов и их зависимость от структуры изображения и применяемых способов приведения цветных изображений к полутоновой гамме.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. Астафьева Н. М. Вейвлет-анализ: основы теории и примеры применения. Успехи физических наук. 1996. Т. 166, №11. С.1145-1170.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Техносфера. Мир цифровой обработки, 2006 г.
3. Павлов А. Н., Анищенко В. С. Мультифрактальный анализ сигналов на основе вейвлет-преобразования. Саратовский государственный университет. 2007.
4. Чуи Ч. Введение в вейвлеты. М. Мир, 2001.
5. Falconer K.J. Fractal Geometry. Mathematical Foundations and Applications. — John Wiley & Sons, 1990.
6. N. Ampilova, E. Gurevich, I. Soloviev. Texture recognition as a quantitative analysis of underdoses in pharmacology and medicine. Proc. 5 Int. Conf. CEMA10, 7-10 Oct. 2010, Athens, Greece. p.78-81.
7. N. Ampilova, E. Gurevich, I. Soloviev. Application of Modified Fractal Signature & Regny Spectrum Methods to the Analysis of Biomedical Preparations Images. Proc. 6 Int. Conf. CEMA11, 6-8 Oct. 2011, Sofia, Bulgaria. p. 96-100
8. Soloviev, N. Ampilova, E. Gurevich. On implementation of a neural network classifier for some classes of biological and medical preparation images. Proc. 7 Int. Conf. CEMA12, 8-10 Nov. 2012, Athens, Greece. P. 94-97.
9. Tang Y. Y., Hong Ma, Dihua Xi, Xiaogang Mao, C. Y. Suen. Modified Fractal Signature (MFS): A New Approach to Document Analysis for Automatic Knowledge Acquisition. — IEEE Trans. Knowledge and Data Eng., vol.9. no. 5, 1997, pp. 742-762.

РЕФАЛ-5Е: РАЗРАБОТКА ЯЗЫКА И РЕАЛИЗАЦИЯ ТРАНСЛЯТОРА.

Язык рефал (РЕкурсивных Функций АЛгоритмический) — функциональный язык программирования, ориентированный на «символьные преобразования»: обработку термов, то есть планарно представленных древовидных структур из символьных строк (например, осуществление алгебраических преобразований); перевод с одного языка (искусственного или естественного) на другой; решения задач, связанных с искусственным интеллектом. Отличительной чертой языка является использование сопоставления с параметрическим образцом и возможность переписывания термов в другие термы как основного способа определения функций. При этом переписывание термов осуществляется посредством замены простейших функциональных термов на произвольные термы [1]. На данный момент наиболее известны следующие диалекты языка рефал: базисный рефал, рефал-2 [3], рефал-5 [5], рефал-6 [4] и рефал+ [2]. Рефал+ является наиболее развитым из них, но в то же время рефал-5 имеет более простой синтаксис и предоставляет достаточно средств для эффективного решения большинства задач программирования.

В современном мире необходимы языки, поддерживающие эффективные технологии. Однако совершенствование вышеперечисленных диалектов языка рефал на данный момент не ведется. Как следствие ни в одном из них нет поддержки новых современных эффективных технологий, присутствующих в современных компьютерах.

Целью данной работы является разработка современного диалекта языка программирования рефал — рефал-5Е. Диалект рефал-5е основан на диалекте рефал-5, но добавляет такие возможности, как интерпретация рефал-5е кода при помощи встроенного интерпретатора, расширенные возможности импорта модулей, импорт функций из динамически загружаемых библиотек. Транслятор языка позволяет выполнять программы на языке рефал-5е в несколько потоков без особых действий со стороны разработчика: язык рефал был разработан таким образом, что почти все функции являются чистыми, то есть на их результат влияет только аргументы, переданные в нее, так же большинство функций не производят побочных действий.

На данный момент полностью разработан язык и реализован транслятор, транслятор проходит тестовую эксплуатацию. При разработке транслятора языка программирования рефал-5е использовалась операционная система Microsoft Windows 7 и Microsoft Visual C++.

ЛИТЕРАТУРА

1. Бабаев И.О., Герасимов М.А., Косовский Н.К., Соловьев И.П. Интеллектуальное программирование. Турбо Пролог и Рефал-5 на персональных компьютерах. - Л.: Издательство ЛГУ, 1992.
2. Гурин Р.Ф, Романенко С.А. Язык программирования Рефал Плюс. - М.: ИНТЕРЕХ, 1991. - 183 с.
3. Климов А.В, Романенко С.А. Система программирования Рефал-2 для ЕС ЭВМ. Описание библиотеки функций. М.:ИПМ им.М.В.Келдыша ФН СССР, 1986, препринт N 200. - 38 с.

4. Климов А.В. - Программирование на языке Рефал. 2004. <http://refal.net/~arklimov/refal6/manual.htm> [дата просмотра: 27.02.2014].
5. Turchin V.F. Refal-5. Programming Guide and Reference Manual. — New England Publishing Co., Holyoke, 1989.

УДК 621.320

А.Ю. Жигулин (асп., каф. информатики, СПбГУ)
В. О.Сафонов, д.т.н., проф., каф. информатики, мат.-мех. фак-т, СПбГУ

АЛГОРИТМ ШПИОНАЖА ДЛЯ КОМПЬЮТЕРНЫХ ИГР С МНОГОЧИСЛЕННЫМИ СОПЕРНИКАМИ

Целью работы является представление накопленных компьютерными соперниками знаний о действиях пользователя в виде ранжируемых и легко определяемых данных, последующее ранжирование и группировка полученных данных с последующим моделированием процесса обмена накопленных данных между компьютерными соперниками таким образом, чтобы пользователь не мог предсказать дальнейшее поведение компьютерных соперников, а компьютерные соперники имели возможность предсказывать действия пользователя по «украденным» знаниям, полученным опытным путем другими компьютерными соперниками.

При создании компьютерных игр одной из главных проблем является быстрое привыкание пользователя к игровому процессу. После привыкания пользователь чаще всего не видит смысла в дальнейшей игре, и решением данной проблемы обычно являются два метода оптимизации игры:

- добавление нового контента в игру, с новой игровой механикой
- изменение механики уже существующего контента.

Оба этих метода требуют денежных и человеческих затрат, так как требуется придумывать изменения «вручную». Чтобы обойти ненужные затраты, наиболее эффективное решение – это создание динамической системы, которая меняется со временем, создавая непредсказуемую для пользователя механику, не добавляя при этом нововведений. В таком случае разработчики должны будут изначально заложить алгоритм изменения игры, после чего она будет подстраиваться под пользователя сама.

Особенности реализации системы:

Система разделена на несколько частей:

- интерфейс, определяющий основные возможные действия пользователя, а также ранжированные сущности, определяющие все возможные действия (организация данного интерфейса и сущностей, ему удовлетворяющих, требует 90% всего времени на интегрирование алгоритма шпионажа в игру и будет уникальна в каждой игре)
- функциональность, отвечающая за запись действий пользователя

- стадия шпионажа между компьютерными соперниками, в которой они с определенной долей вероятности, зависящей от успешности в игре против пользователя и количества полученных знаний, забирают определенные знания либо случайные знания
- дальнейшее применение своих и украденных знаний против пользователя
- (опциональное) уничтожение неактуальных стратегий.

Суть работы системы:

Когда пользователь играет с компьютерными соперниками, каждый из них записывает все его действия (например, нажатия на клавиши, стратегию поведения, возможные остановки или ускорение действий), после чего проверяет среди своих знаний возможные противодействия, либо действует по написанному разработчиком алгоритму в случае, если подходящей по алгоритму шпионажа стратегии не нашлось. На стадии шпионажа (в процессе игры, либо в режиме хода соперников) каждый из компьютерных соперников ищет оптимальные стратегии и «крадет» их. Далее алгоритм повторяется.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 004

А.К. Кретов (асп., каф. информатики, СПбГУ)
В. О.Сафонов, д.т.н., проф., каф. информатики, мат.-мех. фак-т, СПбГУ

СИСТЕМА АВТОМАТИЧЕСКОГО ВСТРАИВАНИЯ СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ В БИНАРНЫЕ ФАЙЛЫ ПРИЛОЖЕНИЙ ДЛЯ ПЛАТФОРМЫ iOS

Целью работы является исследование подходов к модификации бинарных файлов приложений для мобильной платформы iOS, а также проектирование и реализация системы автоматического встраивания средств криптографической защиты в готовые приложения. Кроме того, в рамках работы планируется реализовать переиспользуемые модули шифрования данных для платформы iOS с использованием зарубежных (SHA, RSA) и отечественных (ГОСТ 28147-89) алгоритмов.

Основной задачей, которую будет решать разрабатываемая система, является предотвращение утечек информации, которые могут происходить при активном использовании мобильных приложений, не выполняющих должную защиту хранимой информации. Для решения поставленной задачи система выполняет следующую последовательность действий:

1. Расшифровка бинарного файла приложения (все приложения на платформе iOS хранят бинарное содержимое в зашифрованном виде, и для модификации его необходимо предварительно привести в читаемый вид);
2. Определение входных точек для модификации. Единственным поддерживаемым типом входных точек является, на данный момент, запись на жесткий диск планшетного устройства. В дальнейшем планируется осуществить поддержку следующей потенциально незащищенной функциональности: работа с внутренним хранилищем песочницы модифицируемого приложения (NSUserDefaults), работа с контейнером ключей планшетного устройства (KeyChain), а также взаимодействие с удаленными ресурсами через сеть Интернет;

3. Внедрение разработанных переиспользуемых модулей шифрования в бинарный файл без изменения существующей логики работы приложения;
4. Модификация вызовов методов, направленных во входные точки из п.3, таким образом, чтобы никакие данные не передавались в незашифрованном виде во внутренние системы хранения информации и не отправлялись по открытым сетям передачи данных;
5. Шифрование модифицированного бинарного файла мобильного приложения для корректной установки на планшетное устройство;
6. Подписание приложения сертификатом для прохождения обязательной валидации во время процесса установки;
7. Установка модифицированного приложения на планшетное устройство с помощью iTunes.

Для реализации системы была выбрана операционная система Mac OS X и язык программирования Objective-C, поскольку большинство приложений для целевой платформы разрабатывается на этом языке. В дальнейшем планируется провести работы по поддержке популярных мобильных платформ (Android, Windows 8), в том числе портирование переиспользуемых модулей.

В качестве криптопровайдера для зарубежных алгоритмов используются стандартные библиотеки платформы iOS, которые являются исключительно быстрыми благодаря аппаратной оптимизации, а для реализации модуля шифрования с отечественными алгоритмами применяется коммерческая библиотека Крипто-ПРО.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 004.04

С.Н. Павлов (4 курс, каф. СП, СПбГУ)
А.Г. Петров (ООО "Биомоделирование")

ИЗУЧЕНИЕ МЕТОДОВ РЕГИСТРАЦИИ СООТВЕТСТВИЙ ДЛЯ ВЫРАВНИВАНИЯ 3D-МОДЕЛЕЙ

Целью работы является реализация одного из способов регистрации и выравнивания 3D-моделей, а также изучение основ и подходов других методов, а также некоторое сравнение. Изначально пробовалось два алгоритма по регистрации и выравниванию, был выбран тот, что заработал лучше.

Особенности:

Для реализации алгоритма и проверки их работоспособности была выбрана библиотека PCL, которая уже имеет некоторые реализации методов регистрации соответствий.

В качестве тестируемых 3D-моделей были использованы модели лица и торса из фотограмметрического подхода в трехмерной реконструкции.

Модели тела человека сильно отличаются друг от друга, на них отсутствуют резко выделяющиеся особенности, поэтому стандартные (уже реализованные) алгоритмы плохо работают.

Методы:

Point Feature Histograms (PFH) – метод, уже реализованный в библиотеке PCL. Он основан на поиске некоторых особенностей, например, нормалей поверхности.

Iterative Closest Point (ICP) – алгоритм для выравнивания моделей, который уже реализован в PCL. Идея метода заключается в том, чтоб находить ближайшие точки и приближать к ним.

4-Point Congruent Sets (4PCS) – метод из статьи «4-Points Congruent Sets for Robust Pairwise Surface Registration» под авторством Drog Aiger, Niloy J. Mitra и Daniel Cohen-Or. Моя реализация данного алгоритма выполнена с помощью библиотеки PCL, поэтому имеются существенные изменения по сравнению с авторской реализацией.

На текущий момент имеется несколько экспериментов работы методов ICP и PFH, а также первичная реализация 4PCS-алгоритма, которая дорабатывается для повышения качества поиска соответствий и выравнивания моделей.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 004.4'422

А. В. Пилипенко (асп., каф. информатики, мат.-мех. фак-т, СПбГУ)
В. О. Сафонов, д.т.н., проф., каф. информатики, мат.-мех. фак-т, СПбГУ

АНАЛИЗ ДОСТИЖИМОСТИ МЕТОДОВ С ПОМОЩЬЮ МЕЖПРОЦЕДУРНОГО АНАЛИЗА ПОТОКА ДАННЫХ

Косвенные вызовы в объектно-ориентированных языках требуют специальной обработки при анализе достижимости методов. Методы, достижимые посредством косвенных вызовов, определяются типом объекта времени исполнения, на котором осуществляется вызов. Однако во время статического анализа информация об объектах времени исполнения отсутствует. При статическом анализе для каждого косвенного вызова вычисляют консервативное приближение множества возможных типов объектов, на которых осуществляется вызов [1, 2, 4]. В докладе предложен алгоритм анализа достижимости методов, который для вычисления этого множества использует межпроцедурный анализ потока данных.

Для анализа достижимости методов достаточно анализировать операции со ссылочными типами, а именно операции создания объектов, вызовы методов, записи и чтения объектных ссылок. В процессе работы алгоритма будем строить графа потока данных, содержащий узлы следующих типов:

- узлы создания объектов (для моделирования разных точек создания объектов одного типа используется один модельный объект),
- узлы использования объектов для косвенных вызовов,
- промежуточные узлы:
 - аргументы методов,
 - возвращаемые значения методов,

- поля объектов (для моделирования разных объектов одного типа используется один модельный объект),
- элементы массивов (для моделирования массивов используется один модельный объект),
- узлы слияния контекстов исполнения (ф-узлы).

Данный граф не содержит узлов для локальных переменных, поток данных между ними моделируется с помощью абстрактной интерпретации. Ребра в графе отражают операции присваивания в программе. Направленное ребро из узла А в В означает, что существует такая последовательность локальных присваиваний, результате исполнения которой, значение узла А присваивается узлу В или используется узлом В.

Множеством достигающих типов будем называть типы узлов создания объектов, достижимых из данного узла по обращениям ребер. Для оценки возможных типов объектов времени исполнения, будем вычислять множество достигающих типов для каждого узла.

Граф потока данных строится с помощью итеративного алгоритма. Алгоритм используют две очереди: очередь методов и очередь узлов. Очередь методов содержит достижимые методы, код которых еще не был проанализирован. Очередь узлов содержит узлы, у которых есть зависимости, для которых еще не вычислены множества достигающих типов. В начале работы алгоритма множество методов содержит точки входа приложения, множество узлов пусто. На каждой итерации алгоритма выполняются следующие действия.

- Обрабатывается очередь методов. Код каждого метода анализируется, граф потока данных дополняется узлами и ребрами, соответствующими операциям в коде метода. В очередь узлов добавляются новые узлы создания объектов и существующие узлы, у которых появились новые зависимости. Методы достижимые посредством статических вызовов добавляются в очередь методов.
- Обрабатывается очередь узлов. С помощью обхода в ширину вычисляются типы объектов, которые достигают добавленные на предыдущем шаге узлы. Для каждого узла в очереди выполняются следующие действия:
 - Дополняются достигающие типы всех узлов, зависящих от текущего узла. Узлы, у которых изменилось множество достигающих типов, добавляются в очередь узлов.
 - Если узел является вызовом метода, вычисляется множество достижимых из него методов. В очередь методов добавляются достижимые методы, которые еще не были проанализированы.

Алгоритм завершается по достижению неподвижной точки.

Алгоритм расширяет граф потока данных по мере обнаружения достижимых методов. Операции со ссылочными типами в коде методов анализируются с помощью абстрактной интерпретации. Абстрактный интерпретатор оперирует узлами графа вместо значений времени исполнения. Контекст интерпретатора представляет собой состояние локальных переменных. Операции присвоения в пределах контекста интерпретируются присваиванием значений в контексте. Операции создания объектов, вызовы методов, записи и чтения объектных ссылок создают соответствующие узлы и ребра.

Предложенный подход более точен по сравнению с существующими алгоритмами, которые анализируют статические типы. Алгоритм может быть применен для анализа достижимости методов в языках с динамической типизацией. Граф потока данных, построенный в ходе работы алгоритма, может быть использован для дальнейших оптимизаций программы.

В статье [3] предложен механизм компиляции .NET приложений для встроенных систем. Приложения компилируются в закрытой модели, что позволяет удалять недостижимые методы, поля и классы. Для анализа достижимости методов авторами используется алгоритм Rapid Type Analysis [1]. Описанный в докладе алгоритм может быть использован для уточнения анализа, что позволит сократить итоговый размер приложений.

Документирование и представление результатов осуществлены в технологии MS Office.

ЛИТЕРАТУРА

1. Bacon D., Sweeney P. Fast static analysis of C++ virtual function calls. // ACM SIGPLAN Notices. Vol. 31. Issue 10. 1996. P. 324–341.
2. Dean J., Grove D., Chambers C. Optimization of Object-Oriented Programs Using Static Class Hierarchy Analysis. // Proceedings of the 9th European Conference on Object-Oriented Programming (ECOOP'95). 1995. P. 77–101.
3. Sallenave O., Ducournau R. Efficient Compilation of .NET Programs for Embedded Systems. // Proceedings of the Workshop on the Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS '10). 2010. P. 1–8.
4. Tip F., Palsberg J. Scalable Propagation-Based Call Graph Construction Algorithms. // ACM SIGPLAN Notices. Vol. 35. Issue 10. October 2000. P. 281–293.

УДК 004.052.3

А. В. Игумнов, С. Э. Сараджишвили (СПбГПУ)

СОХРАНЕНИЕ РАБОТОСПОСОБНОСТИ МНОГОАГЕНТНЫХ СИСТЕМ

Использование многоагентного подхода [1] при разработке систем промышленного уровня ставит задачи исследования отказоустойчивости многоагентных систем (МАС) и разработки методов обнаружения отказов и восстановления работоспособности. В качестве основного современного средства обеспечения отказоустойчивости авторы работы [2] выделяют использование избыточности. Современные методики обеспечения отказоустойчивости МАС основаны на введении избыточности некоторых ее компонентов: или отдельных агентов (DarX [3], Brokered MAS [4]), или агентов и отдельных задач системы [5]. Большинство современных систем промышленного уровня реализованы как программно-аппаратные комплексы, в т.ч. распределенные. Однако, существующие методики обеспечения отказоустойчивости МАС при определении стратегии резервирования, а также техник обнаружения отказов и восстановления работоспособности используют определение многоагентной системы как системы, состоящей из взаимодействующих агентов [6], характерное для теории искусственного интеллекта. Таким образом, существующие подходы фактически обеспечивают устойчивость только агентной модели, а не всей программно-аппаратной МАС.

Цель настоящей работы – обеспечить сохранение работоспособности распределенной программно-аппаратной многоагентной системы при отказах отдельных ее компонентов. Авторами были сформулированы и решены следующие задачи:

- Разработана стратегия резервирования распределенной программно-аппаратной МАС;
- Разработана методика восстановления работоспособности распределенной программно-аппаратной МАС после обнаружения отказов отдельных ее компонентов.

В работе [5] для достижения отказоустойчивости было предложено использовать избыточность задач и агентов МАС, поэтому будем считать, что задача, выполняемая агентом, является элементарным функциональным элементом МАС. Тогда будем считать, что агент является компонентом МАС, основная функция которого заключается в исполнении отдельных задач. Пусть агент реализован программно, тогда в составе МАС должны быть аппаратные компоненты, предназначенные для исполнения программных агентов. Назовем такие компоненты агентными платформами. Для выполнения агентами МАС их задач могут быть необходимы ресурсы [5]. Будем считать, что программные ресурсы могут быть предоставлены любой агентной платформой размещенным в ней агентам, и, следовательно, задачам, исполняемым данными агентами. Будем считать, что доступность отдельного аппаратного ресурса обусловлена наличием линии связи между агентной платформой и соответствующим аппаратным ресурсом. Таким образом, определенный аппаратный ресурс доступен отдельной задаче МАС, если данная задача выполняется агентом, который размещен в агентной платформе, обладающей линией связи с данным аппаратным ресурсом. Пусть МАС задана следующими множествами: множеством задач, множеством агентов, являющихся исполнительными контейнерами для задач, множеством агентных платформ, являющихся контейнерами для программных агентов, и множеством аппаратных ресурсов. Распределение задач по агентам и агентов по агентным платформам, доступность аппаратных ресурсов агентным платформам, а также необходимость использования аппаратных ресурсов отдельными задачами заданы предикатами или функциями.

Используя избыточность [2] для обеспечения отказоустойчивости, мы предлагаем резервировать задачи и аппаратные ресурсы МАС. Введем множество типов задач и множество типов аппаратных ресурсов исходной МАС таким образом, чтобы каждому типу однозначно соответствовала задача или аппаратный ресурс исходной системы. Тогда можно использовать резервирование с определенной кратностью для каждого типа задачи и аппаратного ресурса, при этом будем считать, что только одна реплика из множества реплик соответствующего типа является активной, т.е. используется холодное резервирование. Согласно предложенной авторами модели МАС агенты и агентные платформы являются контейнерами для соответственно задач и агентов, значит использовать резервирование агентов и агентных платформ нецелесообразно. Тем не менее, для обеспечения отказоустойчивости мы считаем необходимым внести избыточность в множества контейнеров, т.е. агентов и агентных платформ. Предложенная стратегия резервирования заключается в следующем: вводится резервирование задач и аппаратных ресурсов исходной МАС, а множества исполнительных контейнеров, т.е. агентов и агентных платформ, резервированной МАС вводятся с избыточностью. Процедура синтеза структуры резервированной МАС заключается в выборе распределений задач по агентам и агентов по агентным платформам, удовлетворяющих требованиям о доступности необходимых для выполнения задач аппаратных ресурсов, а также требованиям к уровню надежности резервированной системы при наличии методик расчета показателей надежности.

Модель резервированной МАС, предложенная авторами, предусматривает замену запроса на исполнение конкретной задачи запросом на исполнение задачи определенного типа, приводящим к исполнению соответствующей активной реплики, а также замену запроса на использование определенного аппаратного ресурса запросом на использование аппаратного ресурса определенного типа. Будем считать, что агент обладает всеми необходимыми знаниями об исполняемых им задачах, а агентная платформа, в свою очередь, о размещенных в ней агентах. Кроме того, будем считать, что агентная платформа обладает

знаниями о размещении активных реплик для каждого типа задачи, при этом активная реплика может быть размещена как в одном из агентов данной агентной платформы, так и в любой удаленной агентной платформе. Будем считать, что знания как агентов, так и агентных платформ представлены в виде баз данных (БД).

Рассмотрим следующие отказы резервированной МАС, после обнаружения которых должна быть восстановлена работоспособность: программный отказ задачи, программный отказ агента, вызывающий отказ всех исполняемых данным агентом задач, отказ агентной платформы, вызывающий отказ всех задач, исполняемых агентами, принадлежащими данной агентной платформе, отказ аппаратного ресурса, вызывающий отказ тех задач, для выполнения которых необходимо использование данного аппаратного ресурса. Будем считать, что взаимодействие между компонентами МАС происходит посредством передачи сообщений. Пусть агент обнаружил отказ одной из исполняемых им задач типа *tt*, тогда агент выполняет процедуру *a_t_fail*, которая осуществляет поиск неактивной реплики типа *tt* и активирует ее, а также обновляет собственную БД в случае успеха или передает запрос на активацию реплики *arrepl(tt)* агентной платформе, в которой он размещен. При получении данного запроса агентная платформа выполняет процедуру *hwp_t_fail*, которая осуществляет поиск неактивной реплики типа *tt* в агентах, принадлежащих данной агентной платформе, и активирует ее с обновлением собственной БД и БД соответствующего агента в случае успеха или передает запрос на активацию реплики *hrrepl(tt)* всем удаленным агентным платформам. При получении сообщения *hrrepl(tt)* агентная платформа осуществляет поиск неактивной реплики типа *tt* в принадлежащих ей агентах и активирует ее в случае успеха с обновлением БД соответствующего агента и собственной БД. Агентная платформа, активировавшая реплику типа *tt* по запросу *hrrepl(tt)*, рассылает всем остальным агентным платформам сообщение *ranc(tt, id)*, при получении которого агентные платформы обновляют в собственных БД информацию о размещении активной реплики типа *tt*. Пусть агентная платформа зафиксировала отказ одного из принадлежащих ей агентов, тогда агентная платформа согласно собственной БД определяет множество типов отказавших активных реплик и выполняет для каждого типа процедуру *hwp_t_fail*, которая приводит к поиску и активации соответствующих реплик при условии их наличия в агентах данной агентной платформы или в агентах удаленных агентных платформ. Если агентной платформой обнаружен отказ аппаратного ресурса, то агентная платформа определяет множество типов активных реплик, для выполнения которых необходим данный аппаратный ресурс, и обеспечивает поиск и активацию соответствующих реплик, выполняя процедуру *hwp_t_fail* для каждого типа. Пусть агентная платформа зафиксировала отказ одной из удаленных агентных платформ, тогда агентная платформа согласно собственной БД определяет множество типов отказавших активных реплик, т.е. таких активных реплик, которые размещены в отказавшей агентной платформе. Для каждого типа отказавшей реплики агентная платформа выполняет процедуру *hwp_t_fail*. Если после завершения процедуры *hwp_t_fail* соответствующая реплика активирована в рамках данной агентной платформы, то агентная платформа рассылает всем удаленным агентным платформам сообщение *ranc*, информируя их о произошедших изменениях в размещении активных реплик. Авторами была сформулирована и доказана теорема о том, что при применении разработанной методики восстановления работоспособности МАС сохраняет работоспособность при программных отказах задач и агентов, а также отказах агентных платформ и аппаратных ресурсов, т.е. обеспечивает выполнение задачи любого типа *tt*, если выполнено условие сохранения полноты

резервирования, т.е. если на момент обнаружения отказа в МАС существует неактивная задача типа *tt*, имеющая доступ ко всем ресурсам, необходимым для ее выполнения.

Выводы. Для обеспечения отказоустойчивости МАС необходимы методика синтеза резервированной МАС и методика восстановления работоспособности резервированной МАС. Предложенный авторами подход гарантирует восстановление работоспособности МАС при обнаружении отказов ее компонентов. Кроме того, разработанная стратегия резервирования может быть положена в основу методики синтеза резервированной МАС, однако процедура синтеза резервированной МАС также требует дальнейших исследований в области оценки показателей надежности программно-аппаратных МАС.

Научная новизна результатов работы заключается в следующем:

- Разработанная модель распределенной программно-аппаратной МАС позволяет учитывать как распределение задач по агентам, так и агентов по соответствующим исполнительным аппаратным компонентам в отличие от известных методик;
- Разработана стратегия резервирования МАС, основанная на резервировании задач, как элементарных функциональных элементов, а также на резервировании аппаратных ресурсов и введении избыточных множеств агентов и агентных платформ, выполняющих роль контейнеров для соответственно задач и агентов;
- Разработанная методика восстановления работоспособности гарантирует сохранение работоспособности МАС при программных отказах задач и агентов, отказах аппаратных ресурсов и агентных платформ.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. N. R. Jennings. On agent-based software engineering. Artificial intelligence, 2000, Vol. 117, No. 2, pp. 277-296.
2. L. L. Pullum. Software fault tolerance techniques and implementation. Artech House, 2001. 360 p.
3. Z. Guessoum, J. P. Briot, N. Faci. Towards fault-tolerant massively multiagent systems. Massively Multi-Agent Systems I, Springer Berlin Heidelberg, 2005, pp. 55-69
4. S. Kumar, P. R. Cohen. Towards a fault-tolerant multi-agent system architecture. Proceedings of the fourth international conference on Autonomous agents, ACM, 2000, pp. 459-466
5. S. Mellouli. A reorganization strategy to build fault-tolerant multi-agent systems. Advances in Artificial Intelligence, Springer Berlin Heidelberg, 2007, pp. 61-72
6. M. Wooldridge, N. R. Jennings. Intelligent agents: Theory and practice. Knowledge engineering review, 1995, Vol. 10, No. 2, pp. 115-152.

УДК 004.415.5

А. Д. Янкович, И. В. Никифоров, В. П. Котляров (СПбГПУ)

БИБЛИОТЕКА АНАЛИЗА ПОТОКОВ УСМ ДИАГРАММ

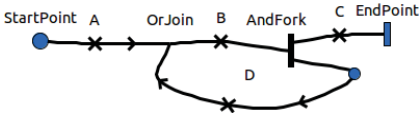
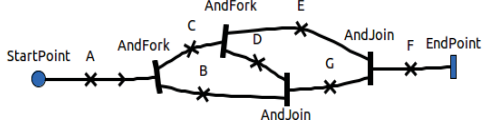
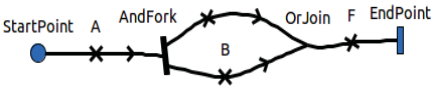
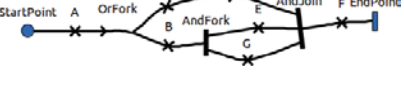
Проблемная область. Создание программного продукта является трудоемким процессом, в котором приходится работать с большим объемом требований на проектируемую систему[1]. Первой фазой разработки является создание дизайна системы на формальном языке[2]. Одним из формальных языков высокоуровневого описания системы является язык УСМ[3]. Он позволяет значительно упростить процесс разработки, а так же уже на стадии

дизайна выявить ошибки в спецификациях, исправление которых в конечном программном продукте может привести к значительному увеличению стоимости конечного продукта.

Обзор недостатков языка UCM. UCM [3] позволяет формализовать модели системы различного уровня сложности, объема и моделировать многопоточные системы, временные задержки и прерывания. Но несмотря на все достоинства он содержит неточности в формулировках семантики многопоточных конструкций, а как следствие позволяет создавать семантически некорректные системы.

В таблице 1 представлены конструкции языка UCM, приводящие к некорректному поведению в модели[4].

Таблица 1. Некорректные многопоточные конструкции

№	Название	UCM диаграмма	Описание
1	Неограниченное рекурсивное порождение потоков		Диаграмма характеризуется циклом OrJoin-AndFork, в котором нет ограничений на число генераций новых потоков, т. к. дочерний поток возвращается по циклу на элемент, который его породил.
2	Нарушение баланса скобок		Синтаксически правильные элементы порождения и синхронизации потоков на графе могут приводить к нарушению структуры параллельных потоков.
3	Некорректная синхронизация		Параллельные потоки объединяются в общее поведение элементом OrJoin, характерным для линейных участков исполнения.
4	Синхронизация родительского и дочернего потоков		Частный случай нарушения баланса скобок, характеризуемый синхронизацией родительского потока с дочерними.

В работе перед автором *ставилась задача* разработки библиотеки анализа UCM проекта на наличие семантически некорректных конструкций, *целью* которой является автоматическое определение ошибочных мест проекта на этапе создания модели системы.

Для *решения поставленной задачи в работе* предлагается использовать формальное определение потока, а так же формулировки некорректных ситуаций, которые реализованы в алгоритме автоматического анализа UCM проекта.

Для анализа UCM диаграммы вводится формальное определение потока.

Определение 1. Под потоком T будем понимать кортеж $T=(S,P,C,E)$, где S — элемент диаграммы, который порождает поток; P — множество всех потоков, которые являются «родителями» потоку T ; C — множество всех потоков, которые являются «дочерними» потоку T ; E — множество элементов диаграммы, которые входят в поток T .

Определение 2. Под «родителями» потока подразумеваются потоки, в которых порожден рассматриваемый поток. У потока может не быть явно заданного родителя в случае, если он считается стартовым (*main*) для системы.

Определение 3. Под «дочерними» потоками подразумеваются потоки, которые порождаются родительским потоком. У потока может не быть дочерних потоков, если он достигает элемента EndPoint.

Каждая семантически некорректная конструкция языка UCM может быть выражена формулой, истинность которой для всех потоков UCM диаграммы является индикатором некорректности.

В формулах используются следующие обозначения:

$T_i \in T_a$ — i поток, содержащийся во всех потоках T_a UCM проекта, где $a \in 1:N$;

$T_i^{S_j}$ — i поток, порожденный на элементе S_j , где $S_j \in S$;

$S_i \xrightarrow{n} S_i^{next}$ — UCM элемент S_i^{next} , достижимый из элемента S_i за n шагов;

$T_i \rightarrow C_j$ — поток T_i порождает дочерний поток C_j ;

$T_i \perp T_j$ - поток T_i синхронизируется с потоком T_j ;

Приведем формулировки некорректных ситуаций (табл. 1).

- *Неограниченное рекурсивное порождение потоков.*

$$(\forall T_i^{S_j} \in T_a) \rightarrow (C_k^{S_k} \in T_a) \wedge (S_j \xrightarrow{N \neq 0} S_k) \wedge (S_i = S_k)$$

Если для любого потока T_i из множества всех потоков диаграммы T_a , порожденного на элементе S_j верно утверждение, что данный поток порождает дочерний поток C_k на элементе S_k , при этом элемент S_k достижим за ненулевое число шагов N от элемента S_j , и элемент S_k и S_j - это один и тот же элемент диаграммы, то диаграмма считается некорректной.

- *Нарушение баланса скобок и синхронизация родительского и дочернего потоков* определяется по следующей формуле.

$$(\forall T_i^{S_j} \in T_a) \perp (C_k^{S_k} \in T_a) \wedge (T_i \rightarrow C_k)$$

Если любой поток T_i из множества всех потоков диаграммы T_a , порожденный на элементе S_j синхронизируется с потоком C_k , порожденным на элементе S_k , при этом поток C_k является дочерним потоком для T_i , то диаграмма считается некорректной.

- *Некорректная синхронизация.*

$$((T_1^{S_1} \in T_a) \perp (T_2^{S_2} \in T_a) \perp \dots \perp (T_n^{S_n} \in T_a)) \wedge (P_i \rightarrow T_i (P_i \notin \emptyset), i \in 1:n) \wedge (C \in T_a = \emptyset)$$

Если синхронизируются потоки T_1, T_2, \dots, T_n , у которых родители не пусты $P_1 \notin \emptyset, P_2 \notin \emptyset, \dots, P_n \notin \emptyset$, и нет общего родителя $C \in T_a$, принадлежащего множеству всех потоков UCM диаграммы, то UCM диаграмма считается некорректной.

В работе реализована библиотека анализа параллельных потоков UCM диаграммы, в основе которой лежит алгоритм поиска и выявления упомянутых некорректных ситуаций. На рис.1 представлена схема алгоритма. Она состоит из 4 этапов:

1. Обход всех элементов UCM диаграммы. При выполнении данного этапа происходит перебор всех элементов UCM диаграммы и формирование списка глобальных стартовых точек, формирование списка точек порождения потоков и точек синхронизации. Под

глобальной стартовой точкой диаграммы понимается элемент StartPoint, который не связан со входами элементов Stub, т. е. не участвует в иерархии путей.

2. Разметка UCM диаграммы потоками. Последовательным движением от каждой глобальной стартовой точки к следующим за ней элементам пути осуществляется обход UCM диаграммы до тех пор, пока не будет достигнут конечный элемент EndPoint. В процессе обхода происходит маркировка участков пути потоками. В начальный момент алгоритм размечает путь потоком, который порожден на стартовой точке. Если во время обхода был встречен элемент порождения или синхронизации потоков, то в массив потоков добавляется новый поток и разметка пути продолжается вновь созданным потоком.
3. Анализ связей и отношений о потоках. На основе формул, задающих признаки некорректных ситуаций и массива потоков, сформированного на этапе (2), осуществляется анализ и отбор мест UCM диаграммы, в которых выполняются соответствующие признаки.
4. Логирование информации о потенциально опасных местах, выявленных на этапе (3).



Выводы: Разработанные в работе признаки выявления некорректных ситуаций и библиотека анализа, реализующая алгоритм автоматического определения некорректных UCM конструкций, позволяет выявить потенциально опасные места и ошибки на самых ранних этапах разработки формальных моделей систем. Библиотека анализа внедрена в инструмент UCM Event Analyzer (EVA)[5].

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Б. В. Тютин, И. В. Никифоров, В. П. Котляров. Построение системы автоматизации статической и динамической проверки требований к программному продукту // Научно-технические ведомости СПбГПУ. № 4(152). СПб.: Изд-во Политехнического ун-та, -2012. -С. 119-123.
2. И. В. Никифоров, А. В. Петров, Ю. В. Юсупов. Генерация формальной модели системы по требованиям, заданным в нотации USE CASE MAP // Научно-технические ведомости СПбГПУ. № 4(103). СПб.: Изд-во Политехнического ун-та, -2010. -С. 191-195.
3. ITU Recommendation Z.151. User Requirement Notation (URN), 2008.
4. И. В. Никифоров, В. П. Котляров, П. Д. Дробинцев. Ограничения на многопоточные конструкции и временные задержки языка UCM // Научно-технические ведомости СПбГПУ. № 3(174). СПб.: Изд-во Политехнического ун-та, -2013. -С. 148-153.
5. И. В. Никифоров, А. В. Петров, В. П. Котляров. Статический метод отладки тестовых сценариев, сгенерированных с помощью эвристик // Научно-технические ведомости СПбГПУ. № 4(152). СПб.: Изд-во Политехнического ун-та, -2012. -С. 114-119.

РАЗРАБОТКА МЕТОДОВ И СРЕДСТВ КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ ДЛЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Внедрение функций защиты в распределенные вычислительные сети (РВС), применяемые при обработке информации ограниченного доступа, сопровождается снижением их масштабируемости и распараллеливаемости, препятствуя совместному использованию ресурсов, тем самым превращая РВС во множество слабосвязанных узлов. Рост числа нарушений безопасности в РВС (например, инциденты CVE-2009-0046 в Sun GridEngine, CVE-2013-4039 в IBM WebSphere Extended Deployment Compute Grid, связанные с повышением полномочий пользователей) определяет необходимость в обеспечении защиты информации при минимальных потерях в функциональности РВС. Данная проблема особо актуальна для РВС, к которым предъявляются требования функциональной надежности и безопасности: для систем моделирования и обработки операционных данных объектов энергетики, анализа финансовых рисков, проведения ядерных, геоинформационных и научных исследований, для информационно-аналитических систем.

Известны результаты исследований в области моделирования РВС и обеспечения их защиты от внешнего нарушителя, в том числе от угроз распространения вредоносного программного обеспечения (например, в [1-5]). В прикладных продуктах, например, в подсистемах GRAM в РВС Globus Toolkit, CAS в РВС gLite, реализованы механизмы авторизации пользователей для доступа к вычислительным мощностям РВС [6, 7]. В существующих решениях описание и исполнение политик безопасности (ПБ) производятся на множестве виртуальных организаций (ВО) и фиксированных состояний РВС, что не учитывает их высокую динамику, распределение прав на уровне вычислительных задач пользователей и приводит к возможности несанкционированного доступа (НСД) к обрабатываемым данным.

Данная работа опирается на результаты указанных исследований и развивает их в направлении обеспечения защиты данных, обрабатываемых в распределенных вычислительных сетях, от угроз превышения полномочий пользователей с сохранением функциональных свойств сетей путем разработки метода контроля и управления доступом. В работе решались следующие задачи:

- Анализ угроз безопасности и механизмов защиты РВС.
- Построение модели распределения пользовательских задач в РВС, позволяющей описать их параллельное выполнение с учетом функциональных свойств РВС и заданных ограничений доступа.
- Разработка метода построения карты состояний РВС, позволяющей определять допустимые распределения пользовательских задач в условиях высокой динамики их миграции между узлами РВС.
- Разработка метода контроля и управления доступом, обеспечивающего защиту информации, обрабатываемой в РВС, от угроз превышения полномочий пользователей.
- Разработка архитектуры и реализация системы контроля и управления доступом пользовательских задач к ресурсам РВС.

Научные результаты работы заключаются в том, что впервые введено понятие ветвящейся раскрашенной функциональной сети Петри, что позволило разработать научную концепцию моделирования распределения пользовательских задач в РВС, связывающую принципы организации распределенных вычислений с предъявляемыми требованиями ПБ. Для определения допустимых распределений пользовательских задач в условиях высокой динамики их миграции между вычислительными узлами предложено использовать аппарат деревьев достижимости, примененный для ветвящихся сетей Петри. В ходе работы доказана теорема безопасности доступа в РВС, которая утверждает, что сеть безопасна тогда и только тогда, когда безопасны начальное состояние и все состояния, являющиеся узлами дерева достижимости ветвящейся сети Петри, описывающей данную РВС. В работе предложен метод контроля и управления доступом в РВС, обеспечивающий разграничение прав доступа на уровне пользовательских задач.

В ходе работы создана архитектура и реализована система контроля и управления доступом пользовательских задач к вычислительным ресурсам РВС, представлены результаты оценки эффективности ее работы.

Внедрение системы контроля и управления доступом в РВС обеспечивает защиту обрабатываемых данных от угроз превышения полномочий пользователей, одновременно позволяет сократить временные затраты, связанные с оценкой безопасности, и увеличить производительность РВС, обрабатывающих информацию ограниченного доступа.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. A. Chakrabarti. Grid computing security. – Berlin: Springer, 2007. 331 с.
2. S. Kar. An Anomaly Detection System for DDoS Attack in Grid Computing. – International Journal of Computer Applications in Engineering, Technology and Sciences, 2009. P. 553-557.
3. H. Koshutanski. Enhancing Grid security by fine-grained behavioral control and negotiation-based authorization. – International Journal of Information Security Volume 8, Issue 4. 2009. P. 291-314.
4. S. Song. Fuzzy Trust Integration for Security Enforcement in Grid Computing. – Network and Parallel Computing, Lecture Notes in Computer Science Volume 3222. 2004. P. 9-21.
5. H. Lohr. Enhancing Grid Security Using Trusted Virtualization. – Autonomic and Trusted Computing, Lecture Notes in Computer Science Volume 4610. 2007. 372-384 сс.
6. Globus toolkit version 4 grid security infrastructure: A standards perspective. – <http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf>, режим доступа: свободный.
7. A. Sciaba. Glite 3.2 User Guide, Manuals Series. – Experiment Integration and Distributed Analysis. 2011.

**THE SEARCH OF PROGRAMS AREAS WITH INTENSIVE IRREGULAR MEMORY ACCESS
BY COMPUTING THE METRICS OF SPATIAL AND TEMPORAL LOCALITY BY THE
ROUTE OF ACCESS ADDRESSES**

The problem of achieving high real performance on applications is one of the most important when using supercomputers [1]. At the same time, the key issue in achieving high real performance is the efficiency of applications interaction with memory [2].

Spatial and temporal locality of access are widely used as the parameters characterizing the applications scheme of access to the memory elements [3]. To investigate the performance of supercomputers memory subsystem according to the spatial and temporal locality of access the APEX-map test was developed and used [4]. According to the results of measurements the dependence of time to access the memory element and the parameters of spatial and temporal locality is plotted (it is apex-map).

The study [2] discusses the methodology of improving the application performance on supercomputers using the metrics calculations of spatial and temporal locality. One of the main ideas of the proposed approach is in the reflection of the point of the spatial-temporal locality of the application on the synthesized apex-map, and, if possible, in an attempt to offset the point of application to the tasks with a good spatial and temporal locality.

The method for mapping the spatial and temporal locality of the application on the synthesized apex-map of the computing system has been proposed in [3] and developed in [5]. The route of application memory accesses received by a special profiler prepared using the library Dyninst [6] was used to calculate the metrics in [5].

However, the use of the developed profiling tool is related to a number of complexities. A significant problem was the presence of the accesses in the route received by using the profiler based on the Dyninst library, system accesses, which sometimes significantly distorted the localities values. In addition, the route of addresses of memory accesses, even in the case of test applications, is of a significant amount. For example, for the tests of NAS Parallel Benchmarks [7] the saved memory calls log comprises hundreds of gigabytes. In this case, the computing cores of today's applications are not uniform with respect to memory access scheme. Thus, the calculation of spatial and temporal locality of the entire application is time consuming, but the result gives little information.

To check the adequacy of the results obtained in the measurement of the spatial and temporal locality of applications, as well as to improve the calculation formulas, the model technology was implemented for getting the calls track to the memory by manual source code instrumentation. Using this model eliminates the influence of systemic factors, and also allowed to work out various locality calculation options.

Two methods were tested to solve the problem of searching the fragments of programs with poor spatial and temporal locality according to the track of memory calls. In the first - the whole calls track was divided into equal portions, and an independent calculation of locality for each of them - interval calculation was performed. In this case, the search of the unit with poor spatial and temporal locality is by the maximum values of localities. In the second the accumulation of locality values was performed from reading to reading, arranged on the track - the calculation by accumulation. In this

case, the search for the unit with poor space and time locality was carried out on the relative dynamics of locality values change.

The problem of solving a system of linear equations with sparse disordered positive definite matrix by using the conjugate gradient method of NAS Parallel Benchmarks (CG test) was discussed as a model problem. The choice of such a model problem is explained by the relative simplicity of the source code and the presence of both units with predictably good, and with predictably bad parameters of spatial and temporal locality.

Block diagram of one iteration of the conjugate gradient method, and the parameters of the spatial and temporal are shown in Fig.1.

Блок схема решения уравнения : $b=Ax$

Инициализация начальных данных :

$$\begin{aligned} x^0 &= x \\ r^0 &= b - Ax^0 \\ z^0 &= r^0 \\ p^0 &= 0 \end{aligned}$$

Итерация метода (шаг k):

A	1) $p^k = Az^{k-1}$
B	2) $h_1 = (p^k, z^{k-1})$ $h_2 = (r^{k-1}, r^{k-1})$ $\alpha^k = h_2/h_1$
	3) $x^k = x^{k-1} + \alpha^k z^{k-1}$
	4) $r^k = r^{k-1} - \alpha^k p^{k-1}$
	5) $h_3 = (r^k, r^k)$ $\beta^k = h_3/h_2$
	6) $z^k = r^k + \beta^k z^{k-1}$
	Конец шага итерации .

Расчет локализации

этап	временная локализация (TL)	пространственная локализация (SL)
A	0,72	0,21
B	0,55	0,99
общая	0,68	0,25

Fig. 1. CG algorithm block diagram and locality parameters

Research results demonstrate the possibility to search for blocks with poor spatial and temporal locality by calculating metrics. It is also shown that the interval calculation of the locality value allows for better division of algorithm blocks by parameters of spatial and temporal of memory calls, which makes this method of calculating the priority in the implementation of the automated search.

Documents development and research results representation are used MS Office.technology

In the further work it is planned to implement a fully automated profiling, which allows to identify the areas of applications code affecting the synthesized apex-map outside the area of good space and time locality of the computer system.

REFERENCES:

1. В.С.Горбунов, Г.С. Елизаров, Л.К.Эйсымонт, «Проекты экзафлопсных суперкомпьютеров за рубежом и в России, ограничения и перспективы роста», материалы Четвертого Московского Суперкомпьютерного Форума, 2013
2. Д.В.Андрюшин, В.С.Горбунов, Л.К.Эйсымонт, «О некоторых подходах к оптимизации работы суперкомпьютеров», материалы Национального Суперкомпьютерного Форума, 2013.

3. Weinberg J. et al. Quantifying Locality In The Memory Access Patterns of HPC Applications // Proceedings of the 2005 ACM/IEEE conference on Supercomputing, SC '05.
4. Strohmaier E., Shan H., Architecture Independent Performance Characterization and Benchmarking for Scientific Applications // Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications System, Washington, 2004.
5. Максимов В. И. Диссертация на соискание учёной степени магистра «Реализация методики измерения пространственно-временной локализации приложения и ее отображения на синтезируемую АРЕХ-поверхность используемого оборудования». — СПб.: СПбГПУ, 2013. 72 с.
6. Paradyn/Dyninst homepage — <http://www.dyninst.org>
7. NAS Parallel Benchmarks homepage — <http://www.nas.nasa.gov/publications/npb.html> .

УДК 004.932.2

Е. И. Короткова, А. В. Самочадин (СПбГПУ)

СРЕДСТВА РАСПОЗНАВАНИЯ СУЩЕСТВЕННЫХ ИЗМЕНЕНИЙ НА АЭРОФОТОСНИМКАХ

Детектирование регионов существенных изменений на аэрофотоснимках является распространенной задачей. Информация о существенных изменениях требуется для работы большого количества приложений, например: разведка использования полезных ресурсов человеком (лес, карьеры), изменения природных условий в связи с катастрофическими явлениями, мониторинг построек, транспорта, выявление недопустимых ситуаций.

Задача детектирования заключается в том, чтобы по двум и более наборам фотографий одной местности, сделанными с беспилотного летательного аппарата (БПЛА) в разное время, найти области "существенного различия". Существенные изменения ищутся только для объектов, принадлежащих заранее определенному множеству (например, здания и сооружения, участки леса и т.д.). Информация может быть нескольких типов: появление, исчезновение, перемещение и изменение объекта.

Изначально распознавание данных изменений проводилось оператором, который сравнивал снимки визуально, но этот подход не эффективен при больших объемах данных. Задача мониторинга требует средств автоматизации, которые позволили бы сократить количество работы оператора и уменьшить вероятность ошибок, обусловленных человеческим фактором. Особенность задачи детектирования в том, что все: трактовка исходных данных, точность решения, алгоритм, трактовка термина «существенное изменение» зависят от того, что будет входить в множество искомых объектов.

В рамках данной работы будет спроектирована автоматизированная система, которая распознает "существенные изменения" объектов заданного типа.

На вход системы поступают два и более наборов снимков заданной местности. На выходе — данные о расположении изменившихся объектов в локальных координатах снимков.

В процессе работы система выполняет следующие этапы обработки снимков:

–Коррекция геометрических искажений (углы фотографирования, высота, неточность в географической привязке снимков).

Выбор соответствующей пары снимков.

1. Коррекция хроматических искажений (уровень яркости, тени).
2. Избавление от помех.

3. Вычисление «разности» снимков.
4. Выявление контуров и объектов.

Предложенные методы должны давать преимущества перед базовыми методами по следующим критериям:

- Процент пикселей, классифицированных правильно.
- Коэффициент Жаккара.
- Yule coefficient.
- Кривая детектирования в зависимости от SNR.
- Кривая детектирования в зависимости от разрешения исходных снимков.

Для реализации подхода, удовлетворяющего указанным требованиям, была разработана методика, которая позволяет избавляться от геометрических искажений на снимках. Снимки проецировались на соответствующий рельеф, данные о котором были известны априорно (модель SRTM [3]), а затем были «сняты заново» в ортопроекциях[4]. Тем самым частично устраняются искажения на снимках, связанные с рельефом, которые невозможно было бы устранить с помощью аффинных преобразований. Данная методика была реализована в программе на Java.

На основе обзора методов для анализа изображений одного объекта на наличие «существенных изменений» на каждом из описанных выше шагов были предложены способы использования, а также улучшения существующих алгоритмов для решения данной задачи.

Предложенные методы рассматривают фотографию как случайный процесс, который паре координат на снимке(сделанном в определенный момент времени) ставит в соответствие интенсивность(а также цвет), что позволяет рассматривать фотографию как случайное поле.

Разработанные методы позволили реализовать этап коррекции геометрических искажений. Реализация последующих этапов обладает следующими особенностями:

- Нормализация интенсивности по всему снимку $\frac{\sigma_1}{\sigma_2}(I_2(x) - \mu_2) + \mu_1$. Улучшение: разбивать снимок на области одинаковой интенсивности и нормализовать отдельно.
- При некоторых условиях можно рассмотреть сцену как Ламбертову поверхность(интенсивность отраженного света не зависит от угла). Тогда можно отфильтровать отраженный свет от падающего с помощью низкочастотного фильтра[2].

Избавление изображений от помех

- При отношении сигнал-шум, вычисленном для фотографий как $\frac{\mu}{\sigma} < 5$ (критерий Роуза) можно применить спектральную фильтрацию[2].
- Удаление теней со снимков на основе минимизация функции Поттса с использованием алгоритма разреза графа[5]. Предлагается адаптировать этот метод под данную задачу и использовать в нем информацию о двух снимках.

Разность фотографий

- Простая разность $D(x) = I_2(x) - I_1(x)$
- $B(x) = 1, \text{if } (D(x)) > \tau$
- $B(x) = 0, \text{otherwise}$
- Чувствительна к шумам, изменениям в освещенности, не учитывает локальных особенностей.
- Проверка статистических гипотез. Выбор разных статистик.

- Использование алгоритма поиска разности изображений на основе Марковских полей и метода Монте-Карло [1]. В данном алгоритме предлагается учитывать специфику задачи при построении модели.

С помощью предложенного алгоритма для исправления геометрических искажений, были обработаны 2 набора снимков Красной поляны и Саблино. Один раз с реальной телеметрией, полученной во время полета, второй раз с телеметрией, искусственно полученной из 3D модели, построенной по снимкам. Во втором случае результат намного лучше - он близок к разрешению снимков.

Выводы. Дальнейшая работа состоит в реализации базовых алгоритмов на всех этапах методики, а также реализации перечисленных более гибких алгоритмов. В соответствии с требованиями нужно представить в отчете графики и отношения, показывающие выигрыш на каждом этапе перед базовыми алгоритмами.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Kasetkasem T. , Varshney P. K. An Image Change Detection Algorithm Based on Markov Random Field Models // Geoscience and Remote Sensing, IEEE. – 2002. – Vol.40 , No. 8. – P. 1815 – 1823
2. Richard J. R. Image Change Detection Algorithms: A Systematic Survey //Transactions on image processing, IEEE. – 2005. – Vol.14 , No. 3. – P. 294 – 307
3. Дубинин М. Описание и получение данных SRTM // GIS-Lab: географические информационные системы и дистанционное зондирование. Интернет-журнал. 23.01.2004. URL: <http://gis-lab.info/qa/srtm.html>
4. Falkner E., Morgan D. Aerial Mapping: methods and applications // 2-е изд. - Boca Raton : CRC Press, 2002.- 216 с.
5. Lu C., Drew M. S. Shadow Segmentation and Shadow-Free Chromaticity via Markov Random Fields // 13th Color imaging conference (IS AND T SID).- 2005. - P. 125-129.

УДК 681.391

Р. А. Морозов, П. В. Трифонов (СПбГПУ)

УКОРОЧЕНИЕ ПОЛЯРНЫХ КОДОВ

Полярные коды [1] — первый класс кодов, достигающих пропускной способности канала при длине кода, стремящейся к бесконечности. Длина полярного кода равна 2^m , на практике же требуются коды произвольной длины.

Чтобы построить полярный код заданной длины $2^{m-1} < n < 2^m$ и размерности $k \leq n$, мы укорачиваем и выкалываем некоторый полярный (N, K)-код, $N = 2^m$, $K \geq k$.

Укорочение [5] означает добавление проверочных соотношений информационных символов так, чтобы на фиксированных позициях в кодовом слове всегда оказывались нули; размерность и длина кода при этом уменьшаются на единицу. При выкалывании символы на фиксированных позициях кодового слова не передаются, т. е. длина кода уменьшается на единицу. Таким образом, нам задано количество укороченных или выколотых символов

$l = N - n$. Мы решаем задачу поиска количества укороченных символов l_s и выколотых символов l_p с помощью перебора по всем возможным вариантам, таким что $l_s + l_p = l$, а также собственно множеств укорачиваемых и выкалываемых позиций S и P , причём $|S| = l_s$, $|P| = l_p$, $S \cap P = \emptyset$. Критерием качества заданной пары множеств S и P выберем вероятность ошибки декодирования, которая может быть найдена методом эволюции плотностей, предложенным в [2]. На вход алгоритму подаются распределения логарифмических отношений правдоподобия (ЛОПП) на выходе каналов передачи данных. Затем с помощью t уровней преобразований, соответствующих преобразованиям в алгоритме декодирования последовательного исключения (рис. 1), получают распределения ЛОПП на выходе алгоритма.

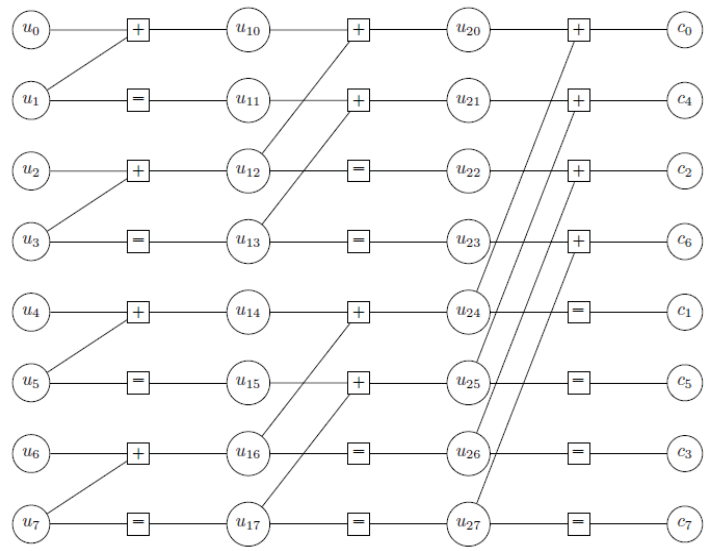


Рис. 1. Факторграф декодирования полярного кода длины 8

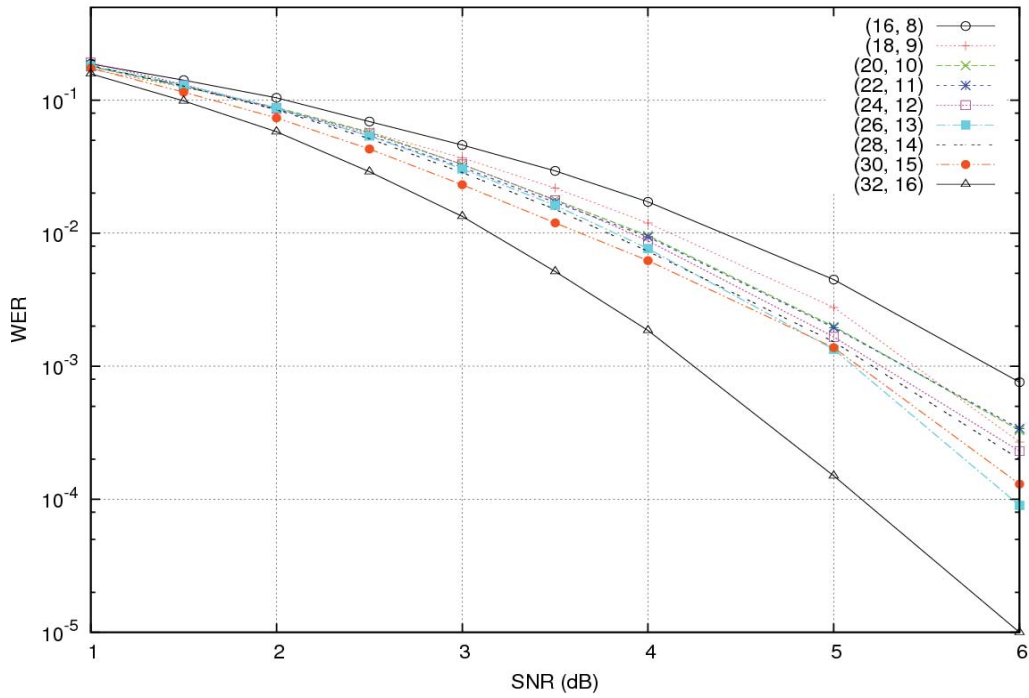
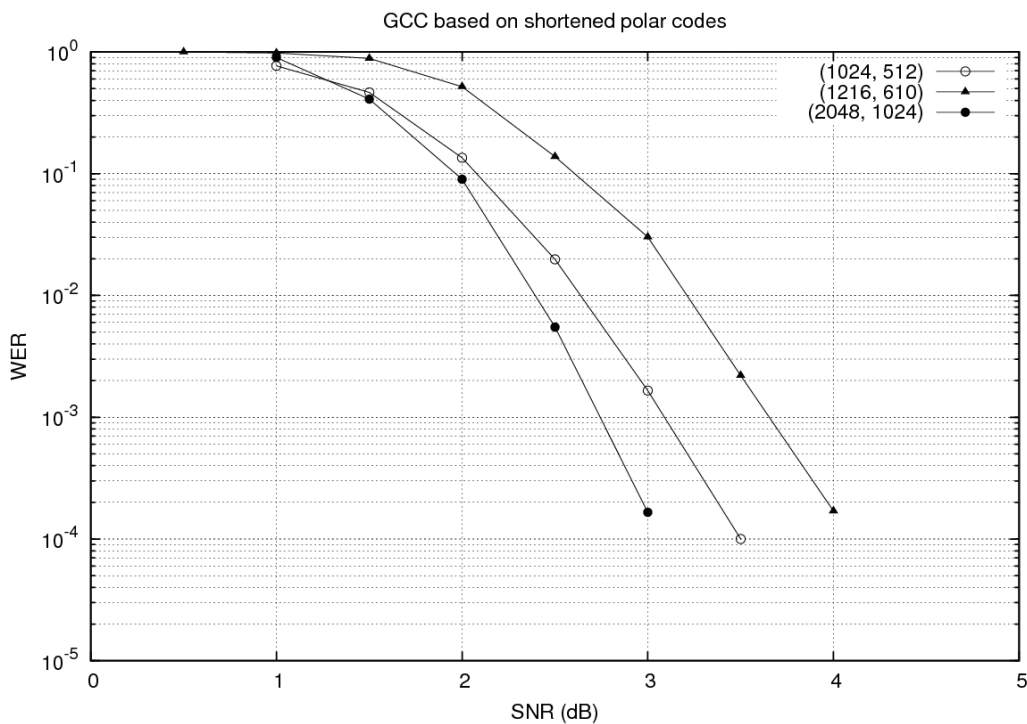


Рис. 2. Вероятность ошибки декодирования укороченных кодов алгоритмом Box and Match



Рассматривается передача нулевого кодового слова. В нашем случае распределения ЛОПП соответствуют каналу W с выходом $1+\eta$, $\eta \sim N(0, \sigma^2)$. Мы считаем, что символы на укороченных позициях передаются по идеальному каналу W_s с выходом $+\infty$, а выколотые

символы — по каналу W_p с выходом 0. Входом алгоритма для заданных S и P являются распределения каналов $(W_i)_{i=1}^{2^m}$, $i \in S \Rightarrow W_i = W_s$, $i \in P \Rightarrow W_i = W_p$, $i \notin S \cup P \Rightarrow W_i = W$.

Мы перебираем все возможные варианты S , фиксируем лучшее найденное множество, а затем перебираем все возможные варианты P с учётом того, что множество укороченных позиций равно S . Перебор осуществляется с помощью дерева разделений укороченных (выколотых) позиций. Корень дерева соответствует числу укороченных позиций, приходящихся на всё кодовое слово (l_s). Потомки корневого узла соответствуют числу укороченных позиций, приходящихся на каждую половину кодового слова в соответствии с рисунком 1. Продолжая такое разделение, на $m+1$ -м уровне мы имеем 2^m листьев, в каждом из которых находится 0 или 1. Единицы соответствуют укороченным позициям.

В процессе работы метода эволюции плотностей применяются две операции над каналами — сумма и исключающее или, которые являются симметричными. Эти операции над распределениями ЛОПП соответствуют следующим операциям над ЛОПП:

$$L_N^{(2^i)}(y_0^{N-1}, u_0^{2^i-1}) = 2 \tanh^{-1} \left(\tanh \frac{L_N \left(y_0^{\frac{N}{2}-1}, u_{0,e}^{2^i-1} + u_{0,o}^{2^i-1} \right)}{2} \tanh \frac{L_N \left(\frac{y_N}{2^{N-1}}, u_{0,o}^{2^i-1} \right)}{2} \right),$$

$$L_N^{(2^{i+1})}(y_0^{N-1}, u_0^{2^i}) = \frac{L_N}{2^{(i)}} \left(\frac{y_N}{2^{N-1}}, u_{0,o}^{2^i-1} \right) + \frac{(-1)_{2^i}^u \cdot L_N \left(y_0^{\frac{N}{2}-1}, u_{0,e}^{2^i-1} \oplus u_{0,o}^{2^i-1} \right)}{2^{(i)}}$$

Заметим, что на рис. 1 мы можем переставить местами половины третьего уровня (самого правого на рисунке) без изменений распределений ЛОПП на втором уровне. На втором уровне можно переставить местами смежные четверти и т. д. Таким образом, учитывая указанную симметрию, при переборе мы можем рассматривать только случаи, когда, например, левый потомок не меньше правого. Это позволяет существенно уменьшить число перебираемых вариантов. Заметим, однако, что если мы нашли множество укороченных (выколотых) позиций, то после, при поиске множества выколотых (укороченных) позиций вышеуказанная симметрия нарушается и приходится применять полный перебор. Однако такое уменьшение перебираемых вариантов позволило нам построить коды длины до 32. На рис. 2 представлены вероятности ошибки декодирования построенных кодов алгоритмом Voh and Match [5]. Для построения длинных кодов предлагается использовать обобщённую каскадную конструкцию с внутренними полярными кодами [4] и стекковый алгоритм декодирования [3]. На рис. 3 можно видеть график вероятности ошибки декодирования алгоритмом последовательного исключения кода (1216, 610).

Выводы. Были построены укороченные полярные коды длины до 32. Длинные коды предлагается получать путём применения обобщённой каскадной конструкции.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. E. Arıkan. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. // IEEE Transactions on Information Theory, Vol. 55, No. 7. — July 2009. — Pp. 3051–3073.
2. Tal I., Vardy A. How to construct polar codes // IEEE Transactions On Information Theory.—2013.— October.— Vol. 59, no. 10
3. Tal I., Vardy A. List decoding of polar codes // Proceedings of IEEE International Symposium on

Information Theory.— 2011

4. Trifonov P. Efficient design and decoding of polar codes // IEEE Transactions on Communications.— 2012.—November.— Vol. 60, no. 11.— Pp. 3221 – 3227

5. Ф.Дж. Мак-Вильямс, Н. Слоэн. Теория кодов, исправляющих ошибки. М. – Связь, 1979.

УДК 004.8

К. Е. Моногаров (СПбГПУ)

МЕТОД ОПРЕДЕЛЕНИЯ ОРИЕНТАЦИИ ОБЪЕКТОВ НА ОСНОВЕ EM-АЛГОРИТМА

В настоящее время различные робототехнические системы начали активно входить во многие сферы жизни человека. Всевозможные роботы начинают активно применяться в быту (пылесосы), промышленности, военном деле и даже при полетах в космос. Часто для подобных систем встает задача определения ориентации относительно каких-либо объектов и планирование траектории при сближении с ними. В данной работе предлагается алгоритм для определения ориентации робототехнической системы относительно заданного объекта на основе трехмерных данных, полученных посредством измерения лазерными измерительными системами (ЛИС).

Целью данной работы является реализация алгоритма определения ориентации удаленного объекта по его снимкам, полученным с лазерной измерительной системы.

Задач данной работы: реализация алгоритма определения ориентации интересующего объекта; реализация программного обеспечения на основе алгоритма определения ориентации; оценка качества определения ориентации с различных расстояний и на различных наборах шаблонных снимков объекта (сделанных с разных расстояний); оценка качества определения ориентации в зависимости от количества шаблонных снимков объекта.

Задачей данного алгоритма является определение ориентации удаленного объекта относительно объекта, имеющего в своем составе ЛИС, результатом работы которой является массив точек в хуз-координатах. Предполагается, что эти объекты постепенно сближаются и для объекта, имеющего ЛИС, требуется определить траекторию, по которой он может приблизиться к заданному месту объекта, ориентация которого определяется.

Итак, для работы алгоритма требуется набор снимков модели, по которым будет определяться ориентация. Ключевая идея алгоритма заключается в том, что имея такой набор снимков модели и получая очередной снимок объекта с ЛИС, мы производим сравнение этого полученного снимка с набором снимков модели и выбираем из них посредством специальной нормы наиболее похожий. Каждый снимок модели должен быть сделан из строго определенного положения. Таким образом, определив снимок модели, самый похожий на снимок, полученный с ЛИС, мы можем сделать выводы о положении объекта с ЛИС относительно объекта, ориентация которого определяется. Следовательно, появляется возможность корректировки траектории движения объекта с ЛИС.

Целью сопоставления облаков точек является установление соответствия между двумя наборами точек и восстановление преобразования, которое переводит один набор точек в другой. На качество определения такого соответствия сильно влияет погрешность в результатах измерений, полученных с помощью ЛИС.

Для реализации алгоритма определения ориентации подходят широко распространенный метод ИСП [1] и параметрический метод CPD [2]. Результат их сравнения между собой показал, что второй из перечисленных алгоритмов предоставляет лучшие и более стабильные результаты.

Процесс выравнивания двух наборов точек рассматривается сводится к задаче оценки плотности распределения вероятностей. Каждая точка в снимке модели является центром 3-мерной гауссианы. Все точки снимка модели представлены как сумма плотностей в гауссовой модели смеси [3, с. 59-73].

Введем обозначения, которые в дальнейшем будем использовать.

D – размерность пространства;

$X(N, D)$ – множество точек снимка, полученного с ЛИС;

$Y(M, D)$ – множество точек снимка модели.

Как отмечалось выше, набор точек Y представляет центры гауссовой модели смеси, а множество точек X интерпретируется как выборка значений из той же модели смеси. Плотность распределения гауссовой модели смеси представлена следующим выражением:

$$p(x_i) = \sum_{j=1}^M P(j) p(x_i|j),$$

где

$$p(x_i|j) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left\{-\frac{\|x_i - Ry_j - t\|^2}{2\sigma^2}\right\}.$$

Кроме этого, в модель смеси можно добавить нормальное распределение, которое позволяет в некоторой степени бороться с шумами и выбросами. Обозначая вес нормального распределения как ω , $0 \leq \omega \leq 1$, модель смеси примет следующую форму:

$$p(x_i) = \omega \frac{1}{N} + (1 - \omega) \sum_{j=1}^M P(j) p(x_i|j).$$

Сопоставление двух облаков точек реализуется посредством итеративного уточнения положения центров гауссовой модели смеси с помощью последовательного уточнения параметров этой смеси методом максимального правдоподобия. Функция правдоподобия выглядит следующим образом:

$$L = \prod_{i=1}^N p(x_i).$$

Следует отметить, что максимизация правдоподобия эквивалентна минимизации отрицательной логарифмической функции правдоподобия, известной как функция ошибки:

$$E = -\ln L = -\sum_{i=1}^N \ln p(x_i).$$

Уточнению подлежат параметры R , t и σ^2 . Для этого используется EM-алгоритм [4, с. 24-30]. Идея EM-алгоритма заключается сначала в приблизительном определении "старых" значений параметров с использованием априорной плотности распределения, а потом с использованием теоремы Байеса в получении апостериорных плотностей распределения компонент смеси:

$$P(j | x_i) = \frac{P(j)p(x_i | j)}{p(x_i)}$$

В этом заключается E-шаг алгоритма (expectation). Далее "новые" значения параметров находятся минимизацией отрицательной логарифмической функции правдоподобия (функции ошибки) относительно новых значений параметров. Так как известно, что минимизация функции ошибки эквивалентна максимизации функции правдоподобия, то данный шаг алгоритма называется M-шагом. Таким образом, EM-алгоритм попеременно действует между двумя этими шагами, пока не сойдется.

Для нахождения минимума функции ошибки ее нужно продифференцировать по параметрам $\theta(R, t, \sigma^2)$.

В результате работы было реализовано программное обеспечение, которое позволяло оценивать качество определения ориентации целевого объекта относительно объекта с измерительной системой.

Принцип работы программы следующий: на вход программы подается набор снимков модели и поочередно подаются снимки, получаемые с измерительной системы, точки съемки которых расположены на сфере относительно ориентируемой модели.

Формальный алгоритм определения качества ориентации таков. После загрузки очередного снимка происходит определение самого "близкого" снимка модели к полученному снимку. В результате на основе SVD-разложения [5, с. 59-70] вычисляется матрица поворота, переводящая один снимок в другой, и ошибка, характеризующая степень схожести снимков, по которой и оценивается "схожесть" снимков. Далее для определения качества оценки ориентации определяется ошибка поворота в градусах. Для этого требуется определить нормали к снимку модели и снимку, полученному с измерительной системы. Затем нормаль снимка модели поворачивается до нормали снимка, полученного с измерительной системы посредством умножения на вычисленную матрицу поворота. Угловое расстояние между этими векторами определяется из следующего выражения:

$$err = \arccos(V1 \cdot V2)$$

где $V1$ – вектор нормали к снимку, полученному с измерительной системы;

$V2$ – вектор нормали к снимку модели, умноженный на вычисленную матрицу поворота.

Процесс определения ориентации считается успешно выполненным, если вычисленный угол будет меньше, чем заданный порог. Подобный набор операций выполняется для всех снимков, полученных с измерительной системы, расположенных на сфере относительно модели. Таким образом можно судить о качестве определения ориентации и судить о том, с каких положений и с каких расстояний определение ориентации осуществляется хуже или лучше при имеющихся на данный момент средствах и методах.

В работе была оценена зависимость качества распознавания от количества шаблонных снимков модели. Наборы снимков делались из 32, 20, 14 и 8 положений. С целью ускорения процесса обработки их разрешение составляло 100 на 100 точек. Наилучший результат при определении ориентации был получен при наборе из 14 снимков. Так, при увеличении количества снимков модели было замечено, что выбирался не самый "лучший" снимок модели и появлялись ошибки в определении ориентации. Также были проведены эксперименты по определению порога выбросов (outliers - выбросы) для алгоритма сопоставления облаков точек. Изображения сравнивались с порогами выбросов "0", "0,0003" и "0,01". Наилучшие результаты были получены при пороге "0,0003". Как и следовало ожидать, качество сопоставления изображений, а, следовательно, и качество определения ориентации объекта ухудшалось с увеличением расстояния.

Так как в результате измерения в кадр попадает не только объект интереса, но и его фон, то часто это отрицательно влияет на качество сопоставления снимков, а соответственно и определения ориентации. Таким образом, в настоящее время имеет актуальность поиск менее чувствительных к фоновым точкам методов сопоставления облаков точек, а также производится сравнение качества работы различных методов сегментации, которые позволяют выделять только интересующий объект и производить дальнейшие операции с ним.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Censi A. [An ICP variant using a point-to-line metric](#). ICRA'08.
2. Myronenko A., Song X. Point Set Registration: Coherent Point Drift. Электронный документ: <https://sites.google.com/site/myronenko/research/cpd> (дата обращения: 11.11.2013).
3. Bishop C. M. Neural networks for pattern recognition. Oxford university press, 2005. 482 p.
4. [Воронцов К. В. Лекции по статистическим \(байесовским\) алгоритмам классификации](#). URL: <http://www.machinelearning.ru/wiki/images/e/ed/Voron-ML-Bayes.pdf> (дата обращения: 9.11.2013).
5. Press W. H. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 1992. 949 p.

УДК 004.056

Е. Ю. Павленко, М. О. Калинин (СПбГПУ)

РАЗРАБОТКА СЕРВИСА РАЗРЕШЕНИЯ КОНФЛИКТОВ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

Стремительный рост использования автоматизированных информационных систем, распределенных компьютерных систем, центров обработки данных и сетевых систем хранения данных стимулирует насущную потребность в надежных, высокопроизводительных, хорошо управляемых компьютерных сетях [1]. В настоящее время наиболее активно развивающейся сетевой технологией являются программно-конфигурируемые сети, в которых, в отличие от традиционных компьютерных сетей, решение об изменении маршрутов передачи сетевых пакетов принимается выделенным устройством — контроллером программно-конфигурируемой сети. Он является единственным элементом сетевой инфраструктуры, осуществляющим управление всей сетевой системой [2]. Добавление правил

передачи данных происходит путем приема контроллером правила от сервисов управления, имеющих возможность добавлять правила передачи данных, преобразования его в необходимый формат и отправки на сетевое устройство по протоколу OpenFlow. Такой подход порождает проблему обеспечения корреляции новых правил с сетевой политикой безопасности.

Описанная выше проблема обусловлена отсутствием соответствия новых правил, добавляемых сервисами ПКС, взаимодействующими с контроллером, правилам политики безопасности программно-конфигурируемой сети.

Для решения описанной проблемы необходимо разработать сервис управления, обеспечивающий авторизацию субъектов, добавляющих правила на основе ролей, и разрешение потенциально конфликтующих правил добавления потоков. Конфликтная ситуация возникает тогда, когда кандидат на запись в таблицу сетевых потоков должен создать или удалить запись в таблице потоков, которая запрещена (или разрешена), в соответствии с существующими правилами [3].

Данный сервис управления должен логически располагаться между интерфейсом управления программно-конфигурируемой сетью и сервисами управления, осуществляющими выработку новых правил передачи данных.

Для аутентификации сущностей, которые имеют возможность работы с таблицами сетевых потоков OpenFlow-сетевых устройств, предлагается введение трех базовых ролей. Первая роль — сетевой администратор, правила обработки сетевого трафика имеют наивысший приоритет. Вторая роль — сервис обеспечения безопасности программно-конфигурируемой сети. Эта роль отводится сервису, обеспечивающему правила управления безопасностью для реализации политики безопасности. Третья роль — сервисы управления, осуществляющие выработку правил передачи данных в программно-конфигурируемой сети. Аутентификация приложений производится на основе электронно-цифровой подписи. Каждое приложение имеет свой ключ для реализации ЭЦП. В случае если правило не подписано, оно получает наименьший приоритет и обрабатывается как правило, полученное от сервиса управления [4].

Для детектирования конфликтов между кандидатом на добавление в таблицу сетевых потоков OpenFlow коммутатора и существующим набором правил формируются кортежи, состоящие из следующих элементов:

- R-адрес источника;
- IP-адрес назначения;
- сетевая маска;
- порт источника;
- порт назначения.

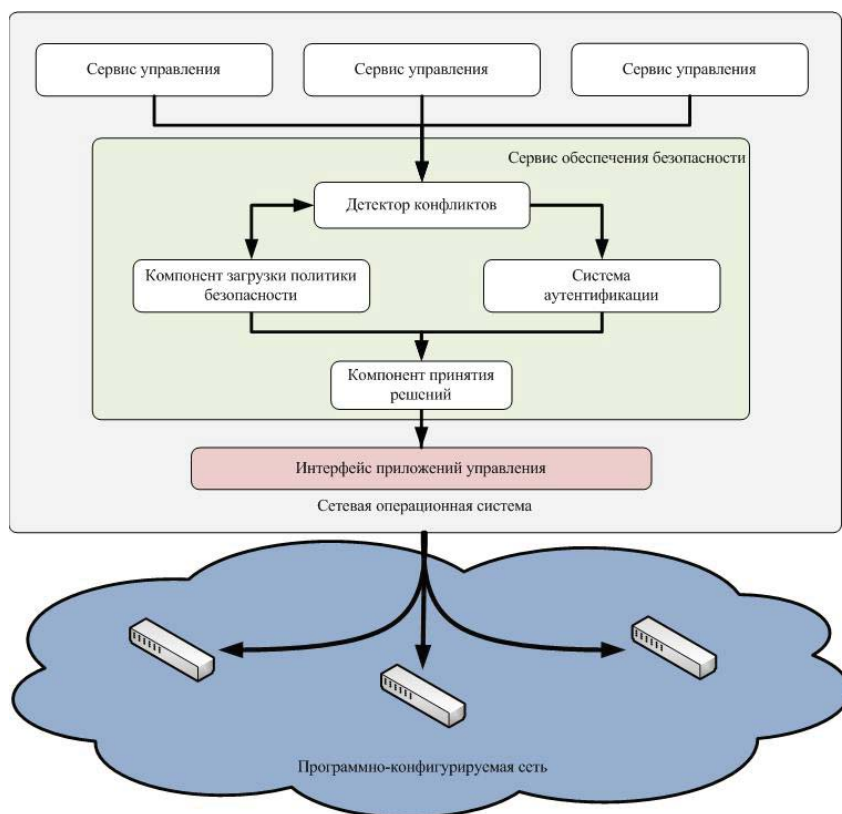


Рис. 1. Структура сервиса обеспечения безопасности программно-конфигурируемой сети

IP-адрес источника, сетевая маска и порт источника объединяются в пару элементов, определяющих источник. IP-адрес назначения, сетевая маска и порт назначения объединяются в пару элементов, определяющих получателя. Данные пары элементов используются для определения конфликта.

В общем виде политику безопасности программно-конфигурируемой сети можно описать правилами вида (1).

$$(a) \rightarrow (b) \text{ отбросить пакет} \quad (1)$$

В данном правиле пара, определяющая источник — (a) , а пара, определяющая получателя — (b) .

Пусть в данной программно-конфигурируемой сети появляется набор правил (2), которое одно из приложений хочет добавить в правило управления передачей сетевого трафика.

$$\begin{aligned}
a &\rightarrow c \text{ правило модификации таблицы потоков} \\
a_1 &\rightarrow c \text{ правило модификации таблицы потоков} \\
a_1 &\rightarrow b \text{ передача пакетов}
\end{aligned}
\tag{2}$$

В соответствии с введенными парами источника и получателя данный набор правил может быть записан в виде (3).

$$(a, a_1) \rightarrow (b, c) \text{ передача пакетов} \tag{3}$$

Данное представление набора правил, полученного от приложения, показывает, что при добавлении этого набора правил, возможно, будет нарушена политика безопасности программно-конфигурируемой сети, т. к. становится возможной передача данных от элемента (a) к элементу (b) .

Для формализации правила поиска конфликтующих записей необходимо определить следующие переменные: S_i — это запись в политике безопасности с номером i ; F_i — это запись в таблице потоков с номером i ; $SI_{i,j}$ — это условие использования записи с номером j , в записи с номером i в политике безопасности; $FI_{i,j}$ — это условие использования записи с номером j , в записи с номером i в полученном от приложения наборе правил; SA_i — это действие в i -ой записи в политике безопасности; FA_i — это действие в записи в таблице потоков с номером i . С использованием введенного обозначения становится возможным сформулировать критерий наличия конфликта в программно-конфигурируемой сети в виде (4).

$$\text{Если существует хотя бы 1 запись в политике безопасности, такая что } SI_{i,j} \cap FI_{i,j} \neq \emptyset \text{ и } SA_i \neq FA_i \text{ для всех } j, \text{ тогда } S_i \text{ конфликтует с } F_i. \tag{4}$$

В случае нахождения конфликтующих записей разрешение конфликта происходит на основе введенных ролей. Если роль существующего правила выше, чем у правила, которое хочет быть добавленным в таблицу сетевых потоков, его добавление не происходит. Если роль существующего правила ниже, чем у правила, которое хочет быть добавленным в таблицу сетевых потоков, существующее правило удаляется, а вместо него записывается новое правило. Если роль существующего правила равна роли правила, которое хочет быть добавленным в таблицу сетевых потоков, решение о добавлении или удалении правил принимает сетевой администратор [5].

Выводы. Предложенный сервис обеспечения безопасности программно-конфигурируемой сети обеспечивает безопасную настройку сетевых ресурсов программно-конфигурируемой сети, а также предотвращает возможность обхода сетевой политики безопасности.

ЛИТЕРАТУРА:

1. Р. Л. Смелянский. Проблемы современных компьютерных сетей. – СПб.: Материалы XIX всероссийской научно-методической конференции «Телематика», 2012.
2. N. McKeown. OpenFlow: Enabling Innovation in Campus Networks. – Стэнфорд: Stanford University, 2008.
3. I. Yamahata. Software Defined Networking, openflow protocol and its controllers. – Чикаго.: LinuxCon, 2010.
4. H. Uppal. OpenFlow Based Load Balancing. – Вашингтон.: University of Washington, 2012.
5. P. Porras. A Security Enforcement Kernel for OpenFlow Networks. – Техас.: Texas A&M University, 2012.

Секция «Подходы к разработке программного обеспечения на основе технологий EMC»

УДК 681.3

Т.А.Берленко, К.А. Фертиков (5 курс, каф. МО ЭВМ, СПбГЭТУ),
В.А. Грозин (3 курс, каф. ИС, ИТМО),
К.В.Кринкин, к.т.н, доц. каф. МОЭВМ СПбГЭТУ

РАЗРАБОТКА МЕХАНИЗМА УДЕРЖАНИЯ ШЛЮЗА В ЯДРЕ LINUX ДЛЯ СИСТЕМ ХРАНЕНИЯ ДАННЫХ

В системах хранения данных, призванных обеспечить заданное качество обслуживания (QoS), высокую производительность и изоляцию клиентских потоков данных, возможностей существующих механизмов маршрутизации недостаточно. В частности, если рассматривать реализацию стека TCP/IP в ядре операционной системы Linux, можно видеть, что для сервера, обладающего несколькими физическими сетевыми интерфейсами выбор обратного маршрута зависит от алгоритма работы подсистемы маршрутизации, что не позволяет простым методом установить правило фиксации адреса ближайшего шлюза для соединений. Однако, такая задача является актуальной и целью настоящего проекта является разработка механизма удержания шлюза и реализация его в ядре Linux.

Основная идея предлагаемого механизма заключается в следующем. При приёме пакета, инициирующего открытие соединения, запоминается MAC-адрес интерфейса, являющегося источником пакета. Эта информация доступна в структуре `sk_buff`, выполняющей роль контейнера для всех данных, связанных с сокетом. Для управления состоянием сокета и соединения в функцию `tcp_v4_rcv` был добавлен флаг, индицирующий режим удержания шлюза и соответствующая проверка при анализе входящих и исходящих пакетов. В момент отправки пакетов в обратном направлении при включённом режиме удержания шлюза пакеты отправляются на тот сетевой интерфейс с которого произошла инициализация соединения в обход правилам маршрутизации на хосте. Таким образом, решается задача перенаправления исходящих пакетов через тот же интерфейс, через который были приняты входящие в пределах конкретного соединения.

Для управления режимом удержания шлюза была добавлена опция `SOCK_IPREFLECT` в поле `sk_flags` структуры `sock`. Для установки этой опции может быть использован стандартный интерфейс `setsockopt`. Все изменения внесённые в ядро Linux были оформлены в виде патча, доступного для скачивания по следующей ссылке: <https://github.com/OSLL/ipreflect>

Проверка работы патча осуществлялась в сети из трех машин: клиента, сетевого шлюза и сервера. Сервер соединён со шлюзом через две подсети: для этого и сервер, и шлюз имеют по два сетевых интерфейса. Шлюз, также, имеет третий интерфейс — для соединения с

клиентом (через третью подсеть). Целью тестирования была проверка работоспособности следующего сценария: при подключении клиента к IP-адресу первого интерфейса сервера, запросы и ответы должны проходить через указанный интерфейс; аналогичным должно быть прохождение пакетов и при подключении клиента к IP-адресу второго интерфейса. Анализ прохождения пакетов через сетевые интерфейсы осуществлялся с помощью утилиты tcpflow, запущенной на машине сервера.

Тесты показали работоспособность решения для указанной сетевой конфигурации. Дальнейшая работа над механизмом удержания шлюза будет направлена на взаимодействие с протоколом UDP, так как одного только TCP недостаточно для полноценного взаимодействия с клиентом, а также на анализ производительности и совместимости с механизмами маршрутизации для всего стека протоколов TCP/IP.

Таким образом, разработанный механизм решает поставленную задачу удержания шлюза для TCP соединений.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 519.872.2; 519.872.8

С.И. Гиндин (асп. каф.ИВС СПбГПУ)

МОДЕЛИРОВАНИЕ ОБЛАЧНЫХ СИСТЕМ С ПОМОЩЬЮ МНОГОКАНАЛЬНЫХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С РАЗОГРЕВОМ

Современные облачные информационные системы, как правило, представляют собой многоканальные системы массового обслуживания (СМО) клиентских заявок. Существуют разнообразные варианты классификации СМО, среди которых особенный прикладной интерес представляют системы с "разогревом" – разогрев требуется системе перед началом обработки первой заявки после перерыва. Примером прикладного смысла "разогрева" в облачных системах являются затраты на установление зашифрованного соединения при подключении пользователя или затраты на кеширование базой данных информации при первом обращении к ней.

СМО с разогревом статически характеризуются по трем показателям: интервалу между соседними заявками, времени, затрачиваемому на "разогрев", и времени на обработку заявок.

В докладе рассматриваются актуальные проблемы развития и недостатки существующих моделей с экспоненциальным распределением, предложено семейство моделей многоканальных СМО с "разогревом" с двухфазным гиперэкспоненциальным распределением "разогрева", предлагается расчетная схема вычисления стационарного распределения числа заявок в системе. Приводятся диаграммы переходов и матрицы переходов между микросостояниями системы. Рассматриваются примеры численных расчетов стационарного распределения числа заявок в системе для различных значений коэффициента вариации, а также для различных значений интенсивности "разогрева".

Рассматривается вопрос приложения исследуемых моделей для расчета практических облачных систем, применяемых в индустрии.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА

1. Kolahi, S.S. Simulation Model, Warm-up Period, and Simulation Length of Cellular Systems // Published in: Intelligent Systems, Modelling and Simulation (ISMS), 2011 Second International Conference on . 25-27 Jan. 2011. Page(s): 375 – 379
2. Takahashi Y., Y. Takami. A numerical method for the steady-state probabilities of a GI/G/c queuing system in a general class//J. of the Operat. Res. Soc. of Japan. – 1976. – V. 19, N 2. – P. 147-157.
3. Рыжиков Ю.И. Распределение числа требований в системе массового обслуживания «с разогревом». Проблемы передачи информации. Том IX, вып. 1, 1973. – С. 88-97.
4. Хомоненко А.Д. Численные методы анализа систем и сетей массового обслуживания. СПб.: МО СССР, 1991. – 179 с.

УДК 004.65

И. И. Закирова(4 курс ИУС), Д. И. Осенняя(4 курс ИУС),
О. В. Александрова(ст. преп. ИУС), П. Д. Дробинцев (доц. ИУС СПбГПУ)

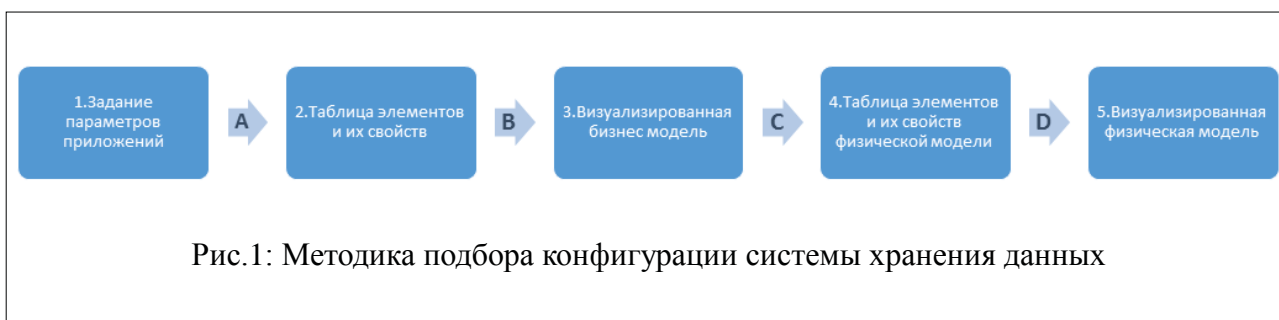
РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ МОДЕЛИРОВАНИЯ СИСТЕМ ХРАНЕНИЯ ДАННЫХ

Хранение информации — важнейшая составляющая информационных технологий. Объём информации, производимой компаниями и частными лицами, растёт с экспоненциальной скоростью. Вместе с этим растёт и сложность систем, обеспечивающих надёжное и эффективное хранение таких объёмов данных. Инновационные продукты корпорации EMC[3] позволяют ИТ-подразделениям создавать гибкие, надёжные и экономичные системы для хранения, защиты информации и управления ею.

Каждому поставщику ИТ-услуг требуется определённая конфигурация системы хранения данных, которая удовлетворяет специфике поставленной задачи и обеспечивает необходимую защиту данных его приложений. Поэтому разработка конфигурации каждой системы хранения трудоёмка и требует индивидуального подхода.

Необходимо разработать программный продукт, позволяющий подбирать конфигурацию системы хранения данных в зависимости от требований заказчика к степени защищённости данных, с которыми работают его приложения, быстродействию приложений, а также к стоимости. Кроме того приложение должно позволять рассчитывать различные характеристики будущей системы, определять её «узкие места» и отказоустойчивость при выходе из строя её различных элементов.

В процессе изучения предметной области была разработана следующая методика, для решения поставленной задачи (рис.1)



На этапе 1 происходит инициализация исходных данных о приложениях. При переходе к этапу 2 на основе набора правил А происходит генерация таблицы элементов бизнес модели с их свойствами. Бизнес модель (в дальнейшем — БМ) - это графическое представление структуры системы хранения в виде логических сущностей (таких как сайт, уровень виртуализации, система хранения данных[1]) и связей между ними. Сгенерированная информация не только отображается на экране, но может также редактироваться пользователем. Далее, используя набор правил В и обращаясь к базе существующих шаблонов, программа генерирует список подходящих конфигураций. Из них пользователь либо вручную, либо указав предпочтения(цена/качество) выбирает нужную. В результате на этапе 3 пользователь получает отображение бизнес-модели и возможность ее редактирования, для более тонкой настройки.

Аналогично предыдущим этапам, с помощью наборов правил С и D, сначала генерируется список элементов физической модели, а затем сама модель. Физическая модель (далее — ФМ) — подробное графическое представление конфигурации системы с помощью реальных устройств (таких как сервер, switch, VPLEX[2]). Как и для БМ существует возможность редактирования свойств элементов. Также появляется возможность моделирования отказов элементов системы и определения «узких мест».

Данная методика используется при создании продукта DataCenter, отвечающего требованиям поставленным выше. Разработка продукта производится в среде Eclipse с помощью инструментов Eclipse Modeling Framework(EMF)[4] и Graphical Modeling Framework(GMF)[5]. На первом этапе создается EMF модель классов и их связей друг с другом. На втором этапе, используя средства GMF, на основе полученной ранее логической модели, генерируется код для графического редактора. В результате, получается RCP приложение, в которое по мере необходимости разработчик вручную вносит требуемые изменения.

К настоящему моменту был разработан прототип программного продукта, позволяющий:

- Создавать, редактировать, сохранять (в формате xml) БМ и ФМ.
- Читать информацию о приложениях из xls-файла, и на основе них генерировать БМ с использованием временных правил преобразования.
- Генерировать на основе БМ ФМ с использованием временных правил преобразования.

Пятиэтапная методика, описанная выше, лежит в основе проекта DataCenter и на данной стадии разработки показывает неплохие результаты для решения задачи конфигурирования систем хранения. Так как используя даже тестовые алгоритмы, она позволяет получить прототип результирующей модели. Усовершенствуя правила преобразования и расширяя

набор встроенных шаблонов, вероятно можно будет добиться более точной работы программы.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. От хранения данных к управлению информацией / EMC – СПб.: Питер, 2010. 544 с
2. Ron Dharma, Li Jiang. Networking for Storage Virtualization and EMC RecoverPoint: EMC, 2013. 102 с.
3. EMC – ведущий поставщик проверенных ИТ-решений, технологий для облачных вычислений и работы с большими данными: URL: <http://russia.emc.com/index.htm>.
4. Eclipse Modeling Framework Project (EMF): URL: <http://www.eclipse.org/modeling/emf/>.
5. Model Driven Architecture approach to domain of graphical editors: URL: <http://eclipse.org/gmf-tooling/>

УДК 004.852

Д. А. Калашников (3 курс, СПбГУ НИУ ИТМО, каф. ИС),
А. А. Суворов (4 курс, СПбГПУ, каф. РВКС).

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДА ВЫБОРА РАЗМЕРА СЛОТА В КЭШЕ НА ОСНОВЕ ПАРАМЕТРОВ РАБОЧЕЙ НАГРУЗКИ

Целью работы является повышение производительности систем хранения данных, использующих кэширование. В работе показано, что выбор размера страницы кэш памяти существенно влияет на эффективность кэширования и напрямую зависит от параметров рабочей нагрузки. Таким образом, основной задачей является разработка алгоритма оценки оптимального размера страницы кэш памяти в зависимости от параметров рабочей нагрузки.

Отправной точкой работы был анализ множества потоков адресов обращения, собранных с работающих систем. В рамках данного проекта было создано программное обеспечение для обработки потока адресов и анализа его характеристик. Также была развернута инфраструктура для обработки множества больших по размеру потоков адресов и сбора обучающей выборки.

В работе были поставлены и решены следующие задачи:

1. Создание утилиты для обработки потока адресов и анализа его характеристик[1]. Это программное обеспечение позволяет оценить такие параметры рабочей нагрузки как перекос популярности данных, распределение размера входных заявок, распределение стековых расстояний[2], процент последовательных заявок и другие. При оценке распределения стековых расстояний использовалась модель кэш-памяти на системах на основе LRU [3] алгоритма с фиксированным размером слота.
2. Перенос разработанного ПО на удаленный сервер, под управлением ОС Windows Server, где при помощи системы непрерывной интеграции Jenkins и компилятора от Visual Studio производилась автоматическая сборка и тестирование.

3. Сбор обучающей выборки. Для сбора обучающей выборки был разработан скрипт на языке Python, выполняющий такие действия как загрузка файлов с потоками адресов с удаленной файловой системы, вызов утилиты для обработки потока адресов и анализа его характеристик (в том числе и оптимального размера кэш слота), сохранение результатов работы утилиты в базу данных PostgreSQL. При этом, схема хранения данных была разработана с учетом удобства последующего доступа к данным из среды программирования R, которую планируется использовать для создания классификатора, оценивающего оптимальный размер слота.

Утилита для обработки потока адресов и анализа его характеристик была разработана на языке C++, при этом не использовались дополнительные библиотеки, помимо стандартных. Это позволило создать систему с высокой производительностью и хорошо предсказуемым поведением.

В рамках теоретического исследования был изучен ряд публикаций, касающихся выбора наиболее эффективных параметров кэш алгоритмов, и получены следующие выводы:

1. Размер страницы может иметь существенное влияние на эффективность работы кэш-памяти [3] (также, ряд практических опытов на развернутой инфраструктуре подтвердили это утверждение на данных, снятых с реальных систем).
2. Существующие подходы к выбору данного параметра строго эмпирические и не основаны на обучении по данным.

Это исследование показало, что существует потребность в создании алгоритма подобного рода.

На текущем этапе завершена работа по сбору обучающей выборки. В дальнейшем, на полученных данных, с использованием среды R, планируется применить стандартные методы машинного обучения для построения модели зависимости оптимального размера слота от параметров рабочей нагрузки и оценить работу полученного алгоритма.

Выводы. В результате нашей работы было создано программное обеспечение для моделирования работы кэш-памяти с подсчетом характеристик рабочей нагрузки и на развернутой инфраструктуре получена обучающая выборка.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Dror G. Feitelson. Workload Modeling for Performance Evaluation. 2002. Springer Verlag.
2. Kristof Beyls, Erik H. D'Hollander. Reuse Distance as a Metric for Cache Behavior. IN PROCEEDINGS OF THE IASTED CONFERENCE ON PARALLEL AND DISTRIBUTED COMPUTING AND SYSTEMS. 2001, pages 617—662.
3. David A. Patterson, John L. Hennessy. Computer Architecture: A Quantitative Approach. Morgan Kaufmann 1990, chapter 5 - Memory Hierarchy Design.

РАНДОМИЗИРОВАННОЕ АВТОМАТИЧЕСКОЕ BLACKBOX ТЕСТИРОВАНИЕ НА ОСНОВЕ СТРОГО ЗАДАННОЙ МОДЕЛИ СИСТЕМЫ

Целью данного доклада является рассмотрение подхода рандомизированного автоматического blackbox-тестирования на основе строго заданной модели объектов системы и методов их управления, практическое применение подхода на примере продукта компании EMC VNХе. Доклад основан на разработке автора, которое используется для тестирования в EMC Russia COE.

В рассмотрение данного подхода войдет:

1. Постановка задачи и описание системы с помощью строго заданной модели
2. Рассмотрение существующих подходов, оценка их преимуществ и недостатков в применении
3. Определение тестов на простоту из класса сложности RP
4. Рассмотрение автоматического blackbox-тестирования:
 - Задание возможных значений для состояний рассматриваемой системы
 - Решение проблемы экспоненциального роста состояний системы в зависимости от заданных значений путем накладывания фильтров и предусловий
 - Задание ожидаемых результатов и постусловий
 - Генерация планов тестирования с использованием SMT-solver'a
 - Собственно тестирование – исполнение планов тестирования с использованием рандомизированного подхода
 - Кластеризация результатов тестирования и верификация их разработчиком
 - Сохранение результатов тестирования в реляционной базе данных
5. Рассмотрение подхода на примере продукта компании EMC VNХе
6. Потенциал и возможности для дальнейшего применения предложенного выше подхода

Предложенный к докладу метод был исследован и реализован для тестирования продукта компании EMC VNХе. В результате реализации была использована строго заданная модель продукта компании EMC, имплементировано дополнение к существующему тестовому фреймворку и протестированно на реальном продукте. Дополнение к фреймворку генерирует планы исполнения, в соответствии с заданными моделью, фильтрами и предусловиями, выполняет тестирование с использованием рандомизированного подхода, проверяет и кластирует результаты тестов, предоставляя возможность к верифицированию разработчику. Таким образом дополнение позволяет автоматически тестировать продукт и верифицировать результаты разработчику по окончании тестирования. Тестовый фреймворк был разработан при непосредственном участии автора, а дополнение к фреймворку написано и синтезировано самим автором на языке программирования Python.

В работе был исследован и предложен способ автоматического тестирования для систем со строго заданной моделью, представлен пример практического применения данного подхода и предоставлено сравнение с существующими подходами. Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. B. Meyer, A. Fiva, I. Ciupa, A. Leitner, Y. Wei, and E. Stapf. Progrmas that test themselves. IEEE Computer, 42:46-55, 2009.
2. Qiwen Xu, Willem-Paul de Roever, Jifeng He. The Rely-Guarantee Method for Verifying Shared Variable Concurrent Programs. Formal Aspects of Computing 9:2:149-174, 1997.
3. S. West, S. Nanz, and B. Meyer. Demonic Testing of Concurrent Programs. 2012.
4. K. Sen and G. Agha. CUTE and jCUTE: Concolic unit testing and explicit path model-checking tools. In Proceedings of the 18-th International Conference on Computer Aided Verification (CAV' 06), volume 4144 of Lecture Notes in Computer Science, pages 419-423. Springer, 2006.

УДК 681.3

Л. А.Скорospelов, Н.С. Раздобреев (2 курс, каф. МО ЭВМ, СПбГЭТУ),
М. М. Заславский (6 курс, каф. САиУ, СПбГПУ),
К.В Кринкин., к.т.н, доц. каф. МИИТ СПбАУ РАН.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИССЛЕДОВАНИЯ КОДОВОЙ БАЗЫ БОЛЬШИХ ПРОГРАММНЫХ ПРОЕКТОВ

На сегодняшний день, размер и сложность программных проектов достигают такого значения, что, не смотря на хорошо организованный процесс разработки (включающий в себя такие техники как использование стилей, ревью и другие), разработчики не в состоянии быстро восстановить логику работы отдельных частей разрабатываемого продукта. При использовании сторонних компонентов ситуация может осложняться наличием недокументированных частей кода, неясности правил сборки проекта и другими факторами. Особенно сложным процессом является поиск причин и исправление программных дефектов в проектах, включающих компоненты различных разработчиков.

Целью данного проекта является разработка такого инструмента анализа программных проектов (Source Code Analyzer), который позволяет создавать базу знаний о программных артефактах (таких как файлы, каталоги, система сборки, элементы программы, включая идентификаторы, фрагменты кода и пр.), устанавливать между ними аннотированные связи, визуализировать их методами инфографики, выполнять эффективный поиск и фильтрацию, а также использовать существующие инструменты (дизассемблеры, анализаторы форматов и пр.).

Основной интерфейс взаимодействия пользователя (разработчика, анализирующего программный проект) с создаваемой программой управления знаниями о кодовой базе наваян дизассемблером IDA. Начальные требования, реализованные в прототипе:

1. Возможность построения ориентированного графа, вершинами которого являются различные объекты исследуемого проекта.
2. Возможность аннотирования объектов графа, связей и объектов файловой системы.
3. Возможность поиска и фильтрации объектов графа.
4. Возможность применить любую утилиту командной строки для объектов графа.
5. Кроссплатформенность.
6. Открытый исходный код.

В качестве средств разработки было решено выбрать связку C++ и Qt, которые позволят выполнить требование кроссплатформенности, а также в дополнение Qt Property Framework

для просмотра свойств объектов графа, Qt-source-highlight для подсветки синтаксиса исследуемых исходных кодов и SQLite в качестве базы данных.

С помощью разработанного приложения можно решить множество задач: расставить связи между файлами проекта; раскрыть иерархию классов, отражая виды их взаимодействия между собой; построить общий граф проекта. Такой граф можно сохранить в файл, что позволит другим разработчикам с его помощью быстрее погрузиться в предметную область.

В ближайшем будущем планируется добавить в приложение ряд новых возможностей. Во-первых, группировать объекты для их более компактного размещения. Во-вторых, автоматизировать создание графов по разным шаблонам (таким как графы вызовов, деревья связей между файлами, визуализация структуры данных). В-третьих, планаризировать граф для более наглядного отображения связей в нем.

В результате проделанной работы было получено приложение, отвечающее всем поставленным требованиям, имеющее дальнейший потенциал для разработки и использования в различных сферах. С помощью полученного приложения программисты могут быстрее достичь понимания ключевых моментов кодовых баз любых крупных программных продуктов.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 681.3

Е. А. Белокопытова (5 курс, каф. ИУС СПбГПУ),
С. А. Фёдоров, ст. преп. каф. ИУС СПбГПУ

ОПТИМАЛЬНОЕ ПОСТРОЕНИЕ И КОНФИГУРАЦИЯ РАСПРЕДЕЛЁННЫХ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

В данной работе исследуется сфера хранения вычислительных ресурсов в современных центрах обработки данных (ЦОД, data centers) и разрабатываются методы повышения эффективности выполнения различных типов вычислительных задач.

Целью работы является программная система, как инструмент, предоставляющий рекомендации по построению ЦОД — описание топологии и архитектуры, которые обеспечивали бы оптимальное соотношение цены и качества ЦОД. Система в виде конечного решения должна учитывать типы задач, нагрузки и требования к работе.

Решением должна предусматриваться возможность не только разработка нового ЦОД, но и эволюция текущей архитектуры ЦОД для повышения его эффективности, безопасности, а также для расширения существующей системы хранения.

Система разрабатывается универсальной для практически всех возникающих задач и ставит своей целью охватить все этапы создания, внедрения и эволюции ЦОД. Многие другие решения специализируются в основном на оптимизации использования отдельных узлов вычислительной сети (например, только системы хранения данных или серверов).

Такой инструмент должен собирать требования от пользователя и в качестве результата предоставлять наиболее подходящий шаблон центра обработки данных, выполняя функции системы поддержки принятия решения. Инструмент будет содержать библиотеку лучших образцов практики топологии, расчета производительности, базу знаний поддерживаемых компонентов ЦОД, интерфейсов, ограничений и т. д. Получив результат, лицо, принимающее

решение, может согласиться с предлагаемым решением или предложить дополнительные параметры для дальнейшего анализа (рис.1).

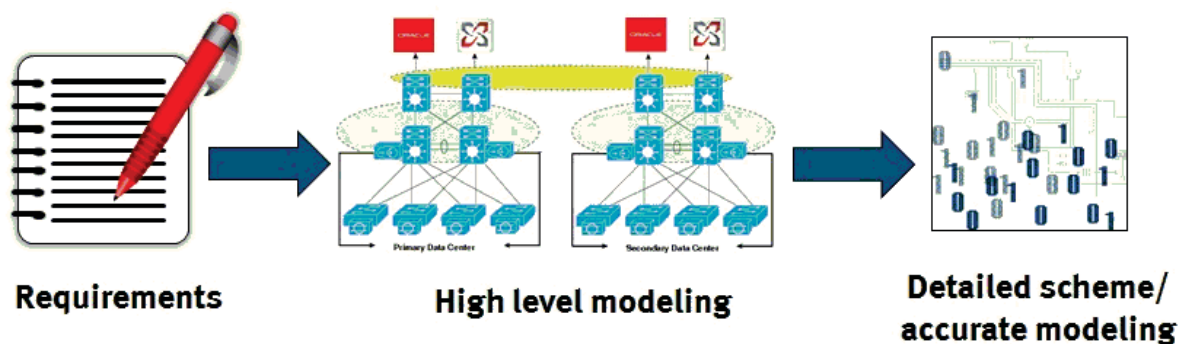


Рис.1 Процесс создания решения.

Основной проблемой является получение и анализ характеристик производительности приложений, то есть сбор показателей работы действующего ЦОД, либо его моделирование. Для этого необходимо учесть требования пользователя к системе. Чтобы информация была наиболее полной, необходимо имитировать реальные ситуации: пиковые нагрузки, отказы и сбои и т. п., и исследовать то, как распределяется рабочая нагрузка и изменяется производительность компонентов системы.

Выводы. Поставленная цель в работе предъявляет определённые особенности по реализации конечного инструмента. Статистика работы ЦОД должна собираться со всех его узлов, так или иначе связанных с его работой, таких как системы хранения данных, коммутаторы, маршрутизаторы и др. Однако, так как элементы ЦОД имеют различную реализацию, то и собранная статистика будет иметь различный формат. По этой причине наиболее разумным решением будет сбор статистики работы только с конкретных узлов системы, например, с хостов. Таким образом, используя инструмент для каждого хоста, можно будет получать информацию в формате, наиболее подходящем для последующей обработки и анализа. Статистику работы хостов удобнее всего строить на основе журналов (logs) — существующей информации, которую итак порождают исследуемые хосты.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

УДК 004.89

С. С. Зобнин, В. Б. Поляков (СПбГПУ)

РЕАЛИЗАЦИЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ХРАНЕНИЯ И ОБРАБОТКИ ВРЕМЕННЫХ РЯДОВ

Целью работы является анализ технологических и алгоритмических аспектов высокопроизводительных интеллектуальных систем хранения и обработки временных рядов с целью последующей интеграции и оптимизации выбранных подходов и воплощения их в виде конечного продукта — интеллектуальной базы данных временных рядов, поступающих с производственного объекта. В настоящее время существует проблема недостаточной масштабируемости и производительности интеллектуальных систем: в то время, как

количество данных (в том числе, временных рядов) нелинейно растет, интеллектуальные алгоритмы усложняются, что не позволяет успевать за увеличивающимися объемами потоков. По этой причине проблемы и задачи, рассматриваемые в данной работе, имеют высокую актуальность, а практические результаты имеют высокую востребованность.

Основным результатом работы является общий алгоритм работы (архитектурное решение) высокопроизводительной интеллектуальной системы хранения и обработки временных рядов. Дополнительным результатом работы является новая схема комбинированного индексирования временных рядов. С одной стороны, данная схема содержит в себе оптимизации, направленные на повышение точности поиска по образам кривых. Результат достигается за счет учета возможных временных сдвигов в сигналах, изменения динамики и масштабов по обеим осям. С другой стороны, использование промежуточного индекса для исходных данных, отображенных в пространство меньшей размерности, позволяет значительно ускорить процесс поиска образа сигнала в базе данных (в работе рассматриваются разные схемы понижения размерности исходных данных). Алгоритм поддерживает одновременно как захват общих деталей сигналов (общей динамики), так и индексирование специфики процессов. Отдельным новшеством является оптимизация и расширение древовидных структур для работы одновременно с простыми запросами по значению и сложными запросами — запросами на кривые, подобные заданной. Вкладом также является и обобщение алгоритма на параллельный: как для систем с разделяемой памятью, так и для систем с отдельной памятью.

При реализации интеллектуальной системы обработки временных рядов естественным образом возникает вопрос о формате представления исходных рядов в памяти машины. В [1] было показано, что подход, основанный на переводе временного ряда в символьное представление через промежуточный этап агрегации имеет ряд неоспоримых преимуществ перед другими подходами. В [2] содержится проанализированное сравнение различных подходов к символьному представлению временных рядов. Из [3] и [4] были почерпнуты идеи для улучшения результатов работы с символьным представлением. После формирования представления исходных данных требуется построить индексы для мгновенного выполнения запросов по виду кривой (образам) и по мгновенному значению потока. Базовой структурой для этого является многомерное дерево, разнообразные модификации многомерных деревьев и многомерных индексов были проанализированы из источников [5, 6, 7]. В работе рассматривается главным образом комбинация из подходов с использованием промежуточных индексов в пространстве меньшей размерности. В качестве базового хранилища для символьного представления и индексов может служить стандартная база данных, как в [8], однако данный подход не удовлетворяет условию задачи, рассматриваемому в данной работе (высокая производительность), так как стандартные хранилища сильно завязаны на структурном описании данных. По этой причине требуется использовать хранилища вида «ключ-значение», лишенные недостатков стандартных баз данных в плане масштабируемости и производительности, а также в плане привязки к формализованному способу описания структуры данных (SQL). Такие хранилища принято называть NoSQL базами данных, существует большое количество различных систем такого класса с преимуществами и недостатками. В данной работе рассматриваются прежде всего библиотеки распределенных вычислений, поддерживающие кеширование рабочего множества в оперативной памяти для ускорения работы и, собственно, хранение в формате «ключ-значение» [9, 10, 11], потому что требуется не только хранить данные в виде пар «ключ-значение», но и обрабатывать их удобным способом. В то же время, описываемая архитектура

системы позволяет не привязываться к конкретному хранилищу, к конкретному способу организации распределенных вычислений, что повышает ее гибкость и переиспользуемость. Существуют и описаны в работе также вопросы представления запросов, интеграции со старыми системами и использования библиотек обработки событий. Система, основанная на изложенных принципах, реализуется в рамках комплексного решения для обработки, хранения и визуализации временных рядов.

Выводы. Подход к реализации высокопроизводительных систем хранения и обработки временных рядов, описанный в данной работе, может быть применен к широкому спектру задач, где входными данными являются временные ряды, есть требования по скорости отклика системы и общей производительности и есть потребность в выполнении разных запросов, как простых, так и сложных. Приведенная схема опережает по основным характеристикам (производительность, время отклика) стандартные подходы к реализации систем данного класса. Основным недостатком схемы является хранение больших индексов в памяти и, как следствие, увеличенные требования к данному ресурсу. Применение комбинированной схемы индексирования временных рядов с промежуточным индексом по данным в пониженной размерности оправданно для задач, где возможно реализовать эффективную схему понижения размерности входного потока.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Jessica Lin, Li Wei, et al. Experiencing SAX: a Novel Symbolic Representation of Time Series. – 2007.
2. A. Sant'Anna; N. Wickstrom. Symbolization of time-series: An evaluation of SAX, Persist, and ACA. – Image and Signal Processing (CISP), 2011 4th International Congress on, vol.4, pp. 15-17.
3. A. M. Ahmed; A. A. Bakar; A. R. Hamdan. Improved SAX time series data representation based on Relative Frequency and K-Nearest Neighbor Algorithm. – Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on, pp.1320,1325.
4. N. D. Pham; Quang Loc Le; Tran Khanh Dang. Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases. – Web Conference (APWEB), 2010.
5. Cui Yu, Bin Cui, Shuguang Wang, and Jianwen Su. Efficient index-based KNN join processing for high-dimensional data. – Inf. Softw. Technol. 49, 2007, 332-344.
6. J. Kuan; P. Lewis. Fast k nearest neighbour search for R-tree family – Proceedings of 1997 International Conference on, vol., no., pp.924,928.
7. Anatoly Sorokin, Gene Selkov, Igor Goryanin. A user-defined data type for the storage of time series data allowing efficient similarity screening. – European Journal of Pharmaceutical Sciences, 2012, pp 272-274.
8. Oleg Bartunov, Teodor Sigaev. Efficient K-nearest neighbour search in PostgreSQL. – conference talk at PGDay, 2010.
9. Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. – Commun. ACM51, 2008, 107-113
10. Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin. Spark: cluster computing with working sets. – In Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10). 2010.

ОПТИМАЛЬНОЕ ХРАНЕНИЕ ВРЕМЕННЫХ РЯДОВ

В ходе работы разработана методика поиска наилучшей (с точки зрения производительности) базы данных, а так же оптимальной структуры данных для хранения временных рядов.

Для эффективного принятия решений необходимо прогнозирование. Для прогнозирования какого-либо явления нужно сперва построить математическую модель этого явления. В свою очередь, для построения математической модели следует выделить структуру из имеющихся данных, а для выявления структуры используются различные статистические методы, такие как регрессионный анализ, кластеризация, и другие. В основе всех методов, упомянутых выше, лежит обработка накопленной истории — данных. Чтобы работать с данными эффективно нужно разработать такую структуру, которая позволяла бы максимально быстро получать данные, сохранять, а так же изменять уже существующие данные. В данной работе речь идет о временных рядах — последовательности значений, собранных в различные моменты времени. Главное свойство временных рядов - взаимосвязь измерений со временем. Типичный пример временных рядов — промышленные системы (накопление и анализ данных с датчиков в разных частях системы для контролирования технического процесса), а так же различные показатели веб-систем (анализ юзабилити интерфейса, где хранятся клики пользователей для последующего анализа и построения более удобного интерфейса), экономических (биржевой курс, используемый для выявления основных направлений развития), и других систем. В условиях огромного потока данных, а так же при временных ограничениях по анализу данных (в промышленных системах важно оперативно находить аномалии в данных чтобы обнаруживать и устранять проблему как только она появилась) появляется проблема эффективного, оптимального хранения таких данных.

Для достижения поставленной цели была разработана методология, которая позволит на основе анализа запросов к данным строить эффективную структуру для более быстрого доступа, записи, а так же обновления данных в документо-ориентированной базе данных на лету (прямо во время действия системы).

На первом этапе был проанализирован подход к хранению временных рядов используя традиционные технологии - реляционной СУБД. Была показана неэффективность данного подхода для одновременного хранения большого количества временных рядов.

На втором этапе работы были проанализированы NOSQL базы данных, их достоинства и недостатки, составлена матрица фич — чем одна база данных лучше другой, или наоборот. В результате данного этапа была выбрана колоночная СУБД [1], где хранение данных производится по парам ключ-значение, а так же по умолчанию упорядочено, как изображено на рис. 1 (что очень важно для временных рядов, и чего нету в традиционной реляционной СУБД, где подразумевается что кортежи хранятся в неупорядоченном виде). В информатике данную структуру можно представить в виде хеш таблицы: `Map<RowKey, SortedMap<ColumnKey, ColumnValue>>`

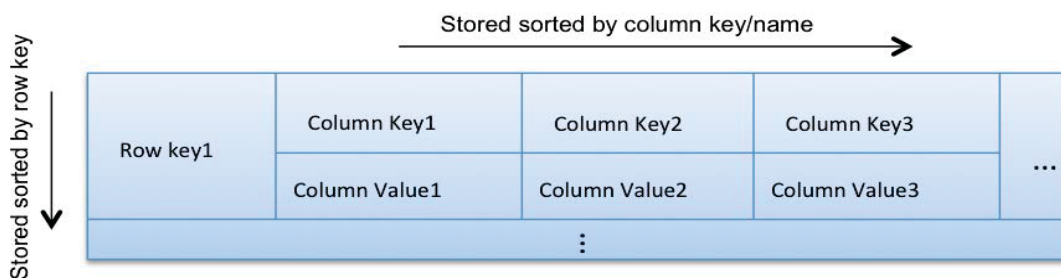


Рис. 1. Структура хранения данных в колоночной NOSQL СУБД

Рассмотрим два варианта хранения временных рядов, полученных от 10000 сенсоров в документо-ориентированной СУБД:

Map<sensor1, SortedMap<Time, Value>>, которая так же может быть представлена в виде таблицы Табл. 1:

Таблица 1. Структура данных в документо-ориентированной СУБД (вариант 1)

sensor1	15:14:59	15:15:00	15:15:01	...
	943.2	950.92	943.2	...

Проблема этой структуры хранения в том, что чем больше хранимый интервал времени — тем длиннее строка, а чем длиннее строка — тем больше время доступа [2]. Еще один существенный минус данной структуры — она очень медленна на запись при 10000 сенсоров. Ведь у каждого сенсора — своя строка, для записи данных в один момент времени нужно обратиться к базе данных 10000 раз. Из существенных плюсов — быстрое чтение любого из сенсоров (за константное время).

Рассмотрим второй способ хранения данных в документо-ориентированной СУБД:

Map<time, SortedMap<sensorName, sensorValue>>, представленной на Табл. 2:

Таблица 2. Структура данных в документо-ориентированной СУБД (вариант 2)

15:14:59	sensor1	sensor2	sensor3	...
	943.2	234.09	1483.5	...

В данной структуре мы ушли от проблемы долгой записи — ведь тут мы производим только 1 обращение к базе данных при каждом моменте времени. Но при этом если мы хотим получить статистику по какому-либо определенному сенсору за определенно время — нужно делать запросы к каждой строке (в каждой строке содержится информация о нужно нам сенсоре только в один момент времени). То есть появилась проблема долгого чтения. Так же к минусам такой структуры можно отнести то, что она не использует свойство упорядоченности по времени во внутренней хеш таблице. (потому что внутренняя хеш таблица не содержит времени, а содержит имена сенсоров, упорядоченность которых нас не заботит)[3][4].

Решение проблем, описанных выше — создание уникальной структуры, которая бы объединяла в себе плюсы обоих подходов. Но при этом была бы максимально оптимизирована под пользовательские запросы. Например если пользователь часто запрашивает sensor1 и sensor2 структуру можно представить следующим образом (см. Табл. 3):

Map<sen1&sen2, SortedMap<Time, Map<sensorName, sensorValue>>

Таблица 3. Оптимизированная структура данных в документо-ориентированной СУБД

sen1&sen2	15:14:59		15:15:00		...
	sensor1	sensor2	sensor1	sensor2	...
	943.2	14.23	-982.15	15.23	...

Третий этап работы — разработка методов, позволяющих автоматизировать процесс создания структур для эффективного хранения временных рядов в документо-ориентированных базах данных[5].

Выводы. Эффективность алгоритмов анализа временных рядов во многом зависит от скорости доступа к данным, то есть напрямую зависит от используемой базы данных, а так же структуры хранения внутри базы данных. При большом потоке информации и при требовании ко времени обработки - появляется необходимость выбора БД, а так же оптимизации по структуре хранения. Цель дальнейших исследований — автоматизация по оптимизации хранения временных рядов в документо-ориентированных базах данных используя статистику запросов к данным. Такие методы позволили бы реконфигурировать БД «на ходу», что в свою очередь позволило бы обращаться а так же записывать данные максимально быстро.

Разработка документации и оформление результатов осуществлялась средствами технологии MS Office.

ЛИТЕРАТУРА:

1. Kelley Reynolds. Basic Time Series with Cassandra. - <http://rubyscale.com/blog/2011/03/06/basic-time-series-with-cassandra>, 2012
2. Robert Rees. NoSQL, no problem. - <http://www.thoughtworks.com/insights/articles/nosql-comparison>, 2010
3. Dennis Shasha. Time Series in Finance. - <http://cs.nyu.edu/shasha/papers/jagtalk.html>, 2013
4. P. J. Bockwell, R. A. Davis -Time Series:Theory and Methods, Springer Series in Statistics, 1986, 337с.
5. А. Ю. Прохоров. Диссертация на тему «Алгоритмы, методы хранения, поиска и анализа временных рядов в промышленных СУБД», 2001, 42 с.

СОДЕРЖАНИЕ

Приветствие Российского представительства корпорации Microsoft	3
Приветствие Российского представительства корпорации EMC	4
Информационное сообщение о проведении учебно-практической конференции студентов, аспирантов и молодых ученых Технологии Microsoft в теории и практике программирования (Новые подходы к разработке ПО на примере технологий Microsoft и EMC)	6
Порядок представления проектов на конкурс-конференцию	8
I. Требования к содержанию конкурсных работ	8
II. Требования к оформлению студенческих работ на конкурс-конференцию	9
III. Требования к оформлению тезисов доклада	10
IV. Методика проведения презентаций и выступлений	12
V. Положение о системе награждения победителей конкурса – конференции	12
Тезисы докладов конкурса-конференции	13
Секция Программная инженерия: приложения, продукты и системы	13
<i>Лихоманова Е.А., Пшеничная К.В.</i> Методические и программные средства для анализа степени защищенности WEB-сайтов	13
<i>Ковыришин П.Ю., Вдовенко А.С., Корзун Д.Ж.</i> Мобильные клиенты для сервисов интеллектуального зала	14
<i>Елесина О.Д., Когоинов С.Ю., Дробинцев Д.Ф., Кауров И.В.</i> Разработка подсистемы планирования загрузок банкоматов на основе Microsoft Analysis Services	15
<i>Андреев А. А., Богоявленский Ю. А.</i> Построение графа ИКТ-инфраструктуры предприятия на канальном уровне с учетом виртуальных локальных сетей	16
<i>Дзюба А. А., Сафонов В. О.</i> Рекомендации музыки на основе графа вкусов	19
<i>Родионов Е.В., Горавнева Т.С.</i> Разработка мультимедийной программы-тренажера	20
<i>Цыбуляк А.Н., Культин Н.Б.</i> Разработка приложения для расчета инновационного потенциала предприятия	21
<i>Янушевский А.А., Дробинцев П.Д.</i> Опыт использования внешних средств разработки среды Eclipse для создания приложений, работающих с большими объемами данных... ..	24
<i>Марченков С.А., Корзун Д.Ж.</i> Распознавание присутствия мобильных устройств в интеллектуальном зале	25
<i>Ковалевский В.Э.</i> Использование свободно распространяемого программного обеспечения для обмена данными между промышленными контроллерами SIMATIC S7 и персональным компьютером	26
<i>Когоинов С.Ю., Каурова Е.А., Кауров И.В., Дробинцев Д.Ф.</i> Разработка модуля	29
<i>Кириленко К.С., Поддорогин С.А., Тутыгин В.С.</i> Разработка портативного анализатора параметров радиоимпульсов	30
<i>Пелконен И.А., Кириленко К.С., Тутыгин В.С.</i> Программа определения времени задержки радиолокационного сигнала	33
<i>Пелконен И.А., Мищенко А.В., Тутыгин В.С.</i> Программа частотного анализа коротких сигналов.	35
<i>Солошенко А. Д., Потехин В. В.</i> Автоматизация сбора и хранения информации об энергопотреблении приборов в домашнем хозяйстве	37
<i>Здробилко С.Н., Дробинцев П.Д.</i> Система выявления аномалий и шумов в недетерминированных сигналах подводных магистралей протяженностью 200 км и более	40
<i>Васильева А. В., Тыжненко Д. А., Потехин В. В.</i> Разработка интеллектуального алгоритма и создание системы управления автономной ветроэлектростанцией	43

<i>Кожяев Л.И., Фёдоров С.А.</i> Разработка модуля автоматизированного проектирования зданий и сооружений в соответствии с нормативной документацией.....	45
<i>Голубев Н.С., Селин И.А., Воинов Н.В., Дробинцев П.Д.</i> Организация сети с ячеистой топологией.....	48
<i>Федоров А. В., Потехин В. В.</i> Разработка интеллектуального алгоритма обработки изображений узи в биомедицине.....	50
Секция Программная инженерия: инструментальные средства и технологии проектирования и разработки.....	53
<i>Борисов Н.Д., Никифоров И.В.</i> Генерация пользовательского представления на основе XML-спецификаций для технологии SAP DYNPAGE.....	53
<i>Полубелова М.И., Григорьев С.В.</i> Генератор абстрактных лексических анализаторов ...	54
<i>Хаберланд Р., Ивановский С.А.</i> Верификация объектно-ориентированных программ, использующих динамическую память, с применением языка логического программирования.....	56
<i>Голубев Н.С., Селин И.А., Дробинцев П.Д.</i> Адаптация сетевых приложений для работы в сети с ячеистой топологией.....	57
<i>Янкович А.Д., Никифоров И.В., Воинов Н.В.</i> Инструментальная поддержка редактирования параметров конкретизации символьных тестовых сценариев.....	58
<i>Крупный Н. В. , Назарьев Е. П. , Веселов А. О. , Тютин Б. В.</i> Пример оптимизации тестирования при внедрении непрерывной интеграции.....	60
<i>Мухин В. А., Сафонов В. О.</i> Использование С# в качестве скриптового языка.....	61
<i>Маслаков А.П., Никифоров И.В., Котлярова Л.П.</i> Поиск и исправление ошибок при формализации требований.....	62
<i>Крупный Н.В., Назарьев Е.П., Веселов А.О., Тютин Б.В.</i> Автоматизация регрессионного тестирования на примере системы ТАТ.....	63
<i>Маслаков А.П., Никифоров И.В., Котляров В.П.</i> Автоматизация этапов проектирования тестовых сценариев.....	67
<i>Иванов Д. В., Москвин Д. А.</i> Исследование и анализ механизма разрешений операционной системы Google Android с точки зрения обеспечения безопасности системы.....	70
<i>Кухта А. Ю., Платонов В. В.</i> Обнаружение вторжений с помощью генетических алгоритмов.....	72
<i>Опарин П.А., Дробинцев П.Д., Котлярова Л.П.</i> Разработка методики обучения операционным системам реального времени с использованием графических языков моделирования.....	74
<i>Kashcheev A. Y., Fyodorov S. A., Lebedev L. K.</i> Graphics processing units comparative testing.....	77
<i>Garkusha N. V., Fyodorov S. A.</i> GObject Fortran object system for language interoperability	80
<i>Волкова А. С., Калинин М. О.</i> Разработка технологии планирования вычислительных процессов при организации высокопроизводительных конвейерно-параллельных вычислений в облачных средах.....	85
Секция Программная инженерия: методы и алгоритмы теории программирования.....	89
<i>Виденеева М. В., Соловьев И.П.</i> Сравнение фрактальных характеристик изображений и их вейвлет-образов в задаче классификации изображений.....	89
<i>Гошев В.А., Косовский Н.К.</i> Рефал-5Е: Разработка языка и реализация транслятора.....	91
<i>Жигулин А.Ю., Сафонов В. О.</i> Алгоритм шпионажа для компьютерных игр с многочисленными соперниками.....	92
<i>Кретов А.К., Сафонов В. О.</i> Система автоматического встраивания средств криптографической защиты в бинарные файлы приложений для платформы iOS.....	93

<i>Павлов С.Н., Петров А.Г.</i> Изучение методов регистрации соответствий для выравнивания 3D-моделей.....	94
<i>Пилипенко А. В. Сафонов В. О.</i> Анализ достижимости методов с помощью межпроцедурного анализа потока данных	95
<i>Игумнов А.В., Сараджиивили С.Э.</i> Сохранение работоспособности многоагентных систем.....	97
<i>Янкович А.Д., Никифоров И.В., Котляров В.П.</i> Библиотека анализа потоков UCM диаграмм	100
<i>Коноплев А. С., Калинин М. О.</i> Разработка методов и средств контроля и управления доступом для распределённых вычислительных сетей	104
<i>Mishchenkov A. V., Fyodorov S. A., Semenov A. A.</i> The search of programs areas with intensive irregular memory access by computing the metrics of spatial and temporal locality by the route of access addresses.....	106
<i>Короткова Е. И, Самочадин А. В.</i> Средства распознавания существенных изменений на аэрофотоснимках	108
<i>Морозов Р. А., Трифонов П. В.</i> Укорочение полярных кодов.....	110
<i>Моногаров К. Е.</i> Метод определения ориентации объектов на основе EM-алгоритма ..	114
<i>Павленко Е. Ю., Калинин М. О.</i> Разработка сервиса разрешения конфликтов в программно-конфигурируемых сетях.....	117
Секция Подходы к разработке ПО на основе технологий EMC.....	121
<i>Берленко Т.А., Фертиков К.А., Грозин В.А., Кринкин К.В.</i> Разработка механизма удержания шлюза в ядре LINUX для систем хранения данных.....	121
<i>Гиндин С.И.</i> Моделирование облачных систем с помощью многоканальных систем массового обслуживания с разогревом.....	122
<i>Закирова И. И. , Осенняя Д. И. , Александрова О. В. , Дробинцев П. Д.</i> Разработка программного продукта для моделирования систем хранения данных	123
<i>Калашиников Д. А. , Суворов А. А.</i> Исследование и разработка метода выбора размера слота в кэше на основе параметров рабочей нагрузки.....	125
<i>Галинский Р.Б., Шалыто А.А.</i> Рандомизированное автоматическое blackbox тестирование на основе строго заданной модели системы.....	127
<i>Скороспелов Л. А., Раздобреев Н.С., Заславский М. М., Кринкин К.В.</i> Инструментальные средства исследования кодовой базы больших программных проектов	128
<i>Белокопытова Е. А., Фёдоров С. А.</i> Оптимальное построение и конфигурация распределённых центров обработки данных	129
<i>Зобнин С. С., Поляков В. Б.</i> Реализация высокопроизводительных интеллектуальных систем хранения и обработки временных рядов.....	130
<i>Голощанов Е. С., Потехин В. В.</i> Оптимальное хранение временных рядов	133

