Секция 7

«НАУЧНО-МЕТОДИЧЕСКИЕ ПРОБЛЕМЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ В ОБЛАСТИ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

СОДЕРЖАНИЕ

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ СЕТЕВОЙ СТЕНД КАК МНОГОФУНКЦИОНАЛЬНЫЙ КОМПЛЕКС СРЕДСТВ ИЗУЧЕНИЯ СЕТЕВЫХ МЕТОДОВ ЗАЩИТЫ ИНФОРМАЦИИ

Абрамова Т.В., Аралбаев Т.З., д-р техн. наук, профессор,
Каменева Е.В., Синицын Ю.И., канд. техн. наук, доцент
Оренбургский государственный университет

Одним из важных факторов успешного проведения учебного процесса является материально-техническое обеспечение изучаемых дисциплин. Настоящая работа посвящена результатам построения учебно-лабораторного комплекса (УЛК) для изучения методов и средств защиты информации в компьютерных сетях, в частности, для изучения аномалий сетевых трафиков.

По результатам исследования Лабораторией Касперского в 2015 году 17% российских компаний столкнулись с DDos-атаками, а сама страна оказалась в первой пятерке государств, чьи веб-ресурсы вызывали наибольший интерес у киберзлоумышленников. По данным компаний Qrator и Wallarm, в 2014 году среднее число зловредных запросов на одного клиента увеличилось в 2.5 раза, при этом для одной атаки, как правило используются запросы с группы IP-адресов [6]. Таким образом, рост количества сетевых угроз определяет необходимость подготовки квалифицированных специалистов в области защиты от сетевых атак, в частности специалистов, занимающихся вопросами анализа сетевого трафика на наличие аномалий.

Следует отметить, данная задача достаточно успешно решается в центрах специальной подготовки и переподготовки специалистов в области сетевых технологий, например, в центрах обучения академии Cisco. Однако в условиях учебного процесса вузов имеются ряд сложностей организационного, методического и финансового характера, преодоление которых возможно лишь на основе системного подхода, учитывающего специфику учебных стандартов, перечня компетенций, дидактического материала, существующей лабораторной базы, квалификационных характеристик разработчиков и других факторов.

Представленный учебно-лабораторный комплекс разработан на кафедре вычислительной техники защиты информации Оренбургского государственного университета на базе кафедральных компьютерных лабораторий и учебно-исследовательского сетевого стенда, установленного в одной их лабораторий. Разработке УЛК предшествовал ряд исследований, в ходе которых проведен аналитический обзор публикаций, посвященных структурным, архитектурным и функциональным особенностям УЛК, определена концепция его построения, выявлены требования и принципы его разработки. В частности, анализ публикаций [1; 2; 5] позволил определить концепцию построения структуры и выбор архитектуры УЛК.

Перечень работ [1; 4] позволил сформировать систему дидактического материала, определить структуру лабораторных работ и порядок их проведе-

ния. На основе анализ работ [3-5] определены функциональные требования к УЛК и особенности его реализации.

Известно, что создание учебно-лабораторного комплекса, как системы, является достаточно сложной, многовариантной задачей, решение которой достигается различными подходами, в частности: на основе методов целочисленного линейного программирования, морфологического анализа, экспертной оценки и других. В данном случае УЛК формировался на базе существующих средств кафедры вычислительной техники и защиты информации. Поэтому основные требования к нему определялись выбором тематики учебно-методического материала по защите информации в компьютерных сетях. Для этого был проведен анализ существующих комплексов лабораторных работ, позволивший определить структуру лабораторных работ и перечень дидактических единиц для изучения.

В указаниях для специализированных курсов имеется свой ряд недостатков, в частности: в частности, недостаточная функциональная полнота, требования повышенной первоначальной подготовки. Целью разработки представленного в работе УЛК является устранение перечисленных недостатков.

В основу концепции разработки положен принцип представления УЛК как учебно-методической системы для изучения аномалий сетевых трафиков [7]. В соответствии с этим:

- УЛК построен с учетом компетенций и дидактического материала специальности «Комплексная защита объектов информатизации» (КЗОИ);
- использован критерий обеспечения функциональной полноты для решения учебно-исследовательских задач мониторинга и отражения сетевых вторжений;
- оптимизация технических и функциональных характеристик УЛК проведена в условиях стоимостных и временных ограничений вуза.

При реализации концепции были разработаны и использованы: реляционная модель выбора и обоснования перечня дидактических единиц и модулей, реляционная модель выбора аппаратно-программных средств и лабораторно-стендового оборудования. Применение данных моделей позволили оптимизировать структуру и качественный состав обеспечивающих подсистем УЛК.

Структура УЛК представляет собой кафедральную локальную вычислительную сеть из 24 компьютерных станций, включающую в себя сетевые узлы, размещенные в стендовой стойке, имеющие возможность подключения к Интернету как посредством средств беспроводной связи, так и через прокси-сервер вуза.

Архитектурно УЛК реализована на базе стационарных компьютерных станций, мобильных компьютеров (ноутбуков), обеспеченных проводной и беспроводной связью. Коммутаторы, маршрутизаторы и межсетевой экран стенда выбраны по принципу доступности из линейки сетевого оборудования фирм DLink и Cisco.

На рисунке 1 представлена стойка и основное сетевое оборудование учебно-исследовательского стенда, используемого в составе УЛК для проведения лабораторных работ по сетевым технологиям.

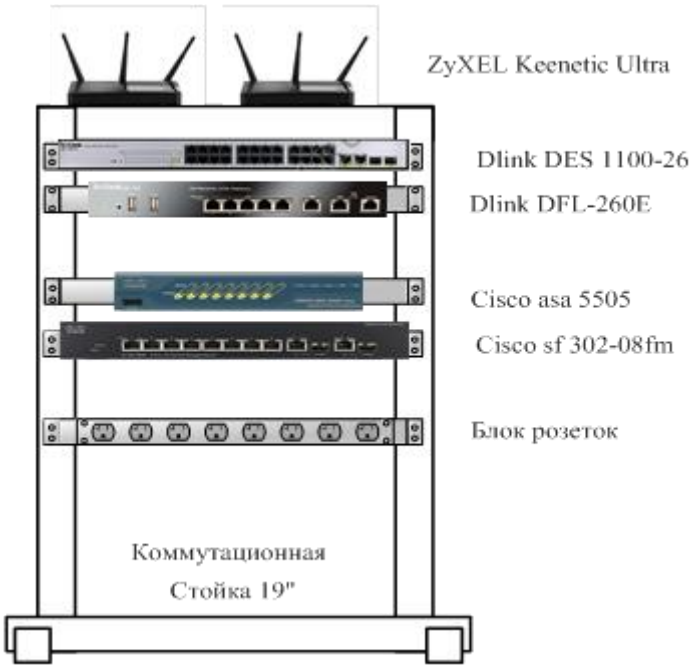

Рисунок 1 - Внешний вид учебно-исследовательского сетевого стенда

В коммутационную стойку установлено следующее сетевое оборудование:

- настраиваемый коммутатор DLink DES-1100-26;
- межсетевой экран DLink DFL-260E;
- межсетевой экран Cisco asa 5505;
- управляемый коммутатор Cisco sf302-08mp;
- беспроводной маршрутизатор ZyXEL Keenetic Ultra.

В перечень программно-аппаратных и программных средств УЛК включены:

- программно-аппаратный комплекс защиты информации от несанкционированного доступа «Аккорд»;
- персональное средство криптографической защиты информации «ШИПКА»;
- сканер сетевого трафика “Wireshark”;
- программное средство защиты информации от несанкционированного доступа “Secret Net 7”;
- сетевой сканер уязвимостей хостов “XSpider 7.7”;
- программное средство подбора паролей архивов “Advanced Archive Password Recovery 4.54.55” и ряд других программ.

Представленный УЛК обеспечивает выполнение лабораторных работ практически по всем сетевым дисциплинам специальности КЗОИ, в перечень которых входят: «Комплексная защита информации в распределенных вычислительных системах», «Сети и системы», «Программно-аппаратные средства защиты информации», «Защита информационных процессов в компьютерных системах».

Технические и функциональные характеристики УЛК позволили разработать и применить в учебном процессе комплекс из 12 лабораторных работ, тематика которых включает вопросы построения и исследования компьютерных сетей различной конфигурации, установки, настройки и исследования сетевого оборудования и программного обеспечения, в частности:

- построение локальных вычислительных сетей на базе учебно-исследовательского сетевого стенда;
- построение и настройка защищенной беспроводной сети;
- безопасное администрирование сетей;
- создание и противодействие простой сетевой атаке;
- анализ трафиков и уязвимостей в сетях с использованием программных средств;
- создание и противодействие DDos-атакам;
- моделирование игровых ситуаций по получению доступа к сетевому ресурсу;
- дистанционное управление и сканирование уязвимостей;
- оперативная идентификация аномалий сетевого трафика и определение мер по их нейтрализации.

Учебно-лабораторный комплекс применяется для физического и имитационного моделирования компьютерных сетей с различной технологией и для реализации различных аномальных ситуаций, необходимых для генерации и регистрации данных в научно-исследовательских разработках студентов и аспирантов.

В частности, на базе УЛК успешно выполнены следующие научно-исследовательские разработки студентов:

- оперативный поиск информации в базах данных сетевого трафика на основе ассоциативного подхода;
- моделирование сетевого трафика и обнаружение аномалий на основе методов нейронных сетей, классификатора Байеса и спектрального анализа;
- дистанционное управление и сканирование уязвимостей в компьютерной сети удаленных автоматизированных систем.

УЛК успешно использован на курсах повышения квалификации по основам информационной безопасности преподавательского состава кафедры.

Достоинства работы сетевого стенда наглядно видны при проведении лабораторных работ среди студентов и магистрантов кафедры. В качестве примера можно рассмотреть эксперимент по организации атаки ping flooding [8] и сбору экспериментальных данных для анализа активности сетевого трафика,

наглядно показывающих студентам возможности сбора и обработки данных при выявлении сетевых атак. Структурная схема сети при проведении атаки представлена на рисунке 2.

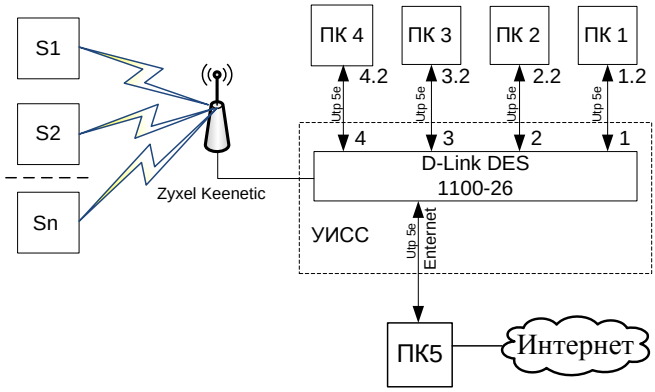

Рисунок 2 – Структурная схема построения сети

На рисунке приняты следующие условные обозначения: ПК1 – ПК2 – компьютеры лаборатории, подключенные посредством проводной связи к коммутатору D-Link, ПК5 – компьютер сети, выполняющий функции DHCP-сервера, Zyxel Keenetic – маршрутизатор, S1 – Sn – компьютеры сети, участвовавшие в атаке, подключенные к сети посредством беспроводной связи.

В ходе моделирования атаки, атакующие были разбиты на 6 групп. С периодичностью, кратной 2, первая группа атаковала каждые 2^3=8 секунд, вторая группа каждые 2^4=16 секунд, третья группа каждые 2^5=32 секунды, четвертая каждые 2^6=64 секунды, пятая каждые 2^7=128 секунд, шестая каждые 2^8=256 секунд. Общее время атаки – 300 секунд. Временная диаграмма режима атаки представлена на рисунке 3.

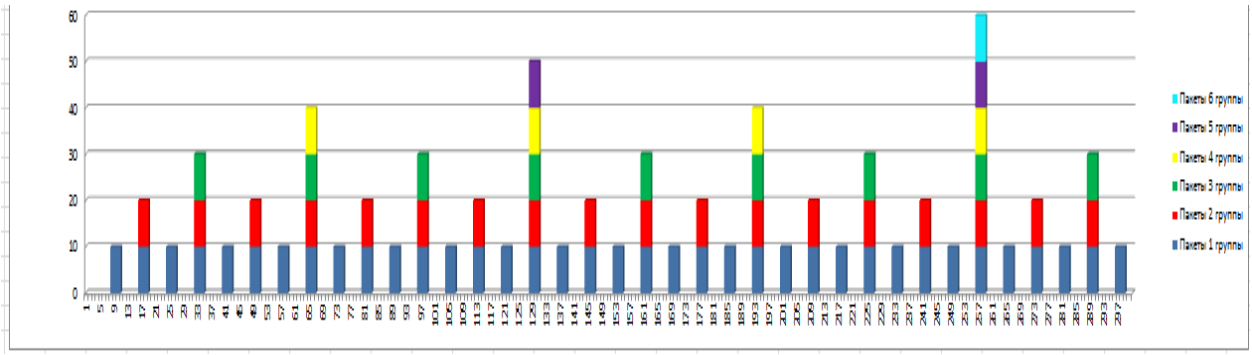

Рисунок 3 – Временная диаграмма экспериментального сетевого трафика в режиме атаки

После проведения атаки с помощью табличного процессора Microsoft Excel была произведена обработка данных и построен график экспериментального сетевого трафика по состоянию сети в нормальном режиме и в режиме

атаки. При анализе графика интенсивности сетевой активности в режиме атаки наблюдается определенная периодичность в активности пользователей сети, в соответствии с проведенными экспериментальными атаками. Рост скачков интенсивность в каждом следующем периоде связан с подключением к атаке новых групп атакующих.

Результаты представлены на рисунке 4.

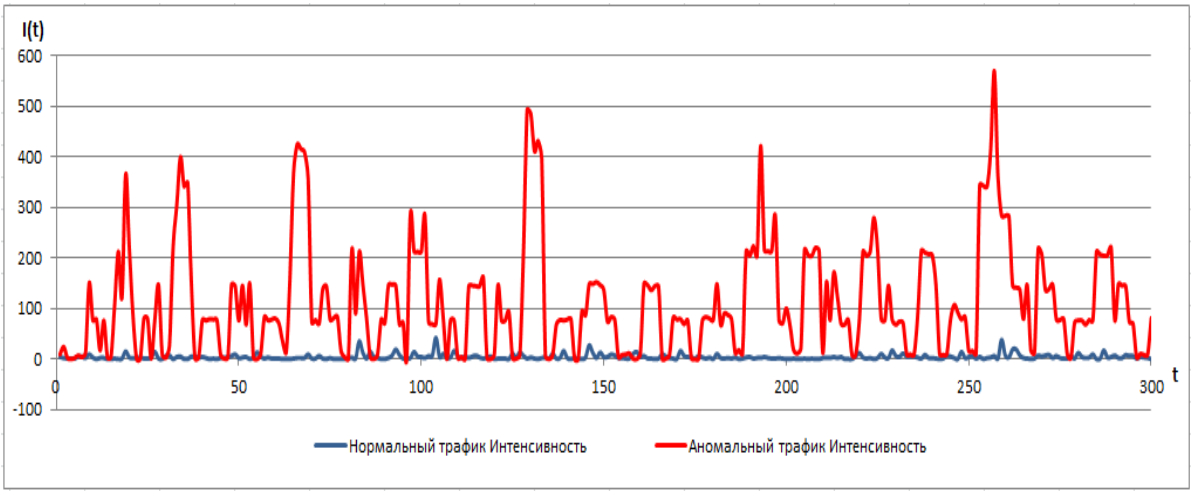

Рисунок 4 – Интенсивность сетевого трафика в режимах покоя и атаки

Полученные графики наглядно показывают, что показатель интенсивность сетевого трафика значительно меняется при смене вида деятельности пользователей. В штатном режиме сетевой трафик удовлетворяет стандартному шаблону. Сетевая активность пользователей не превышает 45 пакетов в секунду. На графике наблюдаются временные скачки интенсивности, связанные с работой сетевого оборудования. В режиме атаки, напротив, интенсивность трафика резко возрастает, вследствие перегрузки при обработке ICMP-сообщений сетевым оборудованием. При этом интенсивность сетевого трафика в режиме атаки в разы превышает интенсивность трафика в нормальном режиме. Это обусловлено высокой активностью пользователей сети и сетевого оборудования.

Полученные данные дают обучающимся наглядное представление о происходящих в сети процессах. Кроме того, их можно использовать для дальнейшего моделирования и прогнозирования сетевых процессов, для генерации и регистрации данных в научно-исследовательских разработках студентов и аспирантов. Например, получив график интенсивности трафика при проведении сетевой атаки и построив линию тренда полученного графика, можно проводить дальнейший прогноз сетевой активности и надежности работы сети при проведении того или иного вида сетевой атаки.

Результаты проведенной работы показывают, что использование сетевого стенда для имитации сетевой атаки дает наглядное представление о сетевых процессах, возможность проанализировать воздействие атаки на сетевое обору-

дование, настроить оборудование для обеспечения информационной безопасности сети. Кроме того, подобный подход способен привести к повышению интереса студентов к учебному процессу и получению практических навыков и, в конечном счете, повышению качества подготовки будущих специалистов в области сетевой безопасности и сетевых технологий.

Список литературы

1. *Богданова Е.А., Руденков Н.А. и др. / Технологии защиты информации в компьютерных сетях. Межсетевые экраны и интернет- маршрутизаторы. 2013 г. – 743 с.*
2. *Гуц А.К., Вахний Т.В. Теория игр и защита компьютерных систем: учебное пособие / А.К. Гуц, Т.В. Вахний. – Омск: Изд-во ОМГУ, 2013. – 160 с.*
3. *Методические указания по выполнению лабораторных работ по дисциплине «Защита информации» – Утв. 2003 г. – Уфа: УГАТУ, 2003. – Режим доступа. – URL: http://www.studfiles.ru/dir/cat32/subj1166/file9309.html (Дата обращения 15.04.2015).*
4. *Основная образовательная программа высшего профессионального образования. Направление подготовки: 090900 – Информационная безопасность. Профиль подготовки – Комплексная защита объектов информатизации. Квалификация – Бакалавр. Форма обучения – Очная. – Утв. 2011-04-16. – Оренбург: ОГУ, 2011. – 43 с.*
5. *Смирнова Е.В., Пролетарский А.В. и др. / Построение коммутируемых компьютерных сетей: учебное пособие, 2012. – 367 с.*
6. *DDos-атаки 2014: реже, но крупнее – [Электронный ресурс] – / БЕСТСЕЛЕРЫ Аналитического рынка ИТ. – Режим доступа. – URL: http://www.itbestsellers.ru/companies-analytics/detail.php?ID=30073 (Дата обращения 15.04.2015).*
7. *Аралбаев Т.З., Романенко С.Ю. УЧЕБНО-ЛАБОРАТОРНЫЙ КОМПЛЕКС ДЛЯ ИЗУЧЕНИЯ АНОМАЛИЙ СЕТЕВЫХ ТРАФИКОВ // Технические науки - от теории к практике: сб. ст. по матер. LVI междунар. науч.-практ. конф. № 3(51). – Новосибирск: СибАК, 2016. – С. 17-24.*
8. *Ping-флуд– [Электронный ресурс] — Режим доступа. – URL: https://ru.wikipedia.org/wiki/Ping-D1%84%D0%BB%D1%83%D0%B4 (Дата обращения 11.01.2017).*

ПРОБЛЕМА ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРЕДПРИЯТИЯ

Белова Т.А.
Оренбургский государственный университет

В условиях развития информационных технологий одним из важнейших приоритетов предприятия является информационная безопасность, поскольку последствия и возможный ущерб из-за нарушения безопасности информационной системы могут привести к высоким убыткам организации.

На сегодняшний день не существует единой количественной методики расчета величин рисков, измеряемой в стоимостной оценке. В первую очередь это связано с отсутствием необходимого объема статистических данных о вероятности возникновения угрозы. Вторая причина заключается в том, что современные методики основаны на опыте иностранных компаний и поэтому в российских реалиях их применение сопряжено с определенными трудностями. Поэтому более часто используются качественные или смешанные методики оценки рисков.

Еще одной проблемой является децентрализация процесса оценки рисков. В следствие этого исключается возможность реализации единого подхода к управлению рисками в организации.

При рассмотрении оценки рисков информационной безопасности в общем виде следует выделить основные функциональные блоки системы экономической безопасности предприятия, обеспечивающие максимальное соответствие менеджмента предприятия и его ресурсного потенциала:

– имущество (активы) предприятия;
– финансы предприятия;
– кадры предприятия;
– технологии и инновации;
– информационная система предприятия;
– организационная структура предприятия.

Данная структура функциональных составляющих соответствует структуре механизма обеспечения экономической безопасности предприятия и затрагивает все функциональные области деятельности предприятия.

Информационная система предприятия, как правило, охватывает все сферы его деятельности: административную, производственную, финансовую, выступает как связующее звено при выработке стратегии бизнеса и качества управления предприятием и персоналом. Однако, из-за сложности оценки зачастую не рассматриваются такие типы объектов защиты как сервисы, нематериальные ресурсы, люди, их квалификация, навыки и опыт.

Рассматривая информационную систему в ее исходном состоянии, мы оцениваем размер ожидаемых потерь от инцидентов, связанных с информационной безопасностью. После этого, делается оценка того, как предлагаемые

средства и меры обеспечения безопасности влияют на снижение рисков, и сколько они стоят. Если представить некоторую идеальную ситуацию, то идею подхода отображает приведенный ниже график (рисунок 1).

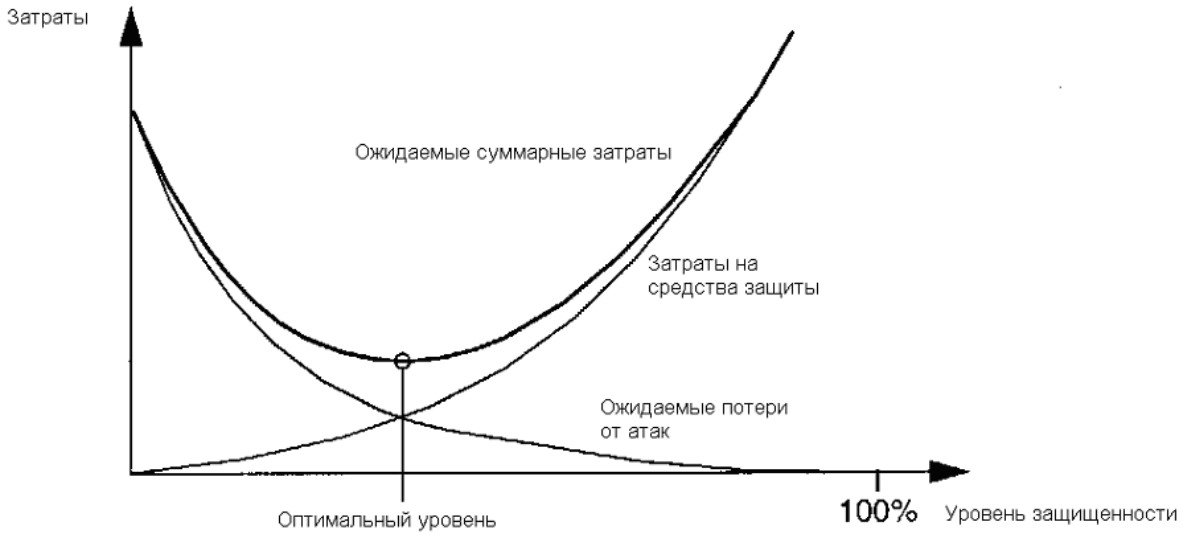

Рисунок 1 – Идеализированный график соотношения
«затраты на защиту/ожидаемые потери»

По мере того, как затраты на защиту растут, размер ожидаемых потерь падает. Если обе функции имеют вид, представленный на рисунке, то можно определить минимум функции «Ожидаемые суммарные результаты», который нам и требуется.

К сожалению, на практике точные зависимости между затратами и уровнем защищенности определить не представляется возможным, поэтому аналитический метод определения минимальных затрат в представленном виде неприменим.

На современном этапе существуют специальные методики и системы анализа рисков, но они не позволяют провести комплексный анализ, решая лишь частные задачи. Среди распространенных методик принята классификация:

– методики, использующие оценку риска на качественном уровне (например, по шкале «высокий», «средний», «низкий»). К таким методикам, в частности, относится FRAP;

– количественные методики (риск оценивается через числовое значение, например размер ожидаемых годовых потерь). К этому классу относится методика RiskWatch;

– методики, использующие смешанные оценки (такой подход используется в CRAMM, методике Microsoft и т.д.).

Проведем сравнительный анализ некоторых методик.

Среди преимуществ методики FRAP можно выделить более подробное раскрытие путей получения данных о системе и ее уязвимостях. Однако, при проведении анализа, как правило, принимают, что изначально в системе отсутствуют средства и механизмы защиты. Таким образом оценивается уровень риска для незащищенной ИС.

Оценка RiskWatch имеет сравнительно небольшую трудоемкость работ по анализу рисков с использованием этого метода. Существенным достоинством RiskWatch является интуитивно понятный интерфейс и большая гибкость метода, обеспечиваемая возможностью введения новых категорий, описаний, вопросов и т. д. Он подходит, если требуется провести анализ рисков на программно-техническом уровне защиты без учета организационных и административных факторов.

В отличие от других методик анализа рисков, ГРИФ предлагает все способы снижения рисков (обход, снижение и принятие). Данная методика учитывает сопроводительную документацию, такую как описание бизнес-процессов или отчетов по проведенным оценкам рисков ИБ.

Положительной стороной CORAS является то, что программный продукт, реализующий эту методику, распространяется бесплатно и не требует значительных ресурсов для установки и применения. Однако, CORAS не предусматривает такой эффективной меры по управлению рисками, как «Программа повышения информированности сотрудников в области информационной безопасности».

Ключевыми показателями при оценке MSAT являются: профиль риска для бизнеса (величина изменения риска в зависимости от бизнес-среды, действительно, важный параметр, который не всегда учитывается при оценки уровня защищенности системы в организациях разных сфер деятельности) и индекс эшелонированной защиты (сводная величина уровня защищенности). MSAT не дает количественной оценки уровня рисков, однако, качественные оценки могут быть привязаны к ранговой шкале. MSAT позволяет оценить эффективность инвестиций, вложенных во внедрение мер безопасности, но не дает возможности найти оптимальный баланс между мерами, направленными на предотвращение, выявление, исправление или восстановление информационных активов.

Методика COBRA позволяет выполнить в автоматизированном режиме простейший вариант оценивания информационных рисков любой компании. Представляет требования стандарта ISO 17799 в виде тематических вопросников, на которые следует ответить в ходе оценки рисков информационных активов и электронных бизнес-транзакций компании. Далее введенные ответы автоматически обрабатываются, и с помощью соответствующих правил логического вывода формируется итоговый отчет с текущими оценками информационных рисков компании и рекомендациями по их управлению. Данная методика является качественной и поэтому дать интерпретацию полученных результатов не всегда возможно.

Особенность методики OCTAVE заключается в том, что весь процесс анализа производится силами сотрудников организации, без привлечения внешних консультантов. Для этого создается смешанная группа, включающая как технических специалистов, так и руководителей разного уровня, что позволяет всесторонне оценить последствия для бизнеса возможных инцидентов в области безопасности и разработать контрмеры.

Компания MethodWare разработала свою собственную методику оценки и управления рисками и выпустила ряд соответствующих инструментальных средств. Risk Advisor, являющийся одним из ПО данной компании, позиционируется как инструментарий аналитика или менеджера в области информационной безопасности. Данная методика позволяет автоматизировать различные аспекты управления рисками компании. При этом оценки рисков даются в качественных шкалах. Подробный анализ факторов рисков не предусмотрен. Сильной стороной рассмотренной методики является возможность описания различных связей, адекватный учет многих факторов риска и существенно меньшая трудоемкость.

Таким образом, система оценки рисков информационной безопасности предприятия должна полно и всесторонне охватывать современные требования к осуществлению анализа рисков.

Методика, разрабатываемая нами, будет централизованно оценивать величину риска при изначально любом уровне защищенности информационной системы. Система будет давать количественную оценку уровня рисков и оптимальный баланс между затратами на защиту и ожидаемыми потерями. Кроме того, она всесторонне будет анализировать факторы рисков. Это особенно важно в тех случаях, когда к информационной системе компании предъявляются повышенные требования в области защиты информации и непрерывности бизнеса.

Список литературы

1 Баранова, Е. К. Информационная безопасность и защита информации : учеб. пособие / Е. К. Баранова, А. В. Бабаш – Москва: ИНФРА-М: РИОР – 2017. – 322 с.

2 Плетнев, П. В. Методика оценки рисков информационной безопасности на предприятиях малого и среднего бизнеса / П. В. Плетнев, В. М. Белов // Доклады ТУСУРа – 2012. – №1 – С. 83-86.

3 Семкина, А. А. Оценка уровня информационной безопасности предприятия через остаточный риск / А. А. Семкина, А. М. Цыбулин // Вестник ВолГУ – 2012. – №6 – С. 156-158.

4 Кузнецова, О. Б. Оценка информационных рисков в обеспечении экономической безопасности предприятия / О. Б. Кузнецова // Труды ИСА РАН – 2007. – Т. 31 – С. 31-98.

5 Баранова, Е. Анализ рисков информационной безопасности для малого и среднего бизнеса / Е. Баранова, А. Мальцева // Директор по безопасности – 2015. – С. 58-63.

ЭЛЕМЕНТЫ АЛГЕБРАИЧЕСКОГО КРИПТОАНАЛИЗА В УЧЕБНЫХ ДИСЦИПЛИНАХ ПО КРИПТОГРАФИЧЕСКИМ МЕТОДАМ ЗАЩИТЫ ИНФОРМАЦИИ

Благовисная А.Н., Михляева А.В.
Оренбургский государственный университет

В процессе преподавания дисциплин, посвященных криптографическим методам защиты информации, довольно часто вместе с изучением методов создания криптоалгоритмов рассматриваются и примеры раскрытия шифров. Как правило, это задачи, использующие методы раскрытия традиционных (исторических) шифров, основаные на идеях перебора, частотного анализа. Практика преподавания дисциплин, связанных с математическими методами защиты информации, показывает, что данные методы криптоанализа доступны для понимания и реализации студентами практически любого уровня подготовки. Следует отметить, что решение задач на раскрытие исторических шифров вызывает интерес у студентов не только к учебной дисциплине, но и к криптографии как науке. В связи с этим возникает вопрос: а возможно ли на доступном для студентов уровне знакомство с методами криптоанализа, позволяющими раскрывать современные криптографические конструкции? На наш взгляд, такое знакомство возможно на уровне изучения основных идей современных разделов криптоанализа на примерах решения учебных задач, подразумевающих раскрытие упрощенных моделей криптографических конструкций.

Рассмотрим данный подход на примере решения задач, демонстрирующих основные идеи алгебраического криптоанализа.

Алгебраический криптоанализ является сравнительно новым методом раскрытия шифров. В 2003 году появилась атака на фильтрующие генераторы [9], которая получила название алгебраической. Позднее было показано применение алгебраической атаки на комбинирующие генераторы и блочные шифры. В настоящее время методы алгебраического криптоанализа активно применяются для исследования поточных [7, 8], блочных [2, 3] алгоритмов шифрования. Развиваются и методы решения систем булевых уравнений [1], составляющих основу алгоритмов алгебраического криптоанализа.

В некоторых учебных изданий [5, 6], предназначенных для студентов вузов, встречаются разделы, посвященные криптографическим свойствам, которыми должны обладать булевы функции для обеспечении стойкости шифров к алгебраическим криптоатакам. В этих же книгах можно найти и формулировки заданий на раскрытие шифров методами алгебраического криптоанализа.

Основная идея алгебраического криптоанализа заключается в составлении системы булевых уравнений, которые описывают преобразование шифра. Такая система строится на основе полностью известного алгоритма шифрования. Особенностью системы булевых уравнений, которая возникает при

криптоанализе, является её непротиворечивость, то есть система имеет как минимум одно решение. Рассмотрим, как получаются такие системы булевых уравнения с позиции формулировок и решения учебных задач.

Покажем алгебраическую атаку на генератор с регистром длины 4, уравнение рекурсии которого $u(i+4)=u(i+1)\oplus u(i)$, фильтрующая функция $f(x_1,x_2,x_3,x_4)=x_1x_2x_3\oplus x_1x_2\oplus x_1x_4\oplus x_3x_4\oplus x_3,$ если известен начальный отрезок гаммы $\gamma=(1,0,0,1)$.

Нам необходимо найти начальное состояние генератора. Пусть u_0, u_1, u_2, u_3 – элементы искомой последовательности, задающей начальное состояние генератора. Составим систему уравнений на основе фильтрующей функции:

$$\begin{cases} f(u_0,u_1,u_2,u_3)=u_0u_1u_2\oplus u_0u_1\oplus u_0u_3\oplus u_2u_3\oplus u_2=1,\\ f(u_1,u_2,u_3,u_4)=u_1u_2u_3\oplus u_1u_2\oplus u_1u_4\oplus u_3u_4\oplus u_3=0,\\ f(u_2,u_3,u_4,u_5)=u_2u_3u_4\oplus u_2u_3\oplus u_2u_5\oplus u_4u_5\oplus u_4=0,\\ f(u_3,u_4,u_5,u_6)=u_3u_4u_5\oplus u_3u_4\oplus u_3u_5\oplus u_5u_6\oplus u_5=1. \end{cases} \quad (1)$$

Используя уравнение рекурсии, выразим u_4, u_5, u_6:

$$u_4=u_0+u_1,\ u_5=u_1+u_2,\ u_6=u_2+u_3. \quad (2)$$

Найденные выражения (2) подставим в систему (1). После преобразований уравнений системы, заключающихся в раскрытии скобок и приведении подобных, получим следующую систему:

$$\begin{cases} u_0u_1u_2\oplus u_0u_1\oplus u_0u_3\oplus u_2u_3\oplus u_2=1,\\ u_1u_2u_3\oplus u_0u_1\oplus u_0u_3\oplus u_1u_2\oplus u_1u_3\oplus u_1\oplus u_3=0,\\ u_0u_2u_3\oplus u_1u_2u_3\oplus u_0u_1\oplus u_0u_2\oplus u_2u_3\oplus u_0\oplus u_2=0,\\ u_0u_1u_3\oplus u_0u_2u_3\oplus u_1u_2u_3\oplus u_0u_3\oplus u_1u_2\oplus u_1=1. \end{cases} \quad (3)$$

Количество уравнений и переменых в получившейся системе (3) не так велико, поэтому решение системы можно найти, перебирая все возможные наборы значений переменных. Перебор различных вариантов решений удобно офомить в таблице (таблица 1).

В результате подбора решений системы (3) удалось установить, что непротиворечивыми все уравнения системы являются лишь в одном случае, когда $u_0=1, u_1=1, u_2=0, u_3=0$. Это решение является единственным решением системы (3), то есть искомое начальное состояние генератора представляется в виде (1,1,0,0).

Таблица 1 – Подбор решений системы (3)

u_0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
u_1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
u_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
u_3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
u_0u_1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
u_0u_2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
u_0u_3	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
u_1u_2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
u_1u_3	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
u_2u_3	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$u_0u_1u_2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$u_0u_1u_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
$u_0u_2u_3$	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
$u_1u_2u_3$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
1 ур-ие	0≠1	0≠1		0≠1	0≠1	0≠1		0≠1	0≠1					0≠1		
2 ур-ие		1≠0		1≠0	1≠0	1≠0		1≠0						1≠0		1≠0
3 ур-ие			1≠0				1≠0	1≠0	1≠0	1≠0	1≠0	1≠0			1≠0	1≠0
4 ур-ие	0≠1		0≠1				0≠1		0≠1	0≠1	0≠1					0≠1
Вывод	–	–	–	–	–	–	–	–	–	–	–	–	+	–	–	–

Перебор всех возможных вариантов решений получающейся при алгебраической криптоатаке системы булевых уравнений не является рациональным методом поиска решений систем. На практике используются другие подходы к поиску неизвестных, удовлетворяющих уравнениям системы. Один из таких подходов основан на понятии аннигиляторов булевой функции.

Определение. Булева функция g, не равная тождественно нулю, минимальной степени, такая, что $fg = 0$ или $(f+1)g = 0$, называется аннигилятором булевой функции f.

В качестве примера рассмотрим алгебраическую атаку на генератор с регистром длины 3, с уравнением рекурсии $u(i+3) = u(i+1) \oplus u(i)$ и фильтрующей

функцией $f(x_1, x_2, x_3) = x_1x_2x_3 \oplus x_2x_3 \oplus x_2 \oplus x_3$. Пусть известен отрезок гаммы $\gamma = (1,1,0)$.

Система уравнений для поиска начального состояния генератора имеет вид:

$$\begin{cases} f(u_0, u_1, u_2) = u_0u_1u_2 \oplus u_1u_2 \oplus u_1 \oplus u_2 = 1, \\ f(u_1, u_2, u_3) = u_1u_2u_3 \oplus u_2u_3 \oplus u_2 \oplus u_3 = 1, \\ f(u_2, u_3, u_4) = u_2u_3u_4 \oplus u_3u_4 \oplus u_3 \oplus u_4 = 0. \end{cases} \quad (4)$$

С учетом того, что $u_3 = u_0 + u_1$, $u_4 = u_1 + u_2$, система (4) примет следующий вид:

$$\begin{cases} u_0u_1u_2 \oplus u_1u_2 \oplus u_1 \oplus u_2 = 1, \\ u_1u_2(u_0 \oplus u_1) \oplus u_2(u_0 \oplus u_1) \oplus u_2 \oplus (u_0 \oplus u_1) = 1, \\ u_2(u_0 \oplus u_1)(u_1 \oplus u_2) \oplus (u_0 \oplus u_1)(u_1 \oplus u_2) \oplus (u_0 \oplus u_1) \oplus (u_1 \oplus u_2) = 0. \end{cases} \quad (5)$$

Далее упростим систему (5). Для этого потребуется найти аннигиляторы для функций f и $f \oplus 1$ (для нашего примера $f(x_1, x_2, x_3) = x_1x_2x_3 \oplus x_2x_3 \oplus x_2 \oplus x_3$). Для поиска аннигиляторов булевых функций нами в среде Visual Studio на языке объектно-ориентированного программирования C++ написано программное средство. В программе реализованы возможности выбора количества переменных булевой функции, ввода булевой функции, предусмотрена обработка исключительных ситуаций. Программа нахождения аннигиляторов булевой функции реализована в соответствии с алгоритмом, рассмотренным в работе [4] и предназначена для работы с учебными задачами.

Аннигиляторами для функций f и $f \oplus 1$ являются функции $g_1 = x_2x_3 \oplus x_2 \oplus x_3 \oplus 1$ и $g_2 = x_2 \oplus x_3$соответственно. Далее, в соответствии со схемой алгебраической атаки, первые два уравнения системы следует заменить на уравнения $g_1(u_0, u_1, u_2) = 0$ и $g_1(u_1, u_2, u_3) = 0$, а третье – на $g_2(u_2, u_3, u_4) = 0$. В результате получаем систему

$$\begin{cases} u_1u_2 \oplus u_1 \oplus u_2 \oplus 1 = 0, \\ u_2(u_0 \oplus u_1) \oplus u_2 \oplus (u_0 \oplus u_1) \oplus 1 = 0, \\ u_0 \oplus u_2 = 0. \end{cases} \quad (6)$$

Будем искать решения системы, рассуждая следующим образом. Из последнего уравнения системы видно, что либо $u_0 = u_2 = 0$, либо $u_0 = u_2 = 1$. По-

этому нам не нужно рассматривать все возможные варианты решений, как это было сделано в предыдущем примере, а достаточно рассмотреть лишь варианты, при которых последнее уравнение имеет смысл. При $u_0 = 1, u_1 = 0, u_2 = 1$ противоречивых уравнений в системе (5) не будет, и мы получим единственное решение системы, которое дает начальное состояние генератора (1,0,1).

Таким образом, рассмотренные примеры раскрытия упрощенных криптографических конструкций можно использовать в качестве учебных задач, демонстрирующих идеи алгебраического криптоанализа.

Список использованных источников

1. Агибалов, Г.П. Методы решения систем полиномиальных уравнений над конечным полем / Г.П. Агибалов // Вестник Томского государственного университета. – 2006. – № 17. – С. 4-9.

2. Бабенко, Л.К. Анализ стойкости блочных алгоритмов шифрования к алгебраическим атакам / Л.К. Бабенко, Е.А. Маро // Известия ЮФУ. Технические науки. – 2011. – №12. – С.110-119.

3. Маро, Е. А. Алгебраический криптоанализ упрощённого алгоритма шифрования Rijndael / Е.А. Маро // Известия ЮФУ. Технические науки. – 2009. – № 11 (110). – С.187-199.

4. Отрыванкина, Т. М. Криптографические свойства булевых функций: методические указания / Т. М. Отрыванкина, А. Н. Благовисная; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2014. – 55 с.

5. Панкратова, И.А. Булевы функции в криптографии: учебное пособие / И. А. Панкратова. – Томск: Издательский Дом Томского государственного университета, 2014. – 88 с.

6. Токарева, Н. Н. Симметричная криптография. Краткий курс: учебное пособие / Н. Н. Токарева. – Н: Новосибирский государственный университет, 2012. – 232 с.

7. Хузина, Э.И. Модель связи в криптографии и алгебраические атаки на поточные шифры / Э.И. Хузина // Молодёжный научно-технический вестник. – 2013. – № 12. – С. 1-4.

8. Чиликов, А. А. Анализ поточных шифров с помощью решения системы алгебраических уравнений / А. А. Чиликов, Э. И. Хузина // Научное издание МГТУ им. Н. Э. Баумана. Наукаиобразование. – 2013. – № 3. – С. 257-268.

9. Courtois, N. Algebraic attack on stream ciphers with linear feedback / N. Courtois, W. Meier // LNCS. – 2003. – V. 2656. – P. 345–359.

О ПРОГРАММНЫХ РЕАЛИЗАЦИЯХ ТЕСТОВ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Самойлов Е.И., Благовисная А.Н.
Оренбургский государственный университет

Тестирование последовательностей псевдослучайных чисел, генерируемых с целью использования их в криптографических конструкциях, является одной из важных задач, возникающих при разработке систем обеспечения информационной безопасности. Исследования, имеющие целью совершенствование криптографических средств, являются актуальными и востребованными, поэтому их необходимо отражать в учебных курсах, посвященных криптографическим методам защиты информации.

В учебных дисциплинах, изучающих методы криптографии, вопросы осуществления тестирования псевдослучайных последовательностей можно рассматривать, решая учебные задачи, содержание которых заключается в изучении и применении существующих пакетов тестов к различным видам последовательностей. Особый интерес представляет тестирование последовательностей, выдаваемых генераторами, используемыми в реальных криптосистемах. В этом случае тестирование последовательностей невозможно без применения специальных программных средств, реализующих тестирование. Возникает проблема поиска и выбора готовых программных продуктов, реализующих оценку псевдослучайных последовательностей.

Статистические тесты обычно объединяются в пакеты, представляющие собой подборку различных методик по оценке качества псевдослучайных последовательностей методами статистического анализа. Наиболее распространенными и известными являются пакеты Diehard, TestU01, NIST, CRYPT-X, тесты Д. Кнута. Кроме того, существуют тесты, разрабатываемые исследователями специально для решения особых, специфичных задач, для которых применение стандартных пакетов некорректно или недостаточно для полной и всесторонней оценки качества генерируемых последовательностей.

Рассмотрим существующие статистические тесты псевдослучайных последовательностей с точки зрения их программных реализаций, которые могут быть использованы в учебном процессе.

Пакет Diehard представляет собой набор статистических тестов для измерения качества набора случайных чисел. Автором тестов является Джордж Марсалья, который разрабатывал их в течение нескольких лет и опубликовал в 1995 году на CD-ROM, посвящённом случайным числам. Содержимое CD-ROM находится в открытом доступе на сайте https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/. Он содержит 4,8 миллиарда случайных бит, поделенных на 60 файлов по десять мегабайт. Еще три файла двоичных последовательностей представляют собой типичный выход наиболее распространенных на тот момент коммерческих

ГПСП. Так же CD-ROM содержит сам пакет статистического тестирования Diehard. Каждый тест представлен в виде исходного кода на языках программирования C и Fortran.

Существует реализация пакета Diehard на языке программирования Java [1]. В отличие от оригинальной реализации 1995 года, эта программа имеет графический интерфейс. Вид диалогового окна программы представлен на рисунке 1.

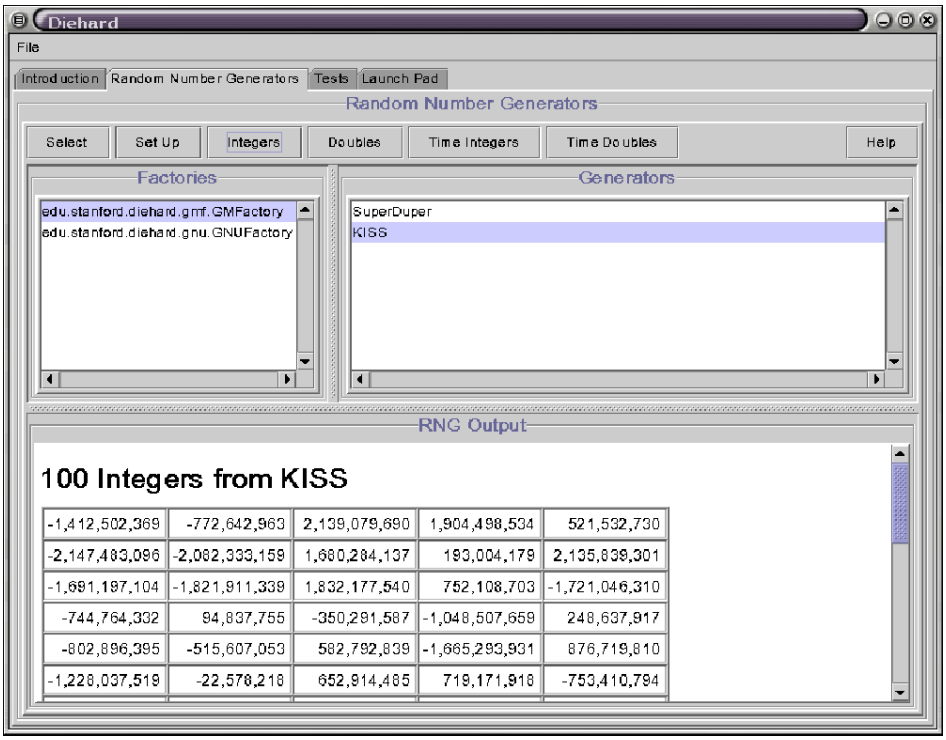

Рисунок 1 – Диалоговое окно реализации пакета Diehard

На момент выхода Diehard считался наиболее строгим пакетом статистического тестирования. Из недостатков можно отметить то, что данный пакет не обновлялся с 1998 года.

Тесты пакета Diehard также реализованы в пакете статистического тестирования Dieharder, предложенном профессором Робертом Дж. Брауном. Dieharder также находится в открытом доступе на сайте https://webhome.phy.duke.edu/~rgb/General/dieharder.php. Кроме тестов Diehard этот пакет содержит три теста из пакета NIST STS, а так же несколько тестов, разработанных автором.

Dieharder поставляется в виде установочного файла, который содержит исходный код статистических тестов и инструкции по установке и использованию пакета. Готовая программа имеет консольный интерфейс.

К преимуществам пакета Dieharder, помимо открытого доступа, следует отнести постоянное обновление и поддержку автором. Кроме того, помимо

установочного файла для операционных систем Windows и Linux, Dieharder так же представлен в виде пакета расширений для языка программирования R [2], что может быть удобно для тех, кто использует этот язык в своих исследованиях.

К одной из первых появившихся программных реализаций тестов псевдослучайных последовательностей следует отнести пакет TestU01. Впервые он был реализован в 1985 году на языке Pascal и содержал тесты, предложенные Д. Кнутом во втором томе книги «Искусство программирования». Постепенно в пакет добавлялись новые статистические тесты и реализации генераторов. Пьер Л'Экуйе и Ричард Симмард [3] полностью переработали библиотеку, реализовав ее на языке программирования C, и перевели руководство с французского на английский язык. Теперь пакет тестов TestU01 реализован в виде библиотеки расширений для языка программирования C. Следует отметить, что вся работа по тестированию генераторов псевдослучайных последовательностей происходит непосредственно с кодом.

Достаточно часто существующие разработанные пакеты тестов улучшают в связи с возрастающими требованиями к псевдослучайным последовательностям и вычислительным мощностям. Так, например, в 2012 году румынскими исследователями (Alin Suciu, Radu Alexandru Toma, Kinga Marton) была представлена распараллеленная с помощью стандарта OpenML реализация библиотеки тестов TestU01 [4], которая, при увеличении размера тестируемой последовательности, показывает большую скорость вычислений и обработки данных. Позже, в 2014 эти же исследователи представили распараллеленную версию TestU01 с использованием объектно-ориентированного подхода [5], которая показала еще больший прирост производительности и скорости вычислений. Однако подобные обновления не всегда оказываются в открытом доступе для всех пользователей.

Еще одним популярным пакетом исследования псевдослучайных последовательностей является пакет статистических тестов NIST, разработанный Лабораторией информационных технологий (Information Technology Laboratory), которая входит в состав Национального института стандартов и технологий (NIST) [6].

NIST STS находится в открытом доступе на сайте https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software. Пакет представляет собой 15 статистических тестов, представленных в виде файлов исходного кода на языке программирования C. После компиляции пользователь получает консольное приложение. Вариант программы с графическим интерфейсом пакета NIST найти не удалось.

В 2010 году румынскими исследователями издана статья, посвященная байт-ориентированной реализации пакета NIST [7], которая позволяет тестировать последовательности большого размера (параметры тестирования последовательностей в оригинальном пакете NIST строго ограничены). Тогда же эти исследователи реализовали распараллеленную версию пакета статистического

тестирования NIST [8]. К сожалению, обе эти реализации не доступны для публичного использования. В 2014 году чешскими исследователями предпринята попытка реализовать более быструю версию оригинального статистического пакета [9]. Эта реализация позволяет достичь большей скорости вычислений, к тому же находится в свободном доступе на сайте https://github.com/sysox/NIST-STS-optimised. Она, как и оригинальная реализация NIST STS, состоит из файлов кода на языке программирования C с возможностью получения консольной программы после компиляции.

Существуют и совместные реализации некоторых тестов. Пример такой реализации, включающей в себя пакеты NIST STS и Diehard, можно найти на сайте http://jrandtest.sourceforge.net/. Программный продукт написан на языке Java, имеет графический интерфейс и является свободным программным обеспечением. Недостатком этой реализации является отсутствие некоторых тестов из оригинальных версий пакетов статистического тестирования, а также 2005 год последнего обновления этой программы.

Таким образом, рассмотрение реализаций пакетов тестов псевдослучайных последовательностей позволяет заключить, что, даже если программные продукты относятся к открытому программному обеспечению, их применение требует от пользователя определенных навыков программирования, знания различных сред и языков программирования, а также временных ресурсов, необходимых для освоения программных версий. Кроме того, часть рассмотренных программ уже достаточно давно не обновлялась. Все это может вызывать затруднения при решении задач, возникающих в процессе освоения разделов учебных дисциплин, связанных с оценкой и тестированием псевдослучайных последовательностей криптографических конструкций.

Список литературы

1 Narasimhan B. JDiehard: An implementation of Diehard in Java [Электронные ресурс] / B. Narasimhan // Proceedings of the 2nd InternationalWorkshop on Distributed Statistical Computing March 15–17, Vienna, Austria. – 2001. – Режим доступа: https://www.r-project.org/conferences/DSC-2001/Proceedings/Narasimhan.pdf

2 Eddelbuettel D. RDieHarder: An R interface to the DieHarder suite of Random Number Generator Tests [Электронные ресурс] / D. Eddelbuettel, Robert G. Brown. – 2014. – Режим доступа: https://cran.r-project.org/web/packages/RDieHarder/ vignettes/ RDieHarder.pdf

3 L'Ecuyer P. TestU01: A C library for empirical testing of random number generators [Электронные ресурс] / P. L'Ecuyer, R. Simard // ACM Transactions on Mathematical Software. – 2007. – No. 4, Vol. 33. – P. 22–40. – Режим доступа: https://www.iro.umontreal.ca/~lecuyer/myftp/papers/testu01.pdf

4 Suciu A. A parallel implementation of the TestU01 statistical test suite / A. Suciu, R. A. Toma, K. Marton // IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca. – 2012, – P. 317–322.

5 Suciu A. Parallel object-oriented implementation of the TestU01 statistical test suite / A. Suciu, R. A. Toma, K. Marton // IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca. – 2014. – P. 311–315.

6 A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications, Version STS-2.1 [Электронный ресурс] / A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo // NIST Special Publication. – 2010. – Режим доступа: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf

7 Suciu A. Byte-oriented Efficient Implementation of the NIST Statistical Test Suite [Электронный ресурс]/ A. Suciu, K. Marton, I. Nagy, I. Pinca, // IEEE International Conference on Automation, Quality and Testing, Robotics. – 2010. – Режим доступа: https://www.researchgate.net/publication/232628446_Byte-oriented_efficient_implementation_of_the_NIST_statistical_test_suite

8 Suciu A. Parallel Implementation of the NIST Statistical Test Suite / A. Suciu, K. Marton, I. Nagy, I. Pinca // IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca. – 2010. – P. 363–368.

9 Sýs M. Faster randomness testing with the NIST statistical test suite / M. Sýs, Z. Říha // International Conference on Security, Privacy, and Applied Cryptography Engineering, Springer, Heidelberg. – 2014. – P. 272–284.

РАЗРАБОТКА КОМПЛЕКСНОЙ СИСТЕМЫ ОПТИМИЗАЦИИ УПРАВЛЕНИЯ ИНФРАСТРУКТУРОЙ ВИРТУАЛЬНОГО ЦЕНТРА ОБРАБОТКИ ДАННЫХ

Болодурина И.П. д.т.н., профессор; Парфёнов Д.И. к.т.н.
Оренбургский государственный университет

На сегодняшний день актуальным решением для размещения корпоративных облачных приложений и сервисов является инфраструктура, построенная на базе виртуального центра обработки данных (ЦОД) [1-2]. Однако, на сегодняшний день у операторов традиционных ЦОД, нет достаточного набора инструментов, позволяющих организовывать доступ и управление ко всем элементам сетевой инфраструктуры [3-6]. Как правило, существующие подходы охватывают только часть ресурсов, требующих управления, поэтому в рамках настоящего исследования разработано комплексное решение для системы оптимизации управления инфраструктурой виртуального центра обработки данных.

Одним из элементов предлагаемого комплексного решения является модуль классификации и построения правил маршрутизации потоков трафика в зависимости от типа облачного приложения и передаваемых данных. В основу алгоритма положена модель классификации потоков данных программно-управляемой инфраструктуры виртуального ЦОД. Для эффективного управления потоками данных такого рода необходимо одновременно осуществлять классификацию и построение правил маршрутизации на сетевых узлах. В качестве сетевой инфраструктуры для разрабатываемого программно-алгоритмического решения, в рамках исследования выбрана программно-конфигурируемая сеть, выступающая в роли среды передачи данных в виртуальном ЦОД и позволяющая вносить изменения в правила управления трафиком в режиме реального времени. Предлагаемая реализация позволяет группировать потоки трафика, циркулирующие внутри виртуального ЦОД в зависимости от типов приложений и передаваемых данных. Для эффективной классификации потоков трафика используются методы интеллектуального анализа данных (Data Mining). За счет интеллектуальной составляющей предлагаемый подход обладает свойством самоорганизации и позволяет обнаруживать и динамически создавать новые классы приложений в зависимости от следующих параметров: текущей структуры виртуального ЦОД, приоритетных задач, решаемых в нем, а так же характера и объема данных, передаваемых между узлами сети. Реализованный модуль классификации и построения правил маршрутизации потоков трафика в зависимости от типов приложений и предаваемых данных можно представить в виде двух компонентов. Первый представляет собой решение классической задачи машинного обучения, направленной на определение основных признаков и разбиения всего множества потоков трафика на группы относительно построенных ассоциативных правил. Второй основан на

подходе, позволяющем агрегировать все передаваемые потоки в единой аналитической системе сбора данных. Для решения поставленной задачи и с целью обеспечения параметров QoS сформулирована оптимизационная задача нахождение кратчайшего пути от отправителя до получателя одноадресного трафика. В качестве весов дуг используются задержки, которые должны минимизироваться. Для решения данной задачи используется классический алгоритм Дейкстры. Для маршрутизации многоадресного трафика предлагается использовать такое же сочетание двоичного поиска и алгоритма Йена. Ключевой особенностью предлагаемого решения является результат, представляемый в виде поддерева маршрутов, входящих в дерево кратчайших путей, полученного по алгоритму Дейкстры или Йена, в котором корневая вершина – отправитель, листья – получатели трафика. Конкурентным преимуществом используемого подхода является возможность внедрения предлагаемого алгоритма не только на сетевых узлах, но и на контроллере сети ЦОД. Используя предлагаемый подход элементы сети виртуального ЦОД преобразуются в множество виртуализированных модулей. Каждый модуль выполняет обработку и анализ проходящего трафика. Все собранные и проанализированные сведения предаются в агрегированной и сжатой форме в единый центр управления сетью, где они преобразуются в соответствующие правила контроллера согласно стандарту OpenFlow. Далее построенные и верифицированные правила маршрутизации трафика распространяются по сети виртуального ЦОД. Использование протокола OpenFlow в качестве базы для обмена информацией позволяет легко интегрировать предлагаемое решение в платформу виртуализации, такой как OpenStack и подобных, поскольку в них уже имеются соответствующие модули взаимодействия с OpenFlow контроллером для управления сетевой связностью запускаемых виртуальных машин. Кроме того использование предлагаемого распределенного решения позволяет избежать проблемы с дефрагментацией трафика, возникающей при поступлении большего количества мелких пакетов с высокой интенсивностью.

В рамках выполнения исследования определено, что поиск оптимального маршрута в распределенной программно-конфигурируемой сети, состоящей из большего числа узлов может занимать значительное время. В основном это связано с большим количеством альтернативных маршрутов между узлами сети. Для сокращения времени, затрачиваемого на поиск оптимального маршрута, в рамках исследования предложено решение, направленное на оптимизацию объектов, задействованных в маршрутизации трафика. Для этого необходимо оптимальным образом разместить имеющиеся объекты внутри ЦОД. Поэтому следующим элементом комплексного программно-алгоритмического решения является задача поиска и определения оптимальной схемы размещения объектов виртуального ЦОД в программно-управляемой инфраструктуре. В основу модуля планирования размещения объектов программно-управляемой инфраструктуры виртуального ЦОД на физической инфраструктуре положена построенная на первом этапе проекта имитационная модель. Полученные на этапе

моделирования данные о законах распределения и интенсивности поступления запросов различных видов трафика при помощи построенного алгоритма анализируется с применением нейросетевого подхода. Для решения оптимизационной задачи разработан алгоритм, осуществляющий мониторинг инфраструктуры виртуального ЦОД, а так же планирование размещения и запуск объектов сетевой инфраструктуры на базе контейнеров и/или виртуальных машин. Подход, применяемый в предложенном алгоритме управления размещением сетевых объектов, позволяет учитывать способ размещения и организовывать работу виртуального ЦОД с учетом циркулирующих потоков трафика, регулируя при этом количество запущенных экземпляров каждого элемента сети.

Кроме сетевых объектов в инфраструктуре виртуального ЦОД размещаются сервис-ориентированные приложения, нагрузка на которые формируется неравномерно. Это оказывает существенное влияние на качество обслуживание поступающих в виртуальный ЦОД запросов пользователей. Для решения данной задачи в рамках настоящего НИР предложено решение, основанное на ансамбле моделей, включающем в себя классификации потоков данных программно-управляемой инфраструктуры виртуального ЦОД и модель сервис-ориентированного приложения, а также модель программно-управляемого масштабируемого хранилища данных. Построенный алгоритм планирования размещения сервис-ориентированных приложений для решения оптимизационной задачи осуществляет сбор данных с системы мониторинга программно-управляемой инфраструктуры виртуального ЦОД. Подход, применяемый в предложенном в НИР алгоритме, основан на интеллектуальном анализе поступающих данных. Это позволяет определить оптимальный метод размещения и организовывать работу виртуального ЦОД с учетом входящего потока запросов пользователей, регулируя при этом число запущенных экземпляров приложений и сервисов. Гибкость предлагаемого решения обусловлена виртуализацией хранилища данных. Это позволяет динамически изменять расположение приложений в облачной системе относительно физических устройств, что дает возможность предоставлять непрерывный доступ к приложениям и сервисам. Предлагаемое решение прозрачно для клиента и масштабирует облачные приложения на несколько виртуальных устройств хранения. Это обеспечивает сокращение времени отклика приложения, а так же повышает отказоустойчивость всей системы в целом.

Для оценки эффективности перечисленных алгоритмических решений, применяемых для организации доступа к сервис-ориентированным приложениям, расположенным в программно-управляемой инфраструктуре виртуального ЦОД, в рамках НИР решена задача разработки универсальной системы количественных и качественных оценок и метрик. В основу предлагаемой системы положены базовые параметры, предлагаемые в методологии IT Infrastructure Library (ITIL), а также характеристики, учитываемые при формировании Service Level Agreement (SLA). На базе SLA в виртуальном ЦОД формируются требования к качеству обслуживания в сети.

Оценка построенных модулей проводилось с использованием симулятора программно-управляемой инфраструктуры виртуального ЦОД. Построенное в рамках НИР программное обеспечение позволяет задавать различные конфигурации инфраструктуры виртуального ЦОД, а так же проводить с их помощью исследование алгоритмов маршрутизации трафика. В симуляторе реализована возможность выбора между классическим алгоритм Дейкстры и алгоритмом, применяемым для многопутевой маршрутизации (алгоритм Йена). Построенное программное средство позволяет применять каждый из разработанных алгоритмов в отдельности и проводить сопоставительный анализ полученных результатов. Кроме того, разработанный симулятор позволяет генерировать потоки заявок различной интенсивности для нескольких классов облачных приложений и сервисов, а так же создавать динамические топологии, применяемые при развертывании виртуального ЦОД. Исследование всех алгоритмов проводилось на традиционных статистических выборках, применяемых для специализированных нагрузочных тестов сетевого оборудования.

В рамках экспериментальных исследований установлено, что созданный подход позволяет с высокой точностью повторять поведение оборудования при различных режимах работы. Проведенные с использованием симуляторов экспериментальные исследования показали не только эффективность предлагаемых в рамках НИР решений, но и существенный рост производительности относительно традиционных методик, применяемых в протоколе OpenFlow. Так, предлагаемые в НИР алгоритмические решения позволяют сократить время отклика приложений и сервисов на 15-20%.

Разработанные алгоритмические решения, основанные на гибридных методах виртуализации и использующие нейросетевой подход к определению оптимальных мест размещения приложений и сервисов виртуальном ЦОД, позволяют сократить накладные расходы на поддержание инфраструктуры на 20-25% за счет эффективного распределения нагрузки на физические устройства.

Исследование выполнено при финансовой поддержке РФФИ (проекты 16-37-60086, 16-07-01004, 18-07-01446) и гранта Президента Российской Федерации для государственной поддержки молодых российских ученых - кандидатов наук (МК-1624.2017.9).

Список литературы

1 Парфенов, Д. И. Оптимизация управления размещением виртуальных сетевых функций в виртуальном центре обработки данных с использованием нейросетевого подхода [Электронный ресурс] / Болодурина И. П., Парфенов Д. И. // Экономика и менеджмент систем управления: науч.-практ. журнал ,2017. - № 3.1 (25). - С. 143-151.

2 Парфенов, Д. И. Оптимизация управления распределением трафика в программно-управляемой инфраструктуре виртуального ЦОД на основе имитационной модели [Электронный ресурс] / И. П. Болодурина, Д. И. Парфёнов // Вестник ВГУИТ,2017. - Т. 79, № 1. - С. 99-105.

3 Ворожцов, А.С. Динамическое распределение вычислительных ресурсов центров обработки данных / Ворожцов А.С., Тутова Н.В., Тутов А.В. // T-Comm: Телекоммуникации и транспорт, 2016. - Т. 10. № 7. - С. 47-51.

4 Зотов, И.А. Алгоритм распределения ресурсов в центрах обработки данных с единым планировщиком для различных типов ресурсов / Зотов И.А., Костенко В.А. // Известия Российской академии наук. Теория и системы управления, 2015. - № 1. - С. 61-71.

5 Вдовин, П.М. Сравнение различных подходов к распределению ресурсов в центрах обработки данных / Вдовин П.М., Зотов И.А., Костенко В.А., Плакунов А.В., Смелянский Р.Л. // Известия Российской академии наук. Теория и системы управления, 2014. - № 5. - С. 71-83.

6 Алексанков, С.М. Модели динамической миграции с итеративным подходом и сетевой миграции виртуальных машин // Научно-технический вестник информационных технологий, механики и оптики, 2015. - Т. 15. № 6. - С. 1098-1104.

КРИТЕРИИ ВЫБОРА СРЕДСТВ ФИЗИЧЕСКОЙ ЗАЩИТЫ ОБЪЕКТОВ ИНФОРМАТИЗАЦИИ

Бурькова Е.В., канд. пед. наук, доцент
Оренбургский государственный университет

Современные объекты информатизации располагают значительным количеством и разнообразием ресурсов, требующих высокого уровня защиты. В качестве защищаемых ресурсов выступают: персонал, материальные и финансовые ресурсы, конструкторская и технологическая документация, информационные ресурсы, включающие различного рода конфиденциальную информацию (коммерческая тайна, служебная тайна, персональные данные и т.д.). Для обеспечения защиты от угроз проникновения нарушителя на объект и как следствие нанесение ущерба предприятию, важной задачей является проектирование системы физической защиты.

Проектирование системы физической защиты является сложным процессом, что обусловлено наличием неопределенностей, таких как качественные признаки описания объекта, неполные знания о вероятных угрозах, о степени оснащенности и подготовленности нарушителя, оценка вероятного ущерба, критерии выбора средств физической защиты. В настоящее время задача проектирования системы физической защиты решается с применением средств автоматизации, что позволяет уменьшить трудозатраты на процесс проектирования и избежать возможных ошибок.

Для создания системы поддержки принятия решений при проектировании системы физической защиты были проанализированы основные этапы:

- анализ деятельности защищаемого объекта;
- характеристика защищаемых ресурсов;
- определение категории объекта;
- формирование основных требований по защите;
- построение модели угроз и модели нарушителя;
- выбор средств физической защиты;
- оценка эффективности системы физической защиты.

Проведение тщательного экспертного анализа объекта защиты является определяющим фактором для принятия правильных решений при выборе средств физической защиты. В рамках обследования выявляется категория защищаемого объекта, которая определяет набор требований нормативно-правовых документов по обеспечению безопасности данного объекта.

Одним из важных этапов проектирования системы физической защиты является определение критериев выбора средств защиты, учитываются следующие:

- перечень актуальных угроз безопасности;
- величина вероятного ущерба от реализации угроз;
- помехи, создаваемые окружением объекта;

- совместимость с существующими средствами защиты;
- планируемые затраты на средства защиты.

Первый критерий – это перечень актуальных угроз безопасности, который необходимо составить в соответствии с выявленными уязвимостями объекта. В качестве актуальных угроз безопасности объекта информатизации с точки зрения физической защиты рассматриваются угрозы:

- проникновение нарушителя на территорию объекта;
- проникновение в кабинеты ограниченного доступа;
- кража материальных и финансовых ценностей;
- кража информации;
- терроризм;
- кража интеллектуальной собственности;
- нарушение технологического процесса;
- возникновение чрезвычайной ситуации (пожар, взрыв и др.).

Подсистема формирования перечня угроз представлена на рисунке 1.

Рисунок 1 - Структурная схема подсистемы формирования перечня угроз

Актуальными признаются те угрозы, вероятность реализации которых высокая, остальные угрозы считаются неактуальными. Вероятность реализации угроз определяется экспертным путем, при этом рассматриваются имеющиеся средства защиты и их способность нейтрализовать угрозы. Задача выбора средств физической защиты заключается в том, чтобы каждое средство защиты по возможности перекрывало несколько угроз. В связи с этим составляется матрица соответствия угроз и средств защиты.

Второй критерий – величина вероятного ущерба, рассчитывается исходя из материальных активов объекта информатизации, которые могут подвергнуться угрозам. А также необходимо учитывать наложение штрафов за несоблюдение мероприятий защиты со стороны регулирующих органов, осуществляющих надзор. Например, несоблюдение требований по защите персональных данных подлежит административной ответственности (наложению штрафов, доходящих до миллиона рублей).

Третий критерий – помехи, создаваемые окружением объекта. При выборе охранных извещателей очень важно учитывать помеховую обстановку на объекте, так как принцип работы некоторых извещателей основан на физических явлениях, действие помех на которые могут значительно искажать их работу. Примером могут служить вибрационные, акустические, радиоволновые, радиолучевые извещатели, на которыенакладывают искажения такие помехи, как железнодорожные линии, линии электропередач и другие. Помеховая ситуация может изменяться, например, возле здания могут начаться строительные работы с использованием тяжелой техники, что создаст акустические помехи.

Частота ложных тревог является основной характеристикой, по которой можно судить о помехоустойчивости датчика. Помехоустойчивость – это показатель качества датчика, характеризующий его способность стабильно работать в различных условиях. Были проанализированы основные дестабилизирующие факторы, являющиеся причиной возникновения ложных тревог. Все они могут быть разбиты на: внутренние шумы и внешние помехи. Усредненное влияние помех на работу извещателей представлено в таблице 1.

Таблица 1 - Усредненное влияние помех на работу извещателей

Вид помехи	Тип извещателя				
	Аккустический	Оптико-электронный	Радиоволновой	Емкостной	Вибрационный
Внешние акустические шумы (уличные, раскаты грома и др.)	+	-	-	-	+

Внутренние (в контролируемой зоне) акустические шумы (холодильники, ТА, шум воды в трубах и др.)	+	-	-	-	-
Внешний свет (свет фар, солнечные блики)	-	+	-	-	-
Движение воздуха в помещении (сквозняки, вентиляторы, батареи отопления)	-	+	-	-	-
Движение предметов (штор, лопастей вентилятора, воды на стеклах, листьев и др.)	+	+	+	-	-
Электромагнитные помехи (сварочные аппараты, разряды высоковольтных линий ЛЭП, трамваев, троллейбусов, люминесцентные лампы и др.)	-	-	-	+	-
Мелкие животные, крупные насекомые	+	+	+	+	+

Совместимость с существующими средствами защиты – это четвертый критерий, который должен быть учтен при проектировании системы физической защиты. Для этого проводится комплексный анализ существующей системы защиты на объекте. В настоящее время создаются интегрированные системы защиты, в которых все подсистемы взаимосвязаны между собой. Считается целесообразным при модернизации выбирать средства защиты совместимые с имеющимся комплексом защиты.

Последним критерием является экономические затраты на реализацию системы физической защиты. Затраты должны полностью соответствовать размеру вероятного ущерба от реализации угроз. Иначе системы защиты будет экономически невыгодной, нерациональной. Этот критерий напрямую связан с критерием ущерба.

Таким образом, при проектировании системы физической защиты выбор средств защиты необходимо осуществлять на основе проанализированных критериев, что позволит обеспечить высокий уровень защиты объекта информатизации с оптимальными затратами.

Список литературы

1. *Бурькова, Е. В., Прикладная программа оценки физической защищенности объекта на основе логико-вероятностного подхода / Е.В. Бурькова, Д. А. Гайфулина, Э. Р. Хакимова // Информационные ресурсы и системы в экономике, науке и образовании : материалы VI Международной науч.-практ. конф.,15 июня 2016 г., Пенза / Приволжский дом знаний – Пенза, 2016. – С. 10–14.*

2. *Бурькова, Е. В. Категорирование объектов информатизации для выбора средств физической защиты [Электронный ресурс] / Е.В. Бурькова // Университетский комплекс как региональный центр образования, науки и культуры : материалы Всерос. науч.-метод. конф. (с междунар. участием), 4-6*

февр. 2017 г., Оренбург / Оренбург. гос. ун-т. – Электрон. дан. – Оренбург , 2017. – С. 3073-3076.

3. Лукоянов, С. В., Основные требования к системам физической защиты на этапе их проектирования / С. В. Лукоянов, С. В. Белов // Вестник астраханского государственного технического университета. – 2010. – № 2. – С. 163-171.

4. Рытов, М. Ю. Модель процесса выбора состава технических средств систем физической защиты / М. Ю. Рытов, В. Т. Еременко, М. Л. Гулак // Информация и безопасность. – 2015. – № 4. - С. 502-507.

АНАЛИЗ ЭТАПОВ СОЗДАНИЯ СИСТЕМЫ ЗАЩИТЫ ПЕРСОНАЛЬНЫХ ДАННЫХ ОБЪЕКТА ИНФОРМАТИЗАЦИИ

Бурькова Е. В., канд. пед. наук, доцент, Недорезова А. С.
Оренбургский государственный университет

В современном мире глобальной информатизации, в условиях непрерывного роста киберугроз, защита персональных данных (ПДн), размещенных в информационных системах, становится все более актуальной задачей, привлекающей внимание международного сообщества. На IV Международной конференции «Защита персональных данных», организованной при поддержке Роскомнадзора в ноябре 2017 года, обсуждались насущные проблемы, связанные с персональными данными, были выработаны перспективные предложения и инициативы в сфере регулирования и контроля защиты персональных данных, взаимоприемлемых норм правоприменения законодательных новаций в области защиты персональных данных.

Главной целью создания системы защиты персональных данных объекта информатизации, является предотвращение или сведение к минимуму ущерба (косвенного, материального, морального, вещественного и т. д.). Ущерб возникает в результате реализации угроз безопасности, источниками которых могут быть субъекты информационных отношений с помощью неправомерного и нежелательного воздействия на информацию, ее носители, систему в целом.

Нормативно-правовой базой обеспечения безопасности персональных данных являются руководящие документы Правительства РФ, ФСТЭК и ФСБ, содержащие требования к мероприятиям по защите ПДн:

− Федеральный закон № 152-ФЗ «О персональных данных»;

− Федеральный закон № 149-ФЗ «Об информации, информационных технологиях и о защите информации»;

− Постановление Правительства РФ от 1 ноября 2012 г. N 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных»;

− Приказ ФСТЭК России № 21 от 18 февраля 2013 г. «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».

− Приказ ФСТЭК России № 17 от 11 февраля 2013 «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах».

Обеспечение требуемого уровня защищенности должно достигаться с использованием мер, методов и средств безопасности, которые подразделяются на следующие группы:

- законодательные (правовые);

– организационные (административные);
– физические;
– технические (аппаратные и программные).

Защищаемыми ресурсами, в которых находят свое отражение персональные данные объекта информатизации, являются:

– информационные ресурсы;
– средства и система обработки информации;
– аппаратные средства – ЭВМ и составные части (процессоры, мониторы, терминалы, периферийные устройства - принтеры, контроллеры, кабели, линии связи и т.д.);
– программное обеспечение (ПО) – приобретенные программы, исходные, дополнительное системное ПО, загрузочные модули утилиты, диагностические программы и т.д;
– персональные данные, хранимые временно и постоянно, на различных носителях, печатные, электронные, системные процессы и т.д.

Персональные данные должны быть защищены как от преднамеренной, так и незапланированной угрозы. Создание системы защиты персональных данных, приведение процессов обработки и обеспечение безопасности персональных данных в соответствие с положениями и требованиями нормативных документов позволяет минимизировать риски, связанные с утечками ПДн.

Разработка этапов создания системы защиты персональных данных предполагает определение основных операций по анализу обстановки на объекте, выявление актуальных угроз безопасности и формирование направлений по защите ПДн.

Нами был разработан перечень и описание основных этапов создания системы защиты персональных данных на объекте информатизации, который представлен в таблице 1.

Таблица 1- Анализ этапов построения системы защиты ПДн

Наименование этапа	Содержание этапа	Результат
1	2	3
Обследование объекта обработки ПДн	1. Анализ категорий обрабатываемых ПДн; 2. Анализ программно-аппаратных средств ИСПДн; 3. Анализ угроз ПДн; 4. Определение уровня исходной защищенности ИСПДн; 5. Анализ возможного ущерба.	1. Уровень защищенности ИСПДн; 2. Модель угроз; 3. Модель нарушителя; 4. Предварительная оценка возможного ущерба.

Продолжение таблицы 1

1	2	3
Формирование требований к системе защиты ПДн	Формирование требований нормативно-правовых документов на основании выявленного уровня защищенности ИСПДн.	Перечень требований приказов ФСТЭК к уровню защищенности ИСПДн
Анализ существующих средств защиты ПДн на объекте	Анализ всех имеющихся на объекте средств защиты ПДн: организационно-распорядительной документации, физических средств защиты; аппаратно-программных средств защиты ПДн	Список подсистем, не удовлетворяющих требованиям нормативных документов к защите ПДн
Формирование концепции защиты	Определение направлений защиты: по подразделениям; по уязвимостям; по категориям ПДн	1. Выбранное направление защиты. 2. Выбранный способ построения системы защиты. 3. Составление плана мероприятий по управлению системой безопасности ПДн
	Выбор способов защиты: по направлениям защиты; по актуальным угрозам; по экономическим затратам	
	Решение вопросов управления защитой	
Решение основных вопросов обеспечения защиты ПДн	Подготовка документации; Финансовое обеспечение; Подготовка кадров; Закупка и установка аппаратно-программного обеспечения защиты	Организационно-распорядительные документы; Реализация системы защиты ПДн
Анализ эффективности защиты ПДн	Оценка эффективности системы защиты ПДн	Результаты расчета экономической и технической эффективности системы защиты ПДн

Формирование концепции защиты, является важным этапом организации систем безопасности ПДн, в результате, которого определяется выбор направлений и способов защиты ПДн. На рисунке 1 показана схема формирования концепции защиты ПДн.

Направления защиты могут быть определены по отдельным подразделениям, по уязвимым звеньям ИСПДн, а также в зависимости от категорий обра-

батываемых ПДн. Способы защиты чаще всего реализуют по принципу нейтрализации актуальных угроз безопасноти ПДн, но обязательно учитывают экономический критерий в целях реализации наиболее рациональной системы защиты ПДн.

Рисунок 1 – Схема формирования концепции защиты ПДн

Таким образом, сформированная последовательность этапов создания системы защиты ПДн объекта информатизации позволит спланировать и организовать работу по созданию системы защиты ПДн с учетом наиболее важных критериев, что способствует обеспечению высокого уровня безопасности ПДн объекта.

Список литературы

1. Алексашина, М. Н. Защита персональных данных как условие обеспечения безопасности личности / М.Н. Алексашина // Право и безопасность. – 2014. – № 1. – С. 68-73.

2. Бурькова, Е. В. Задача оценки защищенности информационных систем персональных данных / Е. В. Бурькова, // Вестник Чувашского университета. – 2016. – № 1. – С. 112–118.

3. Бурькова, Е. В. Система защиты персональных данных в высшем учебном заведении / Е. В. Бурькова // Интеллект. Инновации. Инвестиции. –2017. – № 7. – С. 69–74.

4. *Назаров, И. Г. Особенности организации обеспечения безопасности персональных данных при их обработке в информационных системах персональных данных / И. Г. Назаров, Ю. К. Язов, Е. С. Остроухова // Информация и безопасность. - 2009. Т. 12. - № 1 - С. 71-76.*

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПРИ ВЫПОЛНЕНИИ ЗАДАНИЙ ПО КОМПЬЮТЕРНОЙ ГРАФИКЕ

Ваншина Е.А., канд. пед. наук, доцент,
Ваншин В.В., канд. с.- х. наук, доцент
Оренбургский государственный университет

В настоящее время в условиях изменений, происходящих в российском обществе, в системе высшего образования, большую роль и значение отводят самостоятельной работе студентов в вузе как основному резерву повышения качества подготовки будущих специалистов.

Самостоятельная работа студента (СРС) – главная составляющая учебного процесса, в процессе которого происходит формирование знаний, умений и навыков самостоятельной работы в учебной, научной и будущей профессиональной деятельности, профессионально значимых качеств личности будущего специалиста, способности принимать на себя ответственность, самостоятельно решать проблемы, находить их конструктивные решения.

Самостоятельная работа направлена на то, чтобы углубить и расширить знания, сформировать интерес к познавательной деятельности, овладеть приемами процесса познания, развить познавательные способности (М.В. Буланова-Топоркова).

Ученые и педагоги указывают на то, что роль преподавателя велика, но главные цели образования можно достигнуть только в результате усилий самих обучающихся. Самостоятельная работа студента заключается не в пассивном «поглощении» готовой информации, а ее поиске и творческом усвоении. Она должна подготовить студента к самостоятельной деятельности в будущем.

В отечественной педагогической теории и практике существуют разные подходы к пониманию терминов «самостоятельность», «самостоятельная работа», трактовка которых зависит от того, какое содержание заложено в слово «самостоятельный». Анализ литературы показал, что существуют такие понимания этого термина:

1) обучающийся работает самостоятельно, не принимая помощь от преподавателя;

2) от обучающегося необходимы самостоятельные мыслительные операции, самостоятельная ориентация в учебном материале;

3) исполнение работы обучающимся не отрегулировано, он может сам выбирать, что содержит и какими методами выполнять задание.

Термин «самостоятельная работа» чаще употребляется в первом значении (М.П. Кашин). Для второго значения термина «самостоятельный» важно то, чтобы обучающиеся самостоятельно могли думать и решать проблемы при любой форме учебной работы. В третьем значении самостоятельности определяют

такие виды деятельности обучающихся как самостоятельная и исполнительная. К самостоятельной деятельности относят ту, которую обучающиеся осуществляют, имея внутренние побуждения, находят цели и средства деятельности самостоятельно.

По мнению Л.В. Мац и Ф.С. Лесева, самостоятельная работа студента подразделяется на два вида: организуемую преподавателем и без непосредственного контроля преподавателя (подготовка к практическим занятиям, зачетам, написание рефератов, курсовых, дипломных работ).

Существуют различные точки зрения ученых к пониманию термина «самостоятельная работа» студентов в вузе: так С.И. Архангельский видит ее в самостоятельном поиске нужной информации, приобретении знаний, их использовании для решения учебных, научных и профессиональных задач; А.Г. Молибог – в деятельности, состоящей из творческого восприятия и осмысления учебного материла на лекции, при подготовке к практическим занятиям и лабораторным работам, экзаменам и зачетам, выполнения курсовых работ и дипломного проекта; Р.А. Низамов – в разных видах познавательной деятельности студентов на учебных занятиях или во внеучебное время без руководства, но под наблюдением преподавателя; С.И. Зиновьев – в самообразовании; М.В. Буланова-Топоркова – в планируемой работе студентов, совершаемой по заданию и при методическом руководстве преподавателя, но без его непосредственного участия; В.М. Рогинский – в планируемой познавательной, организационно и методически направляемой деятельности студентов без прямой помощи преподавателя для достижения определенного результата.

Т.А. Ильин дает определение самостоятельной работе как особому виду учебной деятельности обучающихся под руководством, но без участия преподавателя, что характеризуется активностью протекания познавательных процессов на занятиях и во внеурочное время, повышает эффективность процесса обучения и готовит обучающихся к самостоятельному накоплению знаний.

Сущность самостоятельной работы обучающихся, – как отмечает С.Я. Батышев, – «заключается в организации самостоятельной познавательной деятельности. Она является одним из важных средств подготовки обучающихся к активной самообразовательной работе, и в этом состоит ее основная дидактическая цель». Самостоятельная работа авторами подразделяется на: урочную и внеурочную (самоподготовка, самообразование). Она активизирует обучающихся своим организационным устройством и содержанием заданий, позволяет работать в индивидуальном темпе и стиле.

Самостоятельная работа как этап практического занятия состоит в том, что обучающиеся определяют, как решить поставленные перед ними задачи, выбирают алгоритм действий, непосредственно решают эти задачи, оформляют необходимые отчеты (С.Я. Батышев).

Некоторые авторы считают, что самостоятельные работы студентов – это работы по заданиям, которые требуют перестроить изученный материал, комбинировать его по-другому, отыскивать или создавать что-то новое. Работы,

состоящие в подражании образцу, и упражнения, цель которых заключается в тренировке в навыке, не являются самостоятельной работой.

Ученые-педагоги определяют такие типы самостоятельных работ как:

1) формирование у обучаемых умений выявлять во внешнем плане то, что от них требуют на базе показанного им алгоритма деятельности и посылок на эту деятельность, что содержится в условии задания (домашние задания – работа с учебником, конспектом, лекцией);

2) формирование знаний-копий и знаний, позволяющих решать типовые задачи (отдельные этапы лабораторных работ и практических занятий, типовые курсовые работы и проекты);

3) создание условий для творческой деятельности (научно-исследовательские задания, в том числе курсовые работы и дипломный проект) (В.М.Рогинский).

К функциям самостоятельной работы как формы учебно-воспитательного процесса относятся: образовательная, развивающая и воспитательная.

Но вместе с тем основной задачей преподавателя, по мнению многих ученых, является выбор содержания работ, их целевая установка, контроль исполнения, четкое задание времени для выполнения работы, оказание, если требуется, необходимой помощи, выбор вида самостоятельной работы и организационно-методических форм ее осуществления; к задаче обучаемого относится проявление самостоятельности при решении поставленных перед ним задач.

Анализ определений изучаемого термина показал, что самостоятельная работа обучающихся предполагает, что их деятельность осуществляется с помощью и под руководством преподавателя. В то же время для обучающихся, имеющих домашние ПЭВМ, возможна самостоятельная работа вне занятий, но не управляемая и лишь отсрочено и частично контролируемая преподавателем.

Важную роль в организации самостоятельной работы студентов играет преподаватель, который знает структуру и виды самостоятельных работ, методику их использования, методы контроля за их выполнением. В связи с этим методически правильное планирование, организация и реализация самостоятельной работы студента под руководством преподавателя является важной задачей обучения студента в вузе. Это способствует развитию мышления обучающихся, интеграции мыслительной и практической деятельности будущих специалистов, овладению обучающимися экспериментальным методам исследования (умений наблюдать, измерять и оформлять результаты, планировать и др.)

В данной работе мы рассмотрим одно из направлений возможной оптимизации самостоятельной работы студентов, а именно ее организацию при изучении курса компьютерной графики, основной целью которой является формирование представления о возможностях преобразования графической информации на плоских эквивалентах пространства с использованием средств вычислительной техники.

В процессе изучения раздела «Компьютерная графика» дисциплины «Инженерная и компьютерная графика» студенты инженерно-технических

направлений подготовки, выполняя практические задания на компьютере в современных графических системах (AutoCAD, КОМПАС), учатся создавать электронные версии инженерно-технических чертежей, редактировать, оформлять их в соответствии с требованиями действующих стандартов, обрабатывать растровые изображения, создавать модели деталей, их двумерные и трехмерные изображения, создавать плоские чертежи, ассоциативно связанные с построенными моделями.

При этом большое значение имеет построение практических занятий студентов. Для решения обозначенных выше дидактических задач целесообразно на практических занятиях по компьютерной графике включать студентов в разнообразные по содержанию и форме самостоятельные работы. Это позволит им не только овладеть навыками работы с различными компьютерными графическими системами, но и применить данные навыки в дальнейшем при самостоятельном изучении каких-либо других программ компьютерной графики, а также последующих дисциплин учебного плана, где используется графическое представление информации. При разработке практических занятий целесообразно применять самостоятельные работы, разрабатываемые в зависимости от уровня активности обучающихся.

Самостоятельная работа содержит воспроизводящие и творческие процессы в деятельности студента, в зависимости от которых различают, по мнению М.В. Булановой-Топорковой, три уровня самостоятельной деятельности студента:

1) репродуктивный;

2) реконструктивный;

3) творческий.

Нам видится целесообразным принять за основу следующую градацию самостоятельных работ студентов в зависимости от степени их активности:

1) репродуктивная самостоятельная работа студента по заданию преподавателя, образцы выполнения которой ему уже известны; цель таких заданий – закрепление знаний, формирование и совершенствование умений и навыков;

2) реконструктивно-вариативная самостоятельная работа, которая осуществляется без непосредственного участия преподавателя и требует избирательного отношения студента к известным ему средствам и методам решения поставленной задачи;

3) творческая самостоятельная работа включает элементы самостоятельного исследования. Преподаватель в этом случае создает необходимую дидактическую ситуацию, объективно направленную на отыскание студентами новых, ранее им неизвестных, средств и методов решения поставленной перед ними задачи и формулирование новых обобщений.

Репродуктивные самостоятельные работы по образцу, которые требуют переноса известного способа решения в аналогичную внутрипредметную

ситуацию, осуществляются на основе «конкретных алгоритмов», ранее показанных преподавателем и опробованных обучающимися при выполнении предыдущих заданий. Здесь рассматривается самостоятельное решение задания по способу, продемонстрированному преподавателем или подробно описанному в учебно-методических изданиях по изучаемой дисциплине. При выполнении такой самостоятельной работы студенты напрямую переносят известные им способы для решения схожих заданий, что является основой формирования умения планирования собственной познавательной деятельности и опыта познавательной самостоятельности.

Реконструктивно-вариативные самостоятельные работы, состоящие в использовании известного ранее способа с некоторой модификацией в новую внутрипредметную проблемную ситуацию, содержат в себе познавательные задачи, по условиям которых студентам необходимо: провести анализ необычных для них ситуаций; выявить характерные признаки учебных проблем, возникающих в этих ситуациях; найти способы решения этих проблем; выбрать из известных способов наиболее рациональные, модифицируя их в соответствии с условиями ситуации обучения. Все эти действия не очевидны, поэтому для обнаружения возможности использования того или иного ранее известного способа деятельности, необходимо преобразовать исходную ситуацию, выполнив определенные действия.

Указанные виды самостоятельных работ требуют для своего решения устанавливать не только отдельные функциональные связи в ранее усвоенных знаниях и методах их применения, но и их структуру в целом. Их выполнение стимулирует обучающегося применять усвоенные ранее знания, что делает их более глубокими.

Еще более высокий уровень самостоятельности проявляют студенты при выполнении частично-поисковых, эвристических самостоятельных работ, требующих переноса нескольких известных способов решения в необычные внутрипредметные проблемные ситуации и их комбинирования.

Проиллюстрируем данный способ организации самостоятельной работы студентов на практических занятиях по компьютерной графике по теме: «Построение чертежей сложных объектов и наглядных изображений».

На первом этапе студенты выполняют репродуктивную самостоятельную работу по образцу, подробно описанному в разработанных нами учебно-методических изданиях по моделированию в системе КОМПАС, на примере применения простых операций в графической системе КОМПАС при решении таких учебных заданий, как: выполнение «по заданному чертежу наглядного изображения детали простой технической формы по указанным размерам» и «по построенному наглядному изображению ассоциативно связанного с ним плоского чертежа детали».

Следующим этапом является выполнение реконструктивно-вариативной самостоятельной работы по «созданию наглядного изображения детали сложной технической формы по указанным размерам по заданному наглядному

изображению», где студенты применяют опробованные ими ранее операции в условиях усложнения задания.

Завершающим этапом в изучении данной темы является частично-поисковая самостоятельная работа по выполнению «по заданному чертежу наглядного изображения типовой детали «втулка» по заданным размерам с вырезом 1/4», когда студент должен найти рациональный метод проведения графических операций, наглядно оформить чертеж, свободно владеть навыками использования команд.

Таким образом, правильная организация самостоятельной работы студентов является подготовкой студентов к самообразованию, саморазвитию, самореализации, а сформированные в результате выполнения самостоятельных работ по компьютерной графике знания, умения и навыки работы с компьютерными графическими системами, а также навыки самостоятельной творческой работы студент реализует на следующих этапах обучения при выполнении курсовых работ и дипломного проекта, а также в последующей производственной деятельности.

Список литературы

1. Ваншина, Е. А. Проблемы компьютеризации преподавания графических дисциплин / Е. А. Ваншина // Вестник Оренбургского государственного университета. – 2009. – №2. – С.143.

2. Ваншина, Е. А., Гущин, Л. Я. Разработка и применение дидактического материала по инженерной графике с использованием системы КОМПАС-3D / Е. А. Ваншина, Л. Я. Гущин // Актуальные проблемы технических наук в России и за рубежом: сборник статей Международной научно-практической конференции. – Уфа: Аэтерна, 2014. – С.13-17. – ISBN 978-5-906769-51-0

3. Ваншина, Е. А., Гуньков, В. В. Об использовании единого дидактического материала при обучении студентов физике и инженерной графике (на примере Оренбургского государственного университета) / Е. А. Ваншина, В. В. Гуньков // Вестник Оренбургского государственного университета. – 2015. – №2(177). – С.10-16.

4. Ваншина, Е. А., Горельская, Ю. В. 3D-моделирование в среде КОМПАС: методические указания / Е. А. Ваншина, Ю. В. Горельская. – Оренбург: РИК ГОУ ОГУ, 2004. – 30 с.

5. Ваншина, Е. А., Егорова, М.А. 2D-моделирование в системе КОМПАС: методические указания / Е. А. Ваншина, М. А. Егорова. – Оренбург: ГОУ ОГУ, 2010. – 88 с.

6. Ваншина, Е. А., Егорова, М.А. Моделирование в системе КОМПАС: методические указания / Е. А. Ваншина, М. А. Егорова. – Оренбург: ГОУ ОГУ, 2011. – 74 с.

РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА ОПТИМАЛЬНОЙ ЦВЕТОКОРРЕКЦИИ

Влацкая И.В., канд. техн. наук, доцент,
Баранов Д.А., Влацкая Е.Ф.
Оренбургский государственный университет

С развитием цифровой фотографии и её доступности широким слоям населения стало появляться всё больше графических редакторов, направленных на обработку изображений. Одни из них представляют собой объёмные профессиональные продукты, в функционал которых заложено большое число возможностей, способных удовлетворить потребности профессионалов на самом высоком уровне. Многофункциональность таких программ несёт в себе ряд неудобств для рядового пользователя. Несмотря на широкий выбор инструментов в профессиональных программах обработки изображений, эти функции зачастую являются избыточными. В таких программах обычно несколько инструментов, регулирующих яркость или отвечающих за коррекцию цвета, причем используя эти функции, можно добиться одного и того же результата. Интересно, что даже в продуктах одной компании Adobe Photoshop и Adobe Lightroom инструменты цветокоррекции различаются. При этом иногда появляется необходимость использовать для обработки изображения инструменты как первой, так и второй программы. Это не совсем удобно, поскольку при обработке в нескольких редакторах из-за необходимости повторного сохранения, которое влечет за собой сжатие, теряется часть качества.

Второй тип программ объединяет средние и мелкие продукты с меньшим функционалом, но проще в освоении и требованиях. Такие программы имеют более узкую направленность, что ограничивает действия пользователя, но не даёт ему запутаться. Часто они не требуют лицензии, что значительно упрощает работу. Ограниченность программы набором функций для решения всего нескольких задач может быть не только её недостатком, но и преимуществом: этим задачам уделяется больше внимания и представлены различные способы их решения. Такие программы являются оптимальным выбором для пользователя, который применяет обычно один набор функций, например, только поправляет яркость/контраст и цвет изображения.

Таким образом, сравнивая функциональные характеристики существующих графических редакторов, приходим к тому, что все программы имеют как достоинства, так и недостатки. Для создания нового графического редактора необходимо уточнить задачи, подлежащие решению, чтобы, если это возможно, избежать недостатков существующих программ. Таким образом, можно сформировать следующие требования к разработке графического редактора:

- простота освоения. Необходимо максимально сократить время, затрачиваемое пользователем на изучение интерфейса программы и её функциональ-

ных возможностей. С этой целью можно разработать справку программы и реализовать стандартный пользовательский интерфейс;

- оптимизация под многоядерные процессоры продиктована современной техникой. В настоящий момент почти все компьютеры и смартфоны имеют несколько ядер, а изображения хранят в себе всё больше мегапикселей, что означает объёмную работу для функций графического редактора;

- наличие алгоритмов цветокоррекции, ускоряющих работу с цветом. Все графические редакторы имеют такие инструменты цветокоррекции, которые необходимо изучать и набирать опыт, прежде чем будет получаться стабильный результат. Кроме того, от пользователя необходимо понимание цвета, чтобы двигать ползунки не наугад, а точно зная, изменение каких величин принесёт за собой желаемый результат. Необходимо сохранить количество кликов пользователя, позволяющих добиться нужного результата;

- расширенные возможности выборочной коррекции цвета. Необходимо объединить различные варианты выборочной цветокоррекции, представленные в рассмотренных графических редакторах, в одной функции для наиболее полного использования их возможностей;

- построение цветовой схемы по изображению;

- обработка изображения в соответствии с цветовой схемой позволит создавать изображения, подходящие к цветовой схеме web-ресурса.

Решив поставленные задачи, можно будет устранить некоторые недостатки рассмотренных графических редакторов и повысить эффективность цветокоррекции изображений.

Существует множество средств, значительно упрощающих разработку программного средства. При выборе учитывались следующие критерии:

- свободный доступ к программному обеспечению;
- наличие необходимого для разработки функционала;
- относительная простота использования;
- совместимость с другими средствами.

В соответствии с перечисленными критериями необходимо определить инструменты разработки проекта программного средства, среду разработки для подходящего языка программирования с возможностью создания графического интерфейса, а так же выбрать технологию параллельного программирования, призванную повысить эффективность алгоритма цветокоррекции.

В основе разрабатываемо редактора лежит алгоритм статистической цветокоррекции. Целью этого способа цветокоррекции является придание исходному изображению колорита другого изображения так, чтобы исходное выглядело естественным и сохранило свои особенности. С помощью этого алгоритма можно избавиться от нежелательных оттенков на изображении или придать желаемые.

Метод заключается в том, что прежде всего вычисляются математическое ожидание и дисперсия цвета на обоих изображениях. После применения алгоритма математическое ожидание цвета целевого изображения заменяется на

математическое ожидание изображения-источника цвета. Аналогично меняется дисперсия. Для вычислений применяются формулы:

$$C_t^{new} = E_s + (C_t - E_t)\frac{D_s}{D_t}, \quad (1)$$

$$E_i = \frac{1}{n_i}\sum_{j=1}^{n_i} C_i^j, \quad (2)$$

$$D_i = \sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i}(C_i^j - E_i)^2}, \quad (3)$$

$$i \in \{s, t\}, \quad (4)$$

где C - цветовой канал пиксела;

D – дисперсия;

E – математическое ожидание;

new – индекс для обозначения нового значения цветового канала пиксела;

s – индекс принадлежности пиксела к изображению-источнику цвета;

t – индекс принадлежности пиксела к целевому изображению;

n – количество пикселей в изображении.

Эта формула применяется к значению каждого цветового канала каждого пиксела целевого изображения [4].

Поскольку вычисления в данном алгоритме выполняются циклически для каждого пикселя целевого и донорного изображений, целесообразно распараллелить эти циклы для увеличения скорости обработки изображения. При этом эффективнее всего разделить итерации между потоками динамически, чтобы сбалансировать время их работы. С другой стороны, в данном алгоритме можно было бы осуществить параллелизм по данным и предоставить потокам работу с разными изображениями, однако в таком случае дисбаланс времени работы потоков оказался бы существенным. Дисбаланс был бы связан с возможной разницей размеров целевого и донорного изображений, а значит и объёма вычислений.

После изучения теории и проведения практических экспериментов с помощью данного алгоритма, можно сделать следующие выводы:

- метод действительно решает поставленную задачу, т.е. осуществляет изменение колорита целевого изображения с сохранением его особенностей и результат перекраски при разумном подборе пары изображений выглядит естественно, т.е. обеспечивается ожидаемый эффект перекраски. Алгоритм также позволяет убирать нежелательные оттенки с фотографий;

- скорость работы алгоритма напрямую зависит от размера входных изображений, т.к. преобразовывается каждый пиксел. При этом скорость работы алгоритма может быть существенно повышена за счет применения технологии параллельного программирования;

- благодаря использованию статистики метод применим независимо от

размеров входных изображений и соответствия размеров между собой. В чистом виде метод перекрашивает изображение целиком;

- важным нюансом является то, что изображения должны быть схожи в своей композиции и содержании, иначе результат может быть непредсказуемым.

Главное окно программы содержит в себе область прокрутки и 3 пункта меню, соответствующие выделенным модулям.

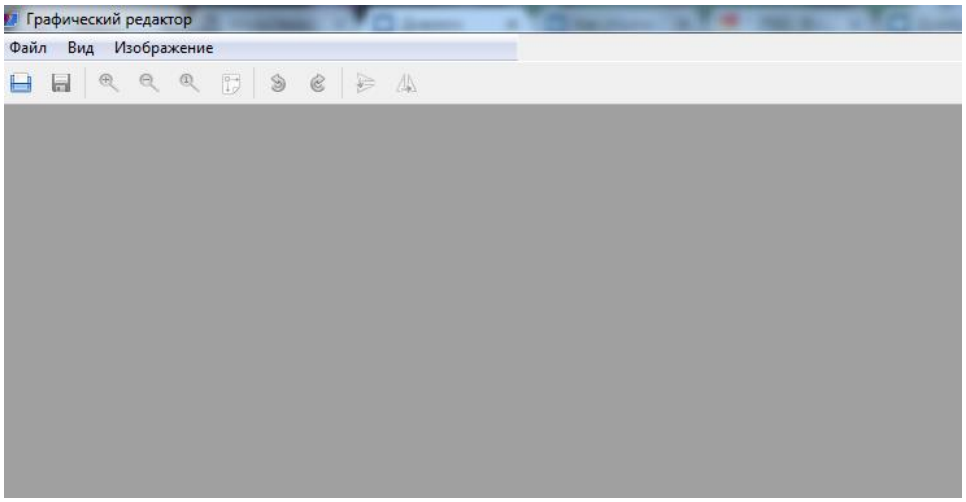

Рисунок 1- Главное окно программы цветокоррекции

После выбора пункта меню «открыть», изображение отображается по центру прокручиваемой области, подстраиваясь под размеры окна.

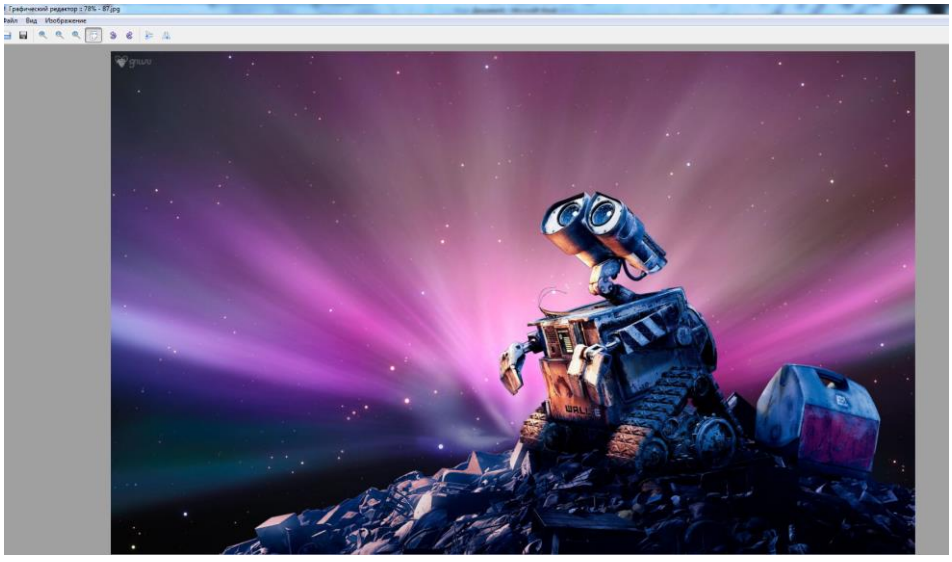

Рисунок 2- Загрузка изображения

При выборе пункта «цветокоррекция» пользователю предлагается выбрать изображение-донор.

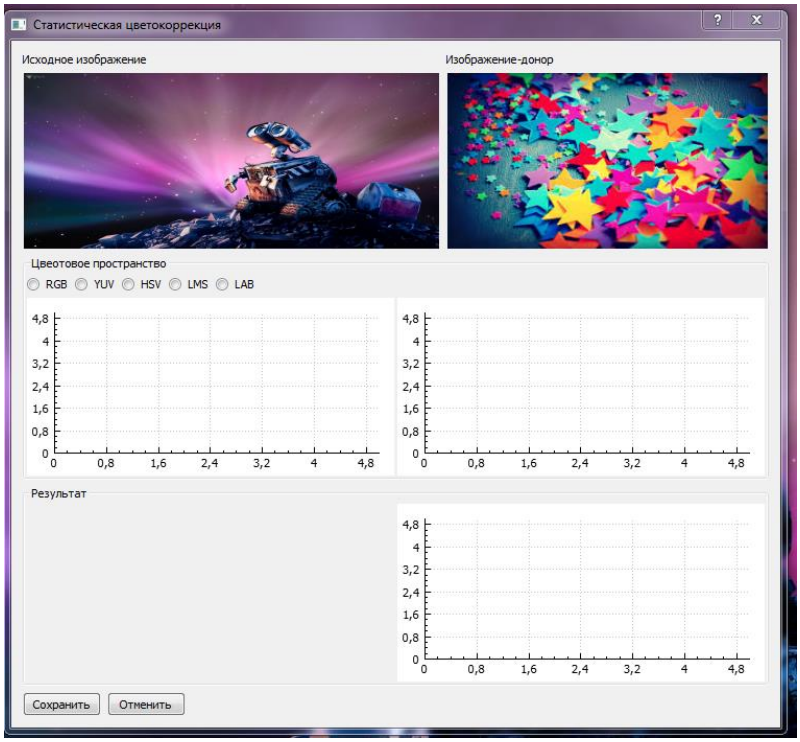

Рисунок 3 – Цветокоррекция: исходное изображение и изображение-донор.

После выбора двух изображений, пользователь может задать необходимую цветовую схему. Наиболее популярными на сегодняшний день являются следующие цветовые модели: RGB (используется в основном в мониторах и камерах), CMY(K) (используется в полиграфии), HSI (широко используется в машинном зрении и дизайне). Субтрактивная модель CMY (от англ. cyan — голубой, magenta — пурпурный, yellow — жёлтый) используется для получения твёрдых копий (печати) изображений, и в некотором роде является антиподом цветового RGB-куба. С целью унификации была разработана международная стандартная цветовая модель CIE XYZ. Модель CIE XYZ, хоть и наиболее близка к восприятию человека, но достаточно сложна в описании. В связи с этим, наибольшее распространение получило цветовое пространство **Lab** и его модификации, представляющее все видимые цвета и оттенки в виде шара с осями L, a и b. При этом по оси **L** измеряется светлота (в диапазоне от 0 до 100%), отображая коэффициент спектрального отражения, по оси **a** измеряется красный-зеленый оттенок, по оси **b** оттенок желтый-синий (в диапазонах от -120 до +120).

Алгоритм статистической цветокоррекции предполагает 4 вычисления для каждого цветового канала каждого пиксела двух изображений. При этом размер изображения может быть любым. Современные цифровые камеры имеют большое разрешение, что влечет за собой большое количество пикселей, большой размер изображения и, соответственно, много работы для алгоритма. Для проведения исследования были выбраны два изображения с одинаковым размером 2500x3750, что является средним размеров качественной фотографии.

Таким образом, необходимо обработать 2 изображения, в каждом из которых содержится 9.375.000 пикселей.

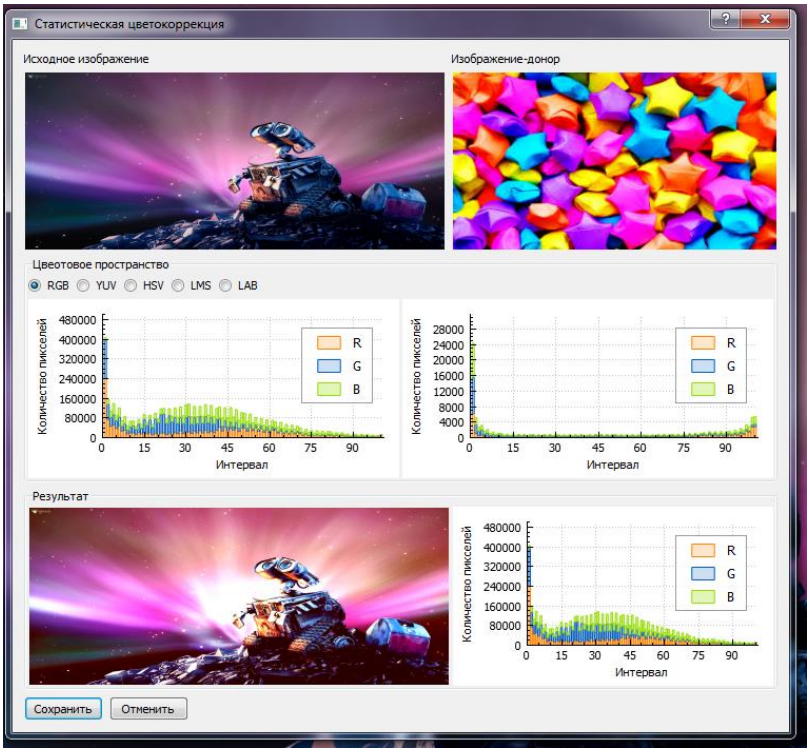

Рисунок 4 – Результат работы цветокоррекции.

Таким образом, был разработан пользовательский интерфейс, удовлетворяющий основным требованиям комфорта пользователя. В графическом интерфейсе используются стандартные элементы интерфейса, привычные пользователю, что позволяет сократить время на освоение программы. Предупреждающие сообщения снижают вероятность допущения ошибки или случайного несохранения внесенных изменений.

1 Список литературы

2 Грабалов, П.К. Компьютерная графика и основные графические редакторы / П.К. Грабалов – Калининград.: Аст, 2003 – 207 с.

3 Яхонтов, В.Н. Компьютерная графика / В.Н. Яхонтов – М.: ТИСБИ, 2003. – 320 с.

4 Косенко, П. Живая цифра / П. Косенко – М.: Тримедиа, 2013. – 286 с.

5 Иттен, Й. Искусство цвета / Й. Иттен – М.: Аст, 2001. – 95 с.

6 Крашенинников, В. Р. Основы теории обработки изображений / В.Р. Крашенинников – Ульяновск: УлГТУ, 2003. – 150 с.

7 Прэтт У. Цифровая обработка изображений / У. Прэтт – М.: Мир, 1982. Т. 1, 2. 791 с.

8 Фисенко, В.Т. Компьютерная обработка и распознавание изображений / В.Т. Фисенко, Т.Ю. Фисенко – СПб: СПбГУ ИТМО, 2008. – 192 с.

ЦИФРОВАЯ СТЕГАНОГРАФИЯ В ГРАФИЧЕСКИХ ФАЙЛАХ

Влацкая И.В., канд. техн. наук, доцент, Зубаиров С.И.
Оренбургский государственный университет

Стеганография – наука и искусство о скрытой передаче информации путем сохранения в тайне самого факта передачи. Термин ввел Иоганн Тритемий, ученый, теолог, алхимик и оккультист, в своем труде «Стеганография». Стеганография имеет свое место в обеспечении безопасности: она не исключает криптографию, а дополняет её. Информация, скрытая средствами стеганографии сильно снижает вероятность обнаружения факта самой передачи, а если скрываемое сообщение было еще и зашифровано, то это уже иной, более высокий уровень защиты.

Существуют несколько направлений стеганографии, а именно:

– Классическая стеганография;

– Компьютерная стеганография – одно из направлений классической стеганографии, использующее специальные свойства компьютерных форматов данных;

– Цифровая стеганография – одно из направлений классической стеганографии, основанное на сокрытии информации в цифровых объектах, приводящее к их искажению. Обычно эти искажения незаметны для человека, так как находятся за пределами чувствительности.

Последнее направление было выбрано в качестве рассмотрения в данной работе, а именно цифровая стеганография в изображениях.

Среди всех методов для сокрытия информации в изображениях выделяются следующие:

– Скрытие данных в пространственной области изображения: данные внедряются в области исходного изображения, поэтому отсутствуют сложные вычислительные операции;

– Скрытие данных в частотной области изображения: наиболее известные методы – на основе ДКП (дискретного косинусного преобразования) и вейвлет-преобразования, так как они используются при сжатии изображений с потерями. Использование метода с преобразованием, которому будет подвергаться изображение со временем, повышает его стойкость к искажениям. В данной работе применяется метод на основе ДКП, так как в дальнейшем изображение подвергается JPEG-компрессии, а именно метод относительной замены величин коэффициентов ДКП - метод Коха и Жао.

Суть метода Коха и Жао состоит в изменении отношения между абсолютными значениями коэффициентов ДКП в среднечастотной области изображения. На первом шаге мы должны разбить исходное изображение на блоки 8*8 пикселей, а именно выделить компоненту B (синий цвет), цветовой схемы RGB (см. рисунок 1).

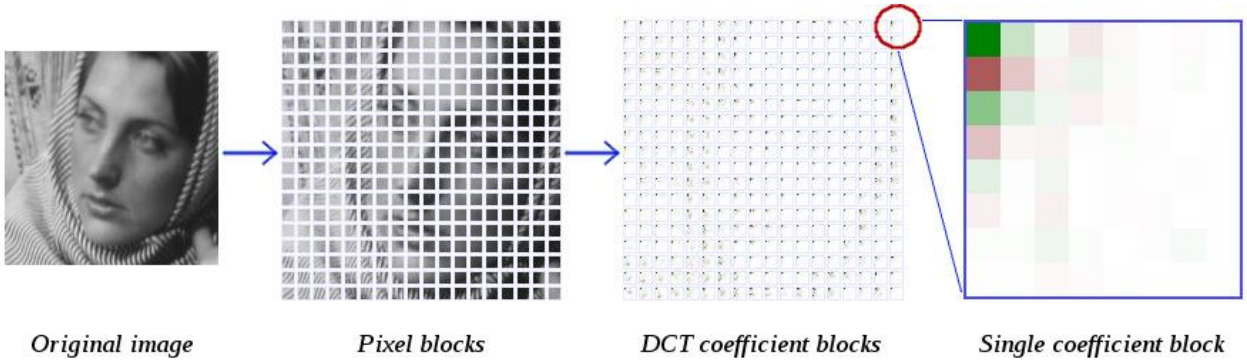

Рисунок 1 – Получение коэффициентов ДКП

Выбирается именно B-компонента пикселя, так как человеческий глаз менее чувствителен к изменениям синего цвета. Для получения коэффициентов ДКП применяется формула (см. рисунок 2).

$$\Omega(\upsilon,\nu)=\frac{\zeta(\upsilon)\cdot\zeta(\nu)}{\sqrt{2N}}\cdot\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}C(x,y)\cdot\cos\left[\frac{\pi\cdot\upsilon\cdot(2x+1)}{2N}\right]\cdot\cos\left[\frac{\pi\cdot\nu\cdot(2y+1)}{2N}\right]$$

Рисунок 2 – Формула получения коэффициентов ДКП

Где Ω(u,v) – коэффициент ДКП, C(x,y) – элемент оригинального изображения размерностью N*N; x,y – пространственные координаты пикселей изображения; u,v – координаты в частотной области; если x = 0 □(x) = 1/√2, иначе □(x) = 1.

Для восстановления B-компоненты используется формула (см. рисунок 3).

$$S(x,y)=\frac{1}{\sqrt{2N}}\cdot\sum_{\upsilon=0}^{N-1}\sum_{\nu=0}^{N-1}\zeta(\upsilon)\cdot\zeta(\nu)\cdot\Omega(\upsilon,\nu)\cdot\cos\left[\frac{\pi\cdot\upsilon\cdot(2x+1)}{2N}\right]\cdot\cos\left[\frac{\pi\cdot\nu\cdot(2y+1)}{2N}\right]$$

Рисунок 3 – Формула восстановления компоненты цвета пикселя из коэффициентов ДКП

Где S(x,y) – элемент восстановленного по коэффициентам ДКП изображения, Ω(u,v)– коэффициент ДКП; x,y – пространственные координаты пикселей изображения; u,v – координаты в частотной области; если x = 0 □(x) = 1/√2, иначе □(x) = 1.

Среднечастотная область выбирается, так как встраивание в ВЧ приводит к гарантированному разрушению при JPEG компрессии, а встраивание в НЧ

приводит к видимым искажениям изображения. На рисунке СЧ область выделена жирным шрифтом (см. рисунок 4).

-603	203	11	45	-30	-14	-14	**-7**
-108	-93	10	49	27	6	**8**	**2**
-42	-20	-6	16	17	**9**	**3**	3
56	69	7	-25	**-10**	**-5**	-2	-2
-33	-21	17	**8**	**3**	-4	-5	-3
-16	-14	**8**	**2**	-4	-2	1	1
0	**-5**	**-6**	-1	2	3	1	1
9	**5**	-6	-9	0	3	3	2

Рисунок 4 – Блок коэффициентов ДКП 8*8

Информация встраивается побитно, причем в один блок b из коэффициентов ДКП 8*8 встраивается только один бит m_b. Выбираются два коэффициента из СЧ области: $\Omega_b(u_1,v_1)$ и $\Omega_b(u_2,v_2)$. Для встраивания нуля добиваются разницы абсолютных значений, большей некоторого значения P, для единицы же – меньше значения –P (см. рисунок 5).

$$\begin{cases} \left|\Omega_b(\upsilon_1,\nu_1)\right| - \left|\Omega_b(\upsilon_2,\nu_2)\right| > P, \text{ при } m_b = 0; \\ \left|\Omega_b(\upsilon_1,\nu_1)\right| - \left|\Omega_b(\upsilon_2,\nu_2)\right| < -P, \text{ при } m_b = 1. \end{cases}$$

Рисунок 5 – Формулы встраивания бита информации

При извлечении информации исходим из неравенств (см. рисунок 6).

$$\begin{cases} m_b^* = 0, \text{ при } \left|\Omega_b^*(\upsilon_1,\nu_1)\right| > \left|\Omega_b^*(\upsilon_2,\nu_2)\right|; \\ m_b^* = 1, \text{ при } \left|\Omega_b^*(\upsilon_1,\nu_1)\right| < \left|\Omega_b^*(\upsilon_2,\nu_2)\right|. \end{cases}$$

Рисунок 6 – Формулы для извлечения бита информации

При запуске программы, пользователю доступны две функции: скрытие данных в изображении и извлечение данных из изображения (см. рисунок 7).

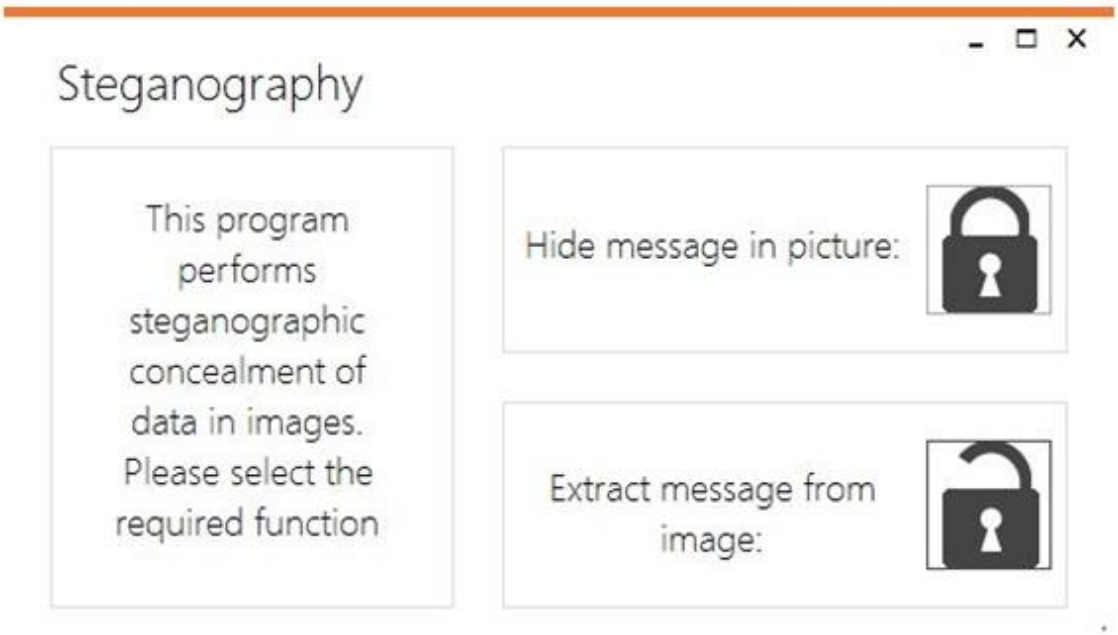

Рисунок 7 – Интерфейс программного средства

При скрытии данных в изображении, пользователь должен выбрать изображение в качестве контейнера для скрытия и ввести сообщение в текстовое поле, пометив окончание сообщения символом *. Под текстовым полем можно

увидеть текущую и максимальную длины сообщений для выбранного изображения. Также имеется возможность сохранения изображения и сброса установок (см. рисунок 8).

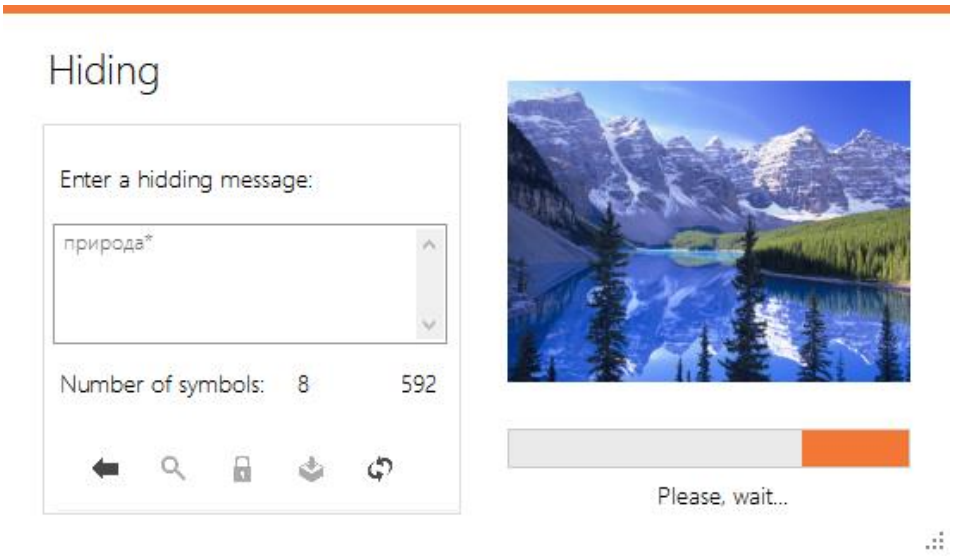

Рисунок 8 – Скрытие данных в изображении

При извлечении данных из изображения, пользователь выбирает изображение со скрытым сообщением. После чего можно запустить процесс извлечения. Также имеется возможность сброса установок (см. рисунок 9).

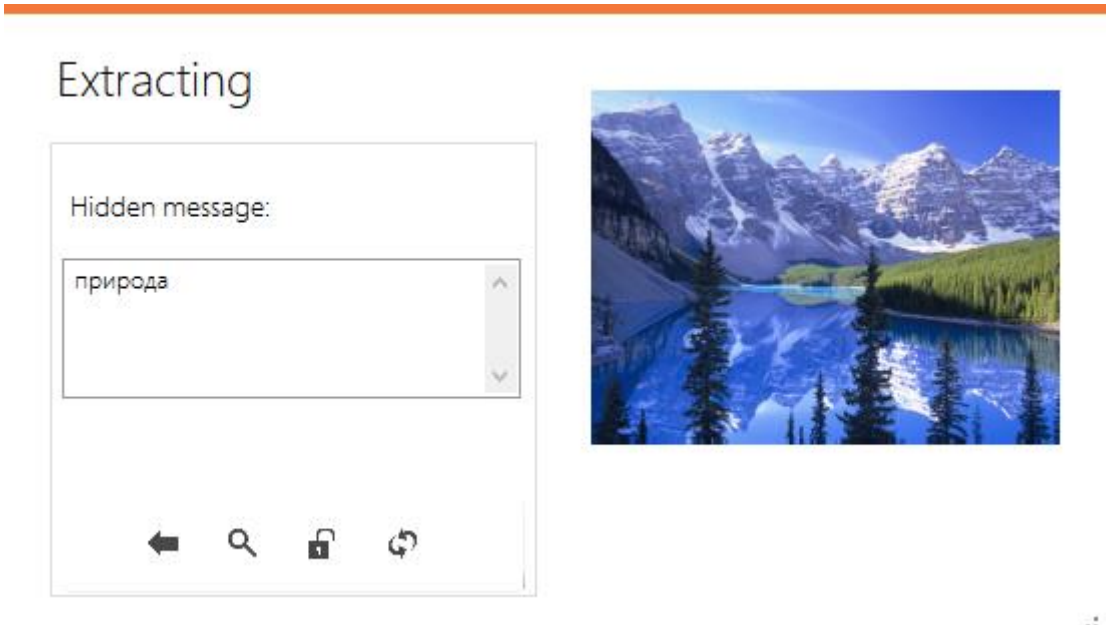

Рисунок 9 – Извлечение данных из изображения

Список литературы

1. *Кохановић Г.Ф., Пузыренко А.Ю.Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко; М.: МК-Пресс, 2006. — 288 с. ISBN: 966-8806-06-9.*
2. *Материалы сайта http://professorweb.ru*
3. *Материалы сайта https://msdn.microsoft.com*
4. *Материалы сайта https://habrahabr.ru*

АВТОМАТИЗИРОВАННАЯ ОЦЕНКА КАЧЕСТВА ПРОГРАММНОГО ПРОДУКТА

Влацкая И.В., канд. техн. наук, доцент,
Чумаков Р.В., Петров А.И.
Оренбургский государственный университет

Существуют различные критерии качества программного кода, которые исследуют его с качественной и количественной точки зрения. С их помощью программист может объективно оценивать написанный код и оптимизировать его, добиваясь наиболее лучших показателей различных критериев. Применение метрик программного кода позволяет разработчикам и руководителям проектов оценивать различные свойства создаваемого или уже существующего программного обеспечения, прогнозировать объем работ, давать количественную характеристику тех или иных проектных решений, качества разработанных систем и их частей, характеризовать сложность или надежность программного обеспечения [1].

Программное обеспечение, оценивающее качество других программных продуктов, в настоящее время достаточно новый вид программ, несмотря на то, что процесс создания ПО в России стал систематизироваться еще в 90-х годах XX века. Тогда были введены ГОСТ 28195-89 и ГОСТ 28806-90, определившие основные положения по оценке качества ПС и основные термины и определения, а также ГОСТ Р ИСО/МЭК 9126-93, который определяет 6 характеристик для оценки качества ПО: функциональные возможности, надёжность, практичность, эффективность, сопровождаемость и мобильность.

Автоматизация оценки качества ПС является актуальным направлением разработки программного обеспечения, поскольку позволяет сократить время на расчет количественных характеристик разрабатываемых или готовых программ в автоматическом режиме (при условии доступности исходного кода).

При выборе метрик для оценки качество необходимо придерживаться некоторых правил, которые позволят сделать оценку наиболее объективной и всесторонней [2].

Во-первых, метрики должны как можно меньше повторять друг друга, чтобы исключить влияние одного фактора на несколько метрик одновременно.

Во-вторых, метрики в своей совокупности должны как можно шире охватывать характеристики исходного кода программы.

Исходя из вышеизложенных правил, были выбраны следующие метрики [3,4]:

Метрика комментирования кода. Оценка проводилась по общепринятому правилу: качество комментирования кода прямопропорционально отношениию количества комментариев к количеству непустых строк программы. То есть чем ближе это отношение к 1, тем больше покрытие кода комментариями. Однако, это конечно не значит, что нужно комментировать каждую строку программы.

Метрика длины кода (SLOC). Оценка целесообразна при сравнении разных алгоритмов, решающих сходные/одни и те же задачи. Однако, здесь также необходимо учитывать обращение к внешним ссылкам, использование различных методов программирования, скорость выполнения кода, и многие другие факторы. Поэтому метрика SLOC играет второстепенную роль при оценке качества ПО.

Метрика Холстеда. Избыточность кода оценивается с помощью исследования идентификаторов и операторов ПО и последующего сравнения этих показателей с некоторыми теоретическими (идеальными) показателями. Соответственно, близость значений к теоретическим и определяет избыточность (чем ближе к теоретическим значениям, тем меньше избыточность).

Метрика Джилба. Как правило, сложность алгоритма оценивается как сложность исполнения программы, то есть программа со сложными алгоритмами включает в себя большое количество циклов и иных передач управления, а также использование разнообразных структур данных. Метрика Джилба предоставляет возможность подсчитать количество операторов передачи управления, а также их вложенность друг в друга.

Цикломатическое число Мак-Кейба. Отражает количество различных путей выполнения программы, то есть подсчитывает количество разных проходов по всем условиям и циклам в программе. Если интерпретировать программу как конечный автомат, то цикломатическое число отражает количество путей из начального состояния в конечное. Соответственно, чем больше это число, тем больше сложность программы.

Мера Берлингера. Оценка энтропии исходного кода носит несколько другое назначение, чем все предыдущие метрики. Она показывает возможности эффективного кодирования исходного текста программы (например методом оптимального кодирования Хаффмана). Это может быть полезно при сжатии исходного кода для хранения/передачи.

Разработанная программа оценки качества ПО (Program Sourse Quality Appraiser, PSQA) работает в операционной системе Windows не ниже версии 7. Для работы необходим .NET Framework 4.5.2 (и выше) – пакет стандартных библиотек, предоставляемый бесплатно корпорацией Майкрософт (Microsoft). Программа имеет графический интерфейс, который представлен на рисунке 1.

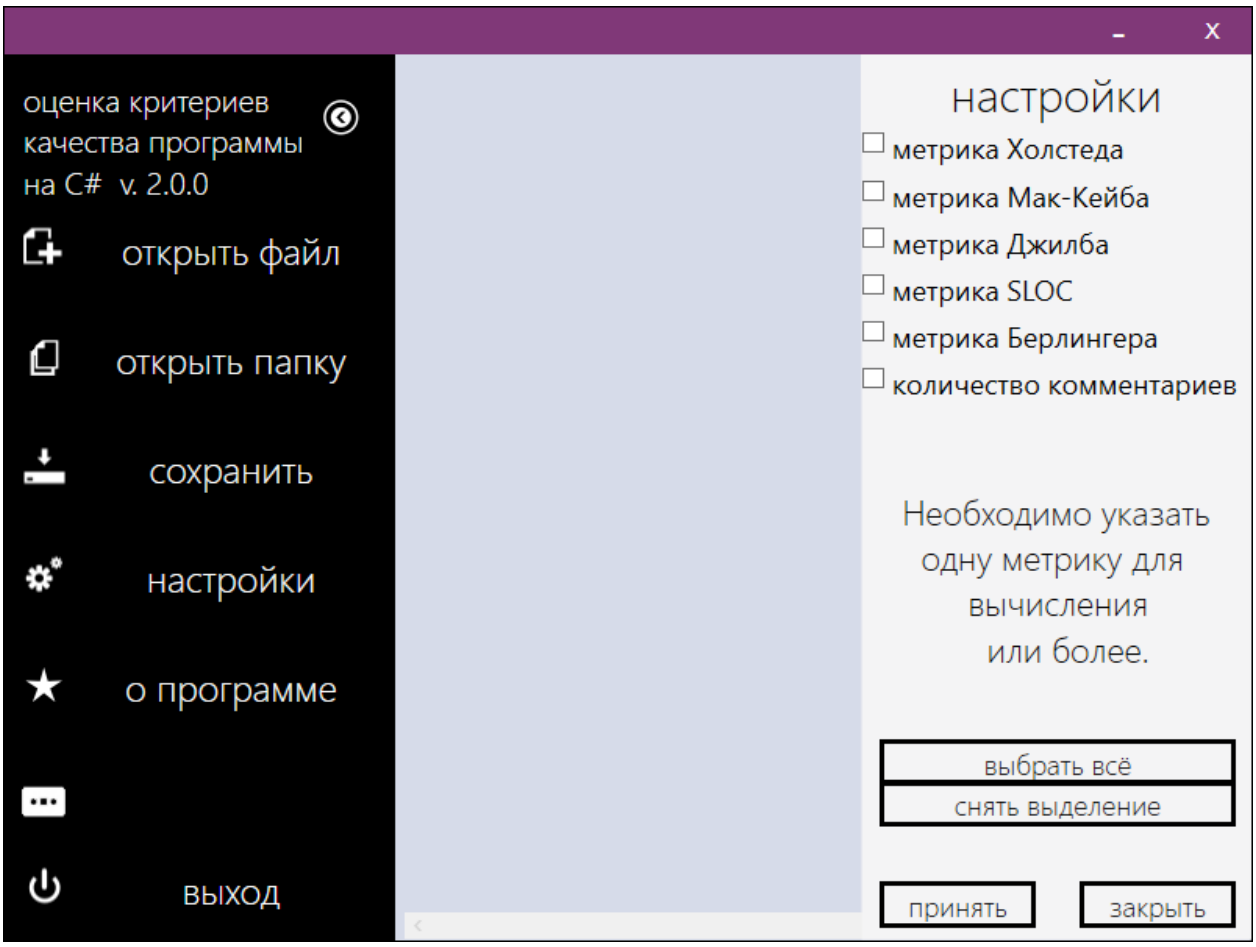

Рисунок 1 – Интерфейс программы PSQA.

Данная программа предоставляет инструменты как для оценки решения в целом (или группы решений), так и для оценки отдельных модулей программы (например для выявления низкого качества определённой составляющей программы).

Исходный код программы разбит на автономные блоки, каждый из которых реализует вычисление определённой метрики или выполняет иные преобразования кода (например, структурирует обфусцированный код), что позволяет использовать её части в других проектах.

Несмотря на то, что программа в автоматическом режиме вычисляет качество программного продукта, пользователю данной программы необходимо выработать свои собственные нормы качества, так как они сильно зависят от области применения исследуемых программных продуктов, от штата сотрудников и стиля написания кода каждого из них. Например, существует концепция, предполагающая комментирование только для обозначения неправильно написанного кода (если код непонятен без комментариев, значит он не удобен и его

можно улучшить). Поэтому в разных компаниях и отраслях разработки ПО одни и те же метрики могут иметь разные нормированные значения.

В процессе разработки программы и исследования метрик были выявлены особенности влияния времени на объективность показателей метрик. Это влияние вызвано прежде всего тем, что программы постоянно усложняются, поэтому многие показатели, считавшиеся раньше оптимальными, теперь отличаются в ту или иную сторону. Этому также способствует нагруженность программы логикой, описанием интерфейсов и другими специфическими аспектами разработки в той или иной области.

Представленный программный продукт может быть использован в различных отраслях разработки программного обеспечения для оценки качества и выявления перспектив улучшения исследуемых программ.

Список литературы

1. *Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011 – 247 с.*

2. *Влацкая И.В., Заельская Н.А., Надточий Н.С.,Проектирование и реализация прикладного программного обеспечения: учебное пособие/ И.В. Влацкая, Н.А. Заельская, Н.С. Надточий; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2015. – 118 с.*

3. *Myers, G., “An Extension to the Cyclomatic Measure of Program Complexity”, SIGPLAN Notices, October 1977*

4. *T.J. McCabe, "A complexity measure," IEEE Transactions on Software Engineering, vol. SE-2, no. 4, pp. 308-320, December, 1976*

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПОКРЫТИЯ ТЕСТАМИ ПРОГРАММНОГО ПРОДУКТА

Влацкая И.В., канд. техн. наук, доцент,
Побежимова Е. В., Секретева А. Д.
Оренбургский государственный университет

В данной статье рассматривается один из подходов к автоматизированной оценки тестового покрытия программного продукта при функциональном тестировании основных алгоритмических конструкций.

Под тестированием программного обеспечения понимают проведение испытаний работоспособности программы с целью обнаружения потенциальных ошибок и недекларированного поведения системы на конечном наборе тестов. Следует отметить, что тестирование - обязательный этап разработки программного продукта. Оно тесно связано с проектированием и разработкой, поскольку может дать ответ на некоторые вопросы о качестве создаваемого продукта. Безусловно, программисты с высоким уровнем профессионализма владеют множеством приёмов тестирования, знают о различных его стратегиях. Но применение всех этих знаний – довольно трудновыполнимая задача. При попытках составить тесты вручную в дело может вступить такой аспект, как человеческий фактор, и возникает риск, что специалист-тестировщик упустит из внимания тот или иной участок кода. И поэтому особо остро встает вопрос об автоматизированной генерации тестовых наборов для программного продукта.

Существуют различные подходы к созданию тестов. Сгенерировать можно как случайные наборы, так и наборы для целенаправленного тестирования тех или иных структур программы. Случайный набор тестов является худшим из вариантов для тестирования кода, поскольку невозможна гарантия проверки работоспособности программы в разных состояниях. При осмысленном тестировании возможны два варианта стратегии проектирования тестов: интеграционное и модульное [1].

Интеграционное тестирование - это тестирование части системы, состоящей из более, чем одного модуля. Смысл такого тестирования заключается в поиске дефектов кода, которые могли возникнуть вследствие недочетов в реализации, и проверке программы на соответствие внешним спецификациям программ и модулей. Данный подход также называется стратегией «черного ящика».

Модульное тестирование - это тестирование отдельно взятых модулей, функций, и т.д. В этом случае составление тестов основывается на знании внутренней структуры программы, на необходимости проверки каждой ветви алгоритма, прохождения каждого возможного участка кода. При этом внешняя спецификация, как ясно из определения, во внимание не принимается. Такое

тестирование, также носящее название «белого ящика», как правило, применяют на ранних стадиях создания программного обеспечения для выявления и дальнейшего устранения преимущественно алгоритмических недочетов. Именно поэтому на этапе модульного тестирования проще всего найти ошибки в написании кода алгоритмов (например, неверная работа с условиями и счетчиками циклов).

В основу разрабатываемой автоматизированной системы оценки покрытия тестами программного продукта ляжет как раз именно тестирование по принципу «белого ящика». Это обосновано тем, что целью является исследование внутренней структуры программного средства – его кода. Иными словами, на выходе должны получиться тестовые наборы, гарантирующие проверку всех независимых маршрутов программы: прохождение ветвей true-false для всех логических решений, выполнение циклов и прохождение граничных значений.

В силу большой трудоемкости и многогранности тестирования программных продуктов в данной системе были введены следующие ограничения:

- язык программирования C#;
- *.cs-файлы должны содержать код консольного приложения;
- код не подразумевает наличие таких типов данных, как массивы, структуры, перечисления, классы, интерфейсы, делегаты, события, и т.д.;
- в функции не передаются никакие параметры - система учитывает только те значения переменных, которыми были инициализированы переменные внутри функций;
- система сильно ограничена в типах данных - она работает только с переменными типа string, int, double;
- структура кода состоит из объявлений переменных и условных конструкций if-else, switch и while;
- переменные, для которых определяются тестовые наборы, должны инициализироваться при помощи операции консольного ввода Console.ReadLine().

При открытии программы пользователь увидит окно ввода C#-кода или загрузки *.cs-файла (рисунок 1). Эти 2 пункта взаимоисключают друг друга, т.е. при выборе «Загрузить файл» вкладка «Ввести данные» автоматически закроется, а содержимое внутри очистится.

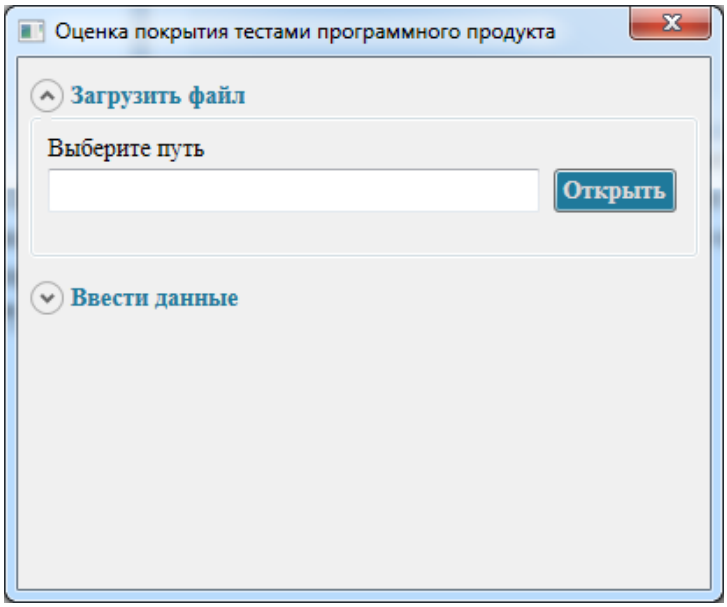

Рисунок 1 – Окно ввода C#-кода или загрузки *.cs-файла

Выбрав вкладку «Ввести данные», пользователь сможет в открывшееся текстовое поле вставить или написать свой код (рисунок 2).

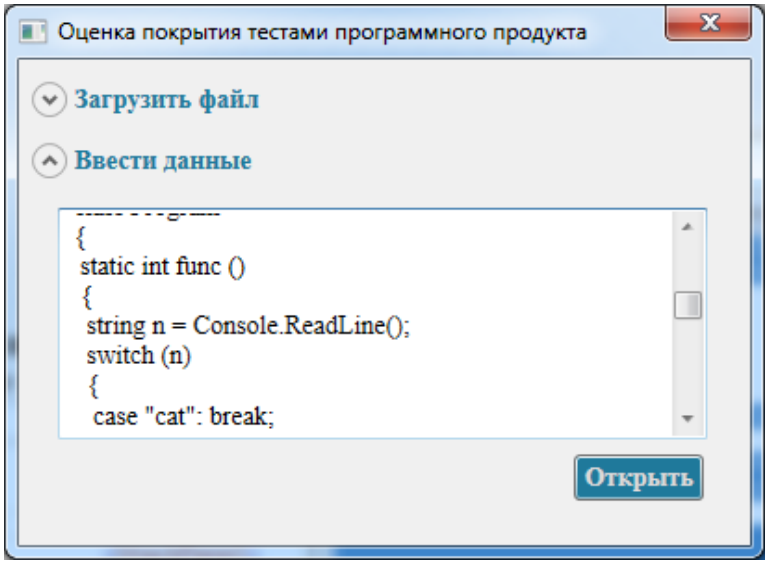

Рисунок 2 – Ввод данных вручную

Нажав на кнопку «Открыть», пользователь будет перенаправлен на окно результатов анализа кода (рисунок 3), а данные переданы системе. После того, как система получила текст C#-кода (введенный в текстовом поле, или же из подгруженного *.cs-файла), она начинает его декомпозицию, вычленяя объявленные методы. Система перебирает строки кода до тех пор, пока не находит строку с синтаксисом типа <модификатор_доступа>*

<объявление_статического_метода>* <тип_метода> <имя метода> (<передаваемые_аргументы>*), где * означает возможное отсутствие в строке. Затем все тем же перебором строк кода определяется тело метода, ограниченное фигурными скобками; при этом учитывается, что внутри метода есть конструкции, чьи границы также обозначаются скобками. Решается эта проблема подсчетом открывающих и закрывающих скобок. На следующем этапе нужно определить переменные, используемые в каждом методе, а также способ их инициализации (ввод с консоли, или же предопределенное кодом программы значение). Здесь уже идет перебор строк тела метода, где система обращает внимание на строки, которые выглядят как <тип_переменной> <имя_переменной_и_ее_инициализация>+;, где + - одно и больше повторений. Такие строки подвергаются детальному разбору, где выявляется, сколько переменных объявлено и какие значения им присваиваются.

В системе определен класс Variables, включающий в себя такие поля, как <имя_переменной>, <тип_переменной>, <значение_переменной>. В специальный список типа Variables заносятся все найденные на предыдущем этапе переменные и их характеристики; при этом, если значение задано в коде, то в список вносится именно оно, а если предполагается, что пользователь должен вводить значение с клавиатуры, ставится пометка «by_user». В свою очередь, такие списки переменных имеет каждый экземпляр класса Procedure, помимо этого содержащий имя метода и его тело. Все методы и их характеристики помещаются в список типа Procedure для упрощения дальнейшего анализа, который начинается после этапа декомпозиции кода.

Анализ представляет собой проход по каждому методу из созданного ранее списка, где, в свою очередь, в теле метода определяется наличие конструкций if-else, while и switch. Когда система находит такие конструкции, то вызывает соответствующую функцию, которая должна проанализировать уже их.

В случае с оператором управления switch система проверяет выражение, передающееся в конструкцию и сравнивает его со списком переменных, определенным в том же методе, что и оператор. Исходя из типа переменной, совпадающей со switch-выражением, ее значения и значений меток case, система генерирует возможные варианты для переменной – как совпадающие с case-метками, так и те, что не совпадают, для прохода всех возможных путей кода.

Когда в коде встречаются условные операторы if-else, функция анализа проверяет условие, записанное в скобках после ключевого слова if. Условие

разбивается на два операнда и стоящий между ними знак. Каждый операнд, как и в случае, описанном выше, проверяется на наличие в списке переменных метода и на принадлежность типу данных. Когда система понимает, что, с чем и как она сравнивает, она предлагает свои варианты значений операндов, исходя, опять же, из принципа, что пройти нужно все варианты путей кода.

После того, как для каждой переменной в коде, чье значение подразумевает ввод с консоли, были определены ее возможные значения, система комбинирует все варианты между собой, получая таким образом тестовые наборы для заданного программного C#-кода.

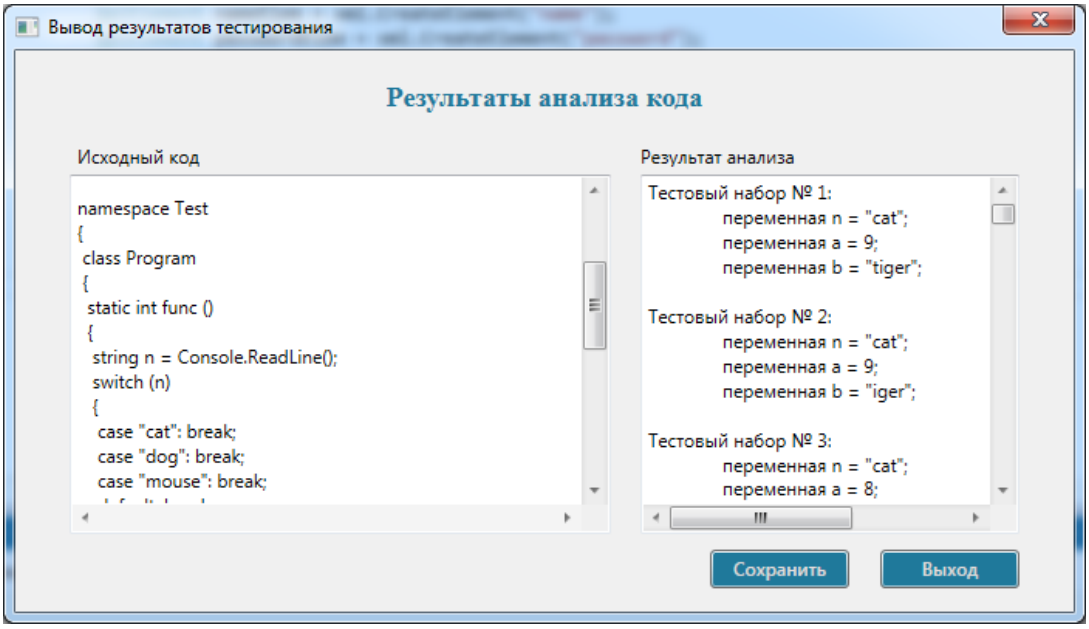

Рисунок 3 – Окно результата анализа кода

Для удобства пользователя полученный результат анализа кода можно сохранить в файле формата *.txt.

При выполнении сохранения пользователь получит окно подтверждения (рисунок 5).

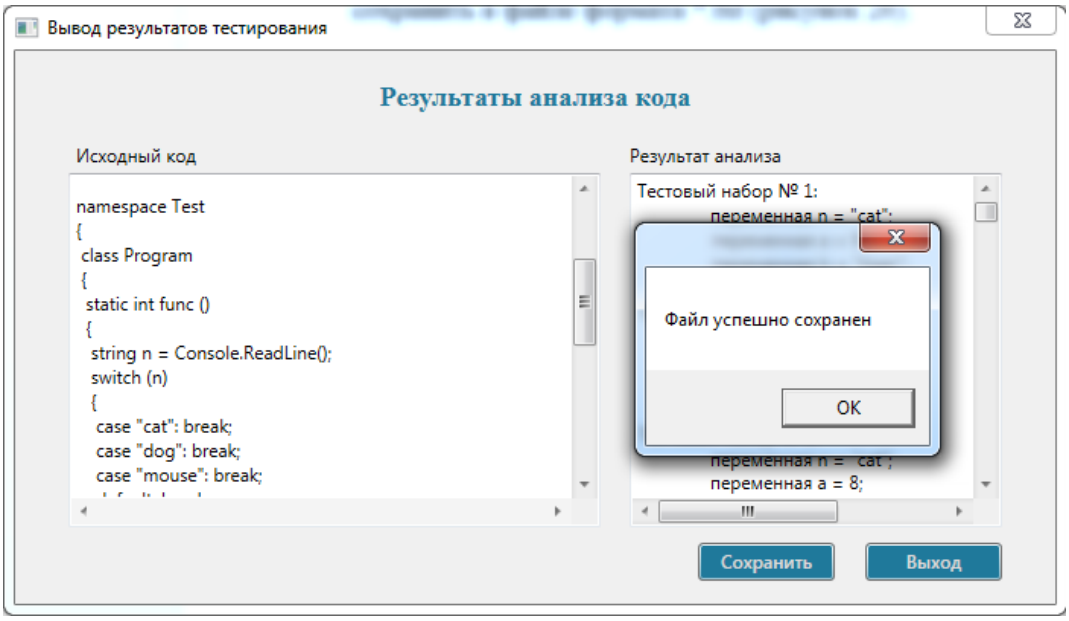

Рисунок 5 – Подтверждение сохранения файла

Аналогичные действия происходят и при загрузке *.cs-файла.

Список литературы

1. *Майерс Г. Искусство тестирования программ. – М.: Финансы и статистика, 1982. – 176 с*

2. *An Introduction to Software Testing [Электронный ресурс] /. — Электрон. текстовые дан. — Режим доступа: /~zyl/articles/testing_intro.pdf, свободный.*

КОММУНИКАЦИОННЫЙ ИНТЕРФЕЙС ДЛЯ ПРОМЫШЛЕННОГО ИНТЕРНЕТА ВЕЩЕЙ

Влацкая И.В., канд. техн. наук, доцент, Пономарева Н.Н.
Оренбургский государственный университет

Одним из перспективных направлений в развитии промышленности является внедрение промышленного интернета вещей на производстве.

Промышленный интернет вещей (Industrial Internet of Things, IIoT) – это многоуровневая система, которая включает в свой состав различные контроллеры и датчики, установленные на конкретных устройствах и установках промышленного объекта, средства, передающие собираемые данные, и средства, визуализирующие эти данные, инструменты для аналитики и интерпретации получаемой информации и другие составляющие. [1]

Внедрение промышленного интернета вещей имеет множество преимуществ: начиная от мониторинга за производством в режиме реального времени и заканчивая полностью автоматизированным производством, позволяющим исключить вмешательства человека. А потребитель может отследить всю цепочку производства продукта или устройства, которое он держит в руках.

Не все предприятия видят явную выгоду от внедрения промышленного интернета вещей. Распространено мнение, что IIoT подразумевает хранение всех данных в открытом виде и он совершенно не защищен, что несет только лишние угрозы для предприятий.

Промышленный интернет вещей является безопасным и его внедрение на предприятие несет максимум выгоды и минимум опасности. Для того чтобы продемонстрировать преимущества IIoT необходимо спроектировать и разработать коммуникационный интерфейс. Он покажет, каким образом осуществляется обмен данными между промышленными устройствами или оборудованием внутри многоуровневой системы. Таким образом, решение будет наглядным, но не требует больших затрат для демонстрации.

Поскольку промышленный интернет вещей на российском рынке достаточно новое решение, ознакомиться с тем, как устроены внутри платформы IIoT, представленные на рынке, не представляется возможным. Поэтому в сравнении аналогов рассматриваются именно платформы для промышленного интернета вещей, но критерии, по которым они сравниваются, взяты именно в отношении коммуникационных интерфейсов. [2]

В таблице 1 приведено сравнение существующих решений.

Таблица 1 - Сравнение существующих решений

Технология / Критерий	«Стриж»	Tarantool IIoT	ThingsPro

Промышленная облачная платформа	+	+	+
Безопасность	+	-	+
Интеграция в существующие решения	-	+	-
Резервное копирование данных	+	+	-
Удаленный доступ к системе	-	+	+

Промышленная облачная платформа – это то, без чего не может существовать IIoT. Она подразумевает три составляющих:

- подключение к различным устройствам и системам;
- аналитика больших данных;
- разработка собственных приложений.

В рамках данной работы внимание уделено аналитике больших данных.

Под безопасностью понимается защищенность системы в целом от всех возможных кибер-угроз.

Функции системы ориентированы на интеграцию с существующими решениями на предприятии, что позволяет использовать часть готовых решений, а не заново разрабатывать систему.

Резервное копирование данных – необходимая функция, позволяющая в случае успешно проведенной атаки на предприятие иметь возможность восстановить утерянные данные.

Удаленный доступ к системе подразумевает возможность подключения к системе не только с рабочей станции, но и с телефона, если есть такая необходимость.

Таким образом, разрабатываемый интерфейс должен иметь в своем составе облачную платформу для возможности аналитики больших данных, собираемых с предприятия, быть защищенным от всех известных угроз и от угроз, выявленных в модели нарушителя, разработанной для конкретной системы, иметь возможность интеграции в существующие системы на предприятии; обеспечивать резервное копирование данных в хранилище, которое будет определяться с директором предприятия, для которого разрабатывается данный интерфейс, позволять удаленно подключаться для возможности своевременного управления.

Согласно документу [3], изданному Консорциумом промышленного Интернета, сопоставление эталонной трехуровневой архитектуры и функциональных модулей для промышленного интернета вещей представлено на рисунке 1.

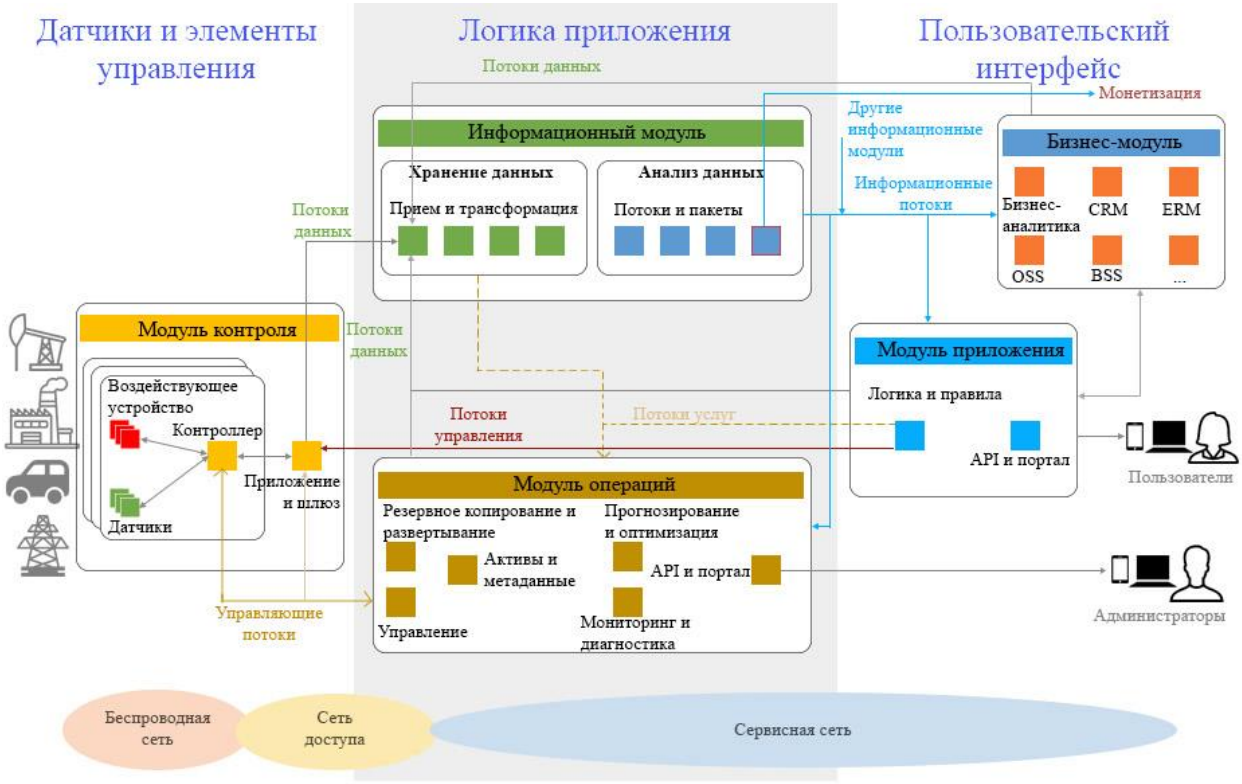

Рисунок 1- Сопоставление трехуровневой архитектуры и функциональных модулей

Таким образом, вся система должна разделяться на три уровня - уровень датчиков и элементов управления, уровень логики приложения и пользовательский интерфейс. Для функционирования всей системы должны быть реализованы все модули, представленные на рисунке. Для достижения цели данной работы необходимо реализовать только некоторые из них:

- модуль приложения, реализующий пользовательский интерфейс;
- информационный модуль, реализующий хранение и анализ данных;
- модуль операций, ответственный за резервное копирование и мониторинг системы.

На примере виртуального предприятия рассмотрим проектирование и разработку коммуникационного интерфейса IIoT.

«Предприятие «XXX» осуществляет свою деятельность в области производства метизов. «XXX» занимает площадь в 50 квадратных метров. На территории данного предприятия располагаются три производственных цеха, склад комплектующих и готовой продукции.

Предприятие было построено в 1982 году. К нынешнему времени 80 процентов оборудования отслужило почти 35 лет. Из-за регулярных поломок и простоев оборудования предприятие несет убытки, что не позволяет экономить средства на обновление станков.»

В этом случае основной задачей является мониторинг за состоянием устройств и оборудования предприятия в режиме реального времени, возмож-

ность дистанционного доступа к системе мониторинга и получение сообщений о достижении показаний датчиком критических значений.

Для достижения поставленной цели и реализации всех необходимых компонентов было решено использовать открытую платформу для организации управления облачной инфраструктурой и виртуальными окружениями OpenNebula. [4] Данная платформа позволяет оперировать вычислительными мощностями серверов, необходимых для аналитики больших данных, пространством для хранения и резервирования баз данных, виртуальными машинами. Необходимо спроектировать и разработать WEB-сервис. Он будет выступать в качестве коммуникационного интерфейса IIoT. Для доступа к WEB-интерфейсу по сети Интернет разместим его на сервере. В качестве сервера будет выступать виртуальная машина с операционной системой CentOs 7, развернутая в программе VirtualBox. Разместим данную виртуальную машину в облаке OpenNebula. Это даст возможность контролировать состояние Web-сервиса. Для решения ряда задач, связанных с WEB-сервисом выбран язык программирования Python и фреймворк Django. Данный фреймворк предусматривает защиту от межсайтового скриптинга (XSS), защиту от подделки межсайтового запроса (CSRF), защиту от внедрения SQL, использование защищенного соединения между клиентом и сервером. [5]

В результате проделанной работы было развернуто и настроено на локальной машине облако OpenNebula, в котором создана и настроена виртуальная машина. На данную виртуальную машину установлен и запущен Web-сервис.

Список литературы

1 What Is the Industrial Internet of Things (IIoT) Platform? [Электронный ресурс]. - Режим доступа: http://blog.lnsresearch.com/iiot-platform - 25.12.2017.

2 Критерии АСУ для настоящих IIoT [Электронный ресурс]. - Режим доступа:http://industry4-0-ukraine.com.ua/2017/06/07/критерии-асу-для-настоящих-iiot/ - 25.12.2017.

3 The Industrial Internet of Things Volume G1: Reference Architecture. [Электронный ресурс]. - Режим доступа: https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf - 25.12.2017.

4 OpenNebula - платформа для организации управления cloud-инфраструктурой и виртуальными окружениями [Электронный ресурс]. - Режим доступа: http://pro-spo.ru/cloud-technology/2513-opennebula - 25.12.2017.

5 Безопасность в Django [Электронный ресурс]. - Режим доступа: https://djbook.ru/rel1.4/topics/security.html - 25.12.2017.

РАЗРАБОТКА ВЫСОКОНАГРУЖЕННОЙ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

Влацкая И.В., канд. техн. наук, доцент, Пятаева Е.В.
Оренбургский государственный университет

С каждым годом неуклонно возрастает число пользователей услугами сети Интернет. Востребованные веб-приложения за короткий срок набирают высокую посещаемость, в результате чего у таких систем могут возникать проблемы с отказоустойчивостью. В результате встает задача усовершенствования аппаратных ресурсов, поиска новых подходов для организации взаимодействия с базами данных и разработки оптимального программного кода.

В настоящее время нет устоявшегося определения высоконагруженной системы, но из существующей литературы можно выделить некоторые особенности, присутствующие у таких систем. Обычно высоконагруженные системы реактивные - это значит, что они не просто должны произвести вычисления, а на полученный запрос отправить ответ. Они должны делать это быстро, и если в такой системе что-то ломается пользователи не должны этого замечать. Нагрузки у таких систем обычно начинаются с 1000 RPS/QPS (Requests per second/Query per second) и 99% всех запросов должны получать ответ не более чем за 300мс. Еще одной особенностью таких систем является высокая утилизация ресурсов (50%-70%). Высоконагруженными ресурсами, в первую очередь, выступают многопользовательские приложения, многие из которых являются распределенными системами, работающими более чем на одном сервере. Таким образом высоконагруженные системы – это системы безостановочного доступа, т.е. те структуры, запрос данных которых дает возможность получать информацию без длительного перерыва при постоянной работе [1].

Актуальность данной работы состоит в том, что возрастает значимость информационных технологий в современном мире, соответственно возрастает численность посетителей веб-приложений и вопрос обеспечения отказоустойчивости при высоких нагрузках встает на передний план. Целью данной работы является построение типового приложения с высоконагруженной распределенной архитектурой.

Были определены требования к разрабатываемой системе. Разрабатываемая система должна выдерживать высокие нагрузки, то есть обеспечивать отказоустойчивость. Будем считать, что система выдерживает нагрузку, если она не выходит из строя при 1000 запросов в секунду. Система должна иметь подсистему управления доступом. Для защиты разрабатываемой системы следует реализовать защитные механизмы: аутентификация пользователей в системе, разграничение доступа пользователей к ресурсам и авторизация пользователей, регистрация и оперативное оповещение о событиях, защита от SQL-инъекций, защита от CSRF-инъекций, защита от XSS-атак. Для этого важное значение имеет использование надежных технологий, в которых уже реализована часть

защиты от основных атак. Также безопасность будет обеспечиваться за счет использования контейнерной технологии Docker.

Проанализировав типовую архитектуру высоконагруженных систем и микросервисную архитектуру, было принято решение объединить данные подходы. На рисунке 1 представлена архитектура для разрабатываемой системы. Основные причины для использования микросервисов – это аппаратные преимущества, недостижимые с помощью единой архитектуры, скорость и простота разработки одного узла системы, удобное и эффективное тестирование [2].

Чтобы повысить безопасность, а также для автоматизации механизма доставки (Continuous Delivery), развертывания (Continuous Deployment), непрерывного тестирования (Continuous Testing) и управления разрабатываемого приложения будем использовать открытую платформу для разработки, доставки и эксплуатации приложений Docker. Основные преимущества использования Docker контейнеров это легкость в обновлении и использовании образов, распределение ресурсов, изолированные среды выполнения и простота масштабирования [3].

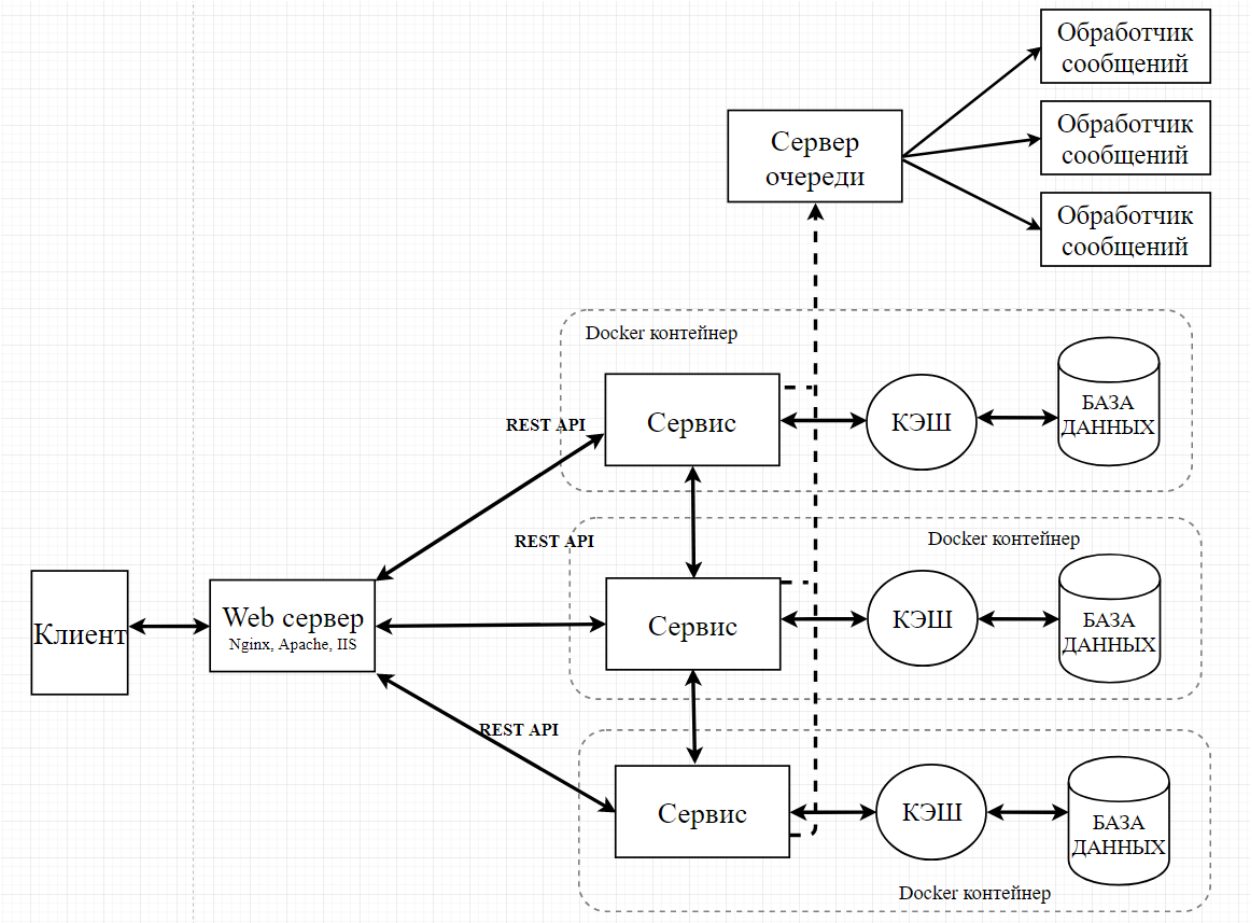

Рисунок 1 – Микросервисная архитектура высоконагруженной системы

Каждый из сервисов запускается, как отдельное приложение через Docker контейнер и не знает о существовании других, общение между ними происходит через REST.

Поскольку задача логирования является достаточно важной, и при одновременном использовании приложения несколькими десятками или сотнями клиентов, все сообщения должны быть обработаны и сохранены, то необходимо использовать очередь сообщений, чтобы избежать выполнения задачи сохранения логов непосредственно после отправки запроса. Для этого была реализована очередь сообщений посредством AMQP-брокера RabbitMQ.

Очень важной частью нашей работы является нагрузочное тестирование, поскольку разрабатываем высоконагруженную систему. Чтобы определить, является ли разработанное приложение отказоустойчивым, необходимо провести нагрузочное тестирование. Нагрузочное тестирование (Load Testing) или тестирование производительности – это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком-либо общем ресурсе [4].

В качестве инструмента для проведения нагрузочного тестирования использовали Apache JMeter, разрабатываемый Apache Software Foundation. В таблице 1 приведена зависимость времени отклика от количества пользователей, данные для которых были получены использую программу JMeter.

Таблица 1 – Таблица зависимости времени отклика от количества пользователей.

Кол-во одновременных пользователей	Минимальное время отклика, мс	Максимальное время отклика, мс	Среднее время отклика, мс
1	21	64	24
25	34	71	42
100	35	102	48
1000	38	332	97
10000	45	2845	922

На рисунке 2 представлено время отклика при пользовании приложения 1000 пользователями.

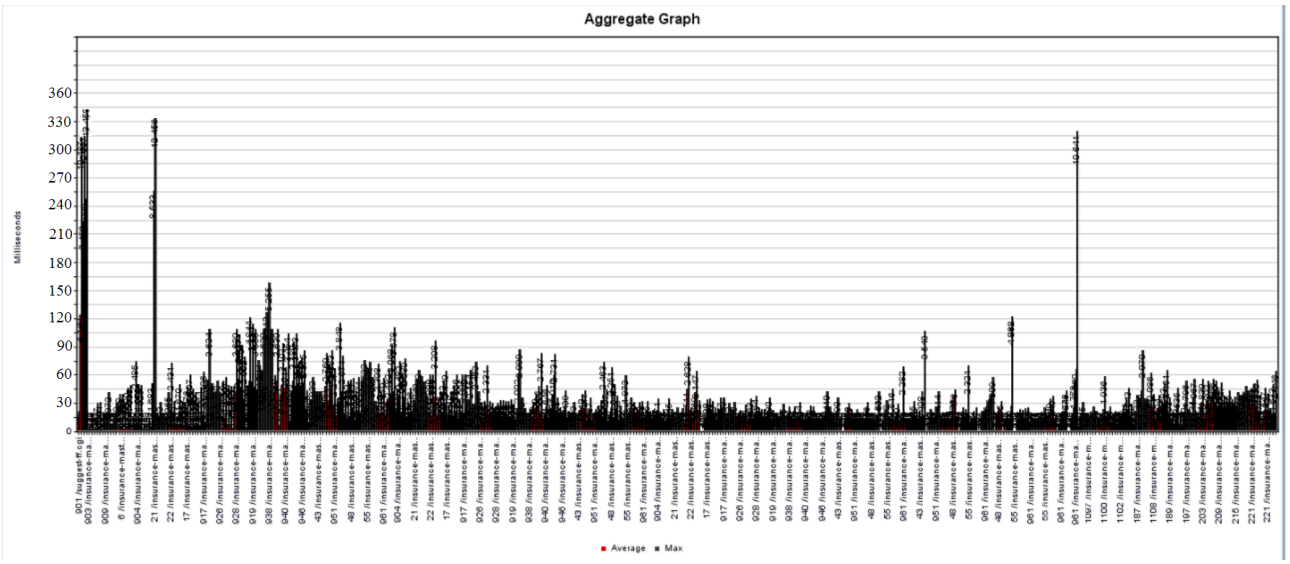

Рисунок 2 - Время отклика при 1000 одновременных запросов

Таким образом, были получены результаты, удовлетворяющие требованиям. Система при максимальной нагрузке в 10000 пользователей не вышла из строя, а максимальное время отклика при этом составило 2845 мс. Отметим, что мы запускали все контейнеры на одном сервере локально, поскольку не было возможности запускать каждый контейнер на отдельном сервере. В последнем случае показатели были бы значительно выше за счет выделения каждому сервису большее количество ресурсов.

Для мониторинга ресурсов серверной части приложения под нагрузкой использовали jvisualvm (Java VisualVM). Эта утилита находится в составе JDK и позволяет снимать дампы, анализировать производительность, состояние потоков и памяти. На рисунке 3 показана нагрузка на ЦП, на рисунке 4 – количество используемой памяти, на рисунке 5 – количество потоков при 1000 различных запросов.

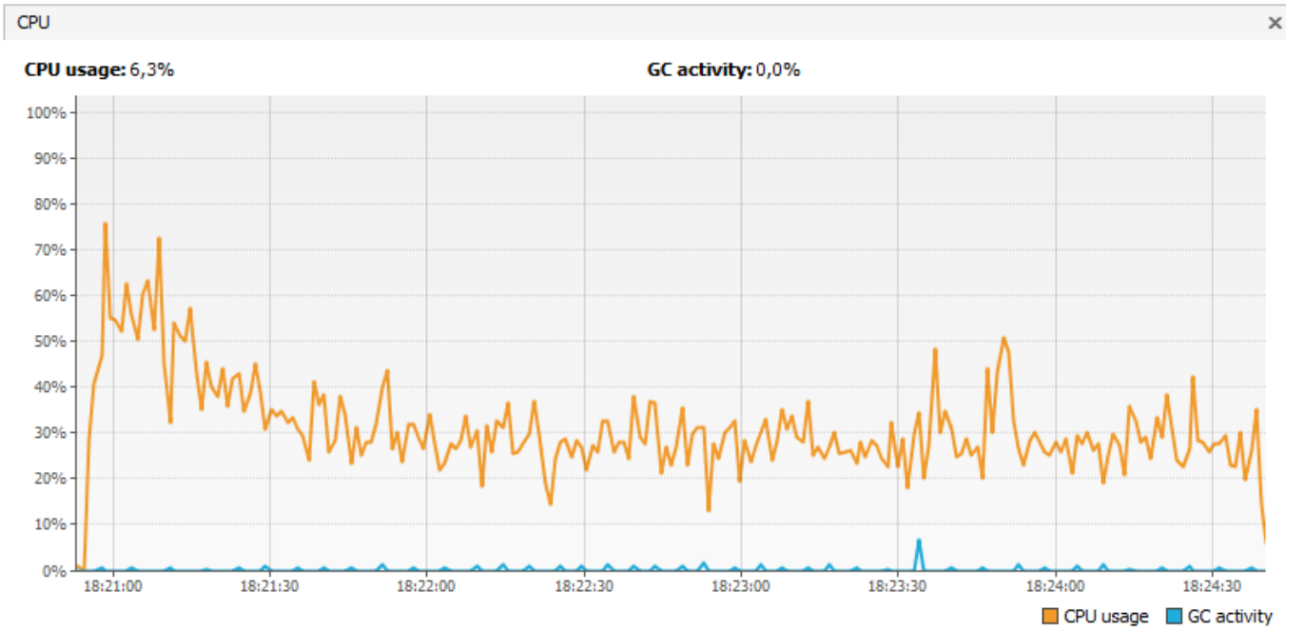

Рисунок 3 – Нагрузка ЦП при 1000 различных запросов

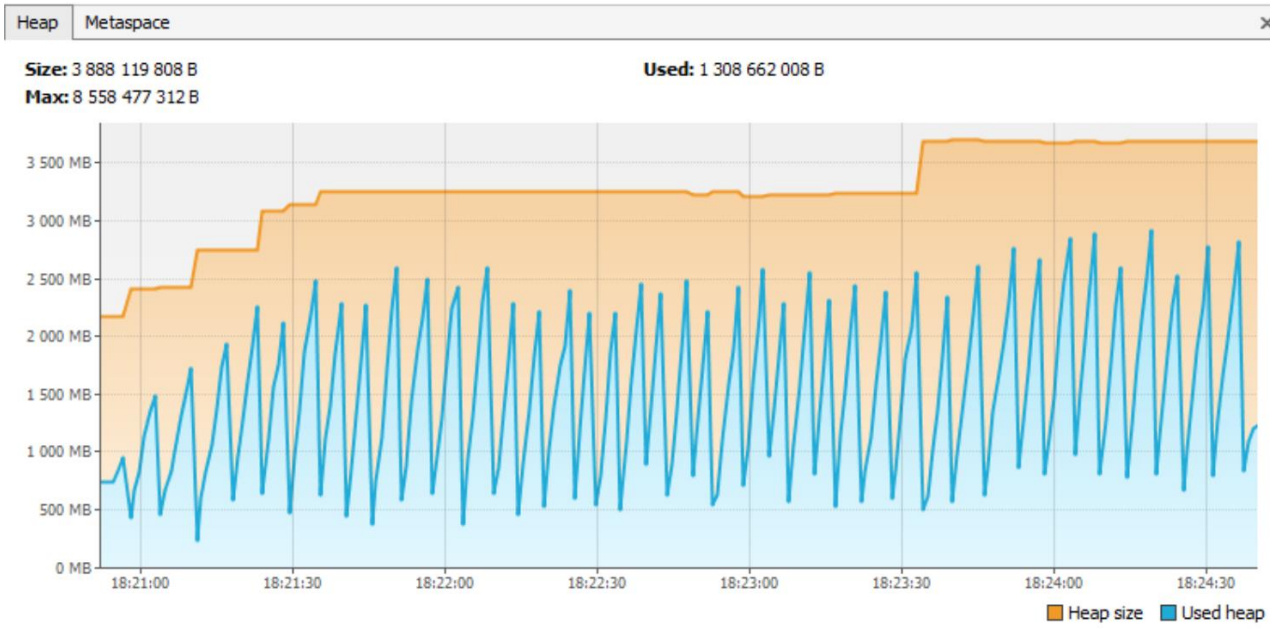

Рисунок 4 – Количеств используемой памяти при 1000 различных запросов

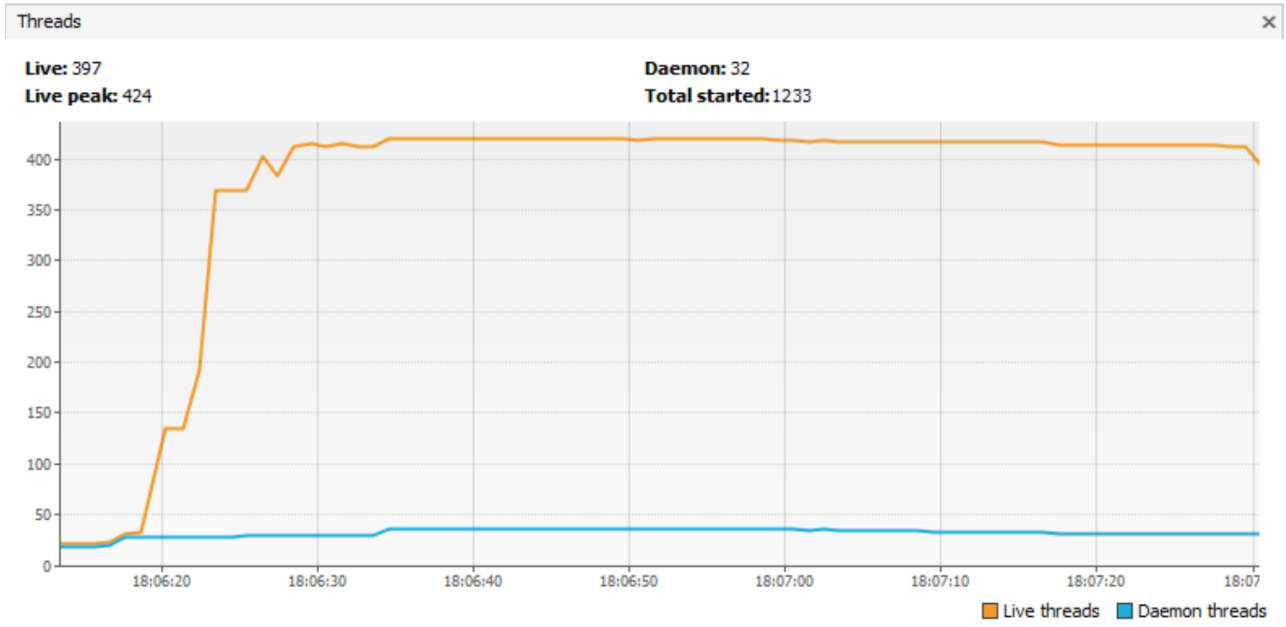

Рисунок 5 – Количество потоков при 1000 различных запросов

Исследование данной зависимости также было проведено для 1, 25, 100 и 10000 запросов. По полученным результатам была сформирована сводная таблица, результаты которой представлены в таблице 2.

Таблица 2 - Таблица зависимости показателей ресурсов от количества запросов.

Количество запросов	Средняя нагрузка ЦП	Количество используемой памяти	Количество потоков
1	6%	750 Мб	30
25	9%	800 Мб	48
100	19%	980 Мб	167
1000	67%	1,6 Гб	1256
10000	89%	2,4 Гб	15678

Данный анализ не выявил аномальных результатов в использовании ресурсов сервера под нагрузками. Начиная со 100 запросов в секунду пиковые показатели процессора примерно одинаковые, но есть различие в их продолжительности. Это обусловлено тем, что процессор тратит больше времени на обслуживание большего количества запросов. Java из коробки предоставляет несколько различных возможностей для организации работы Garbage Collector. Для анализа его работы рассмотрим частоту и длительность срабатывания сборки неиспользуемых объектов в памяти приложения. По графику на рисунке 4 видно, что при возрастании использования heap-памяти, Garbage Collector достаточно результативно освобождает память и не дает приложению упасть с ошибкой Out Of Memory Error, сигнализирующей о недостатке памяти. При увеличении одновременного количества запросов возрастает количество потоков, что объясняется тем, что для каждого запроса используется свой поток. Таким образом можно сделать вывод, что ресурсы сервера расходуются эффективно.

Список литературы

1 Henderson, C. Building Scalable Web Sites / C. Henderson – California : O'Reilly Media, 2008. – 352 с.

2 Newman, S. Building Microservices: Designing Fine-Grained Systems / S. Newman – Sebastopol : O'Reilly Media, 2015. – 280 с.

3 Matthias, K. Docker: Up & Running / J. Allspaw., S. Kane. – Sebastopol: O'Reilly Media, 2015. – 232 с.

4 Schlossnagle, T. Scalable Internet Architectures / T. Schlossnagle - California: O'Reilly Media, 2006. – 244 с.

ЭЛЕКТРОННЫЙ ДОКУМЕНТООБОРОТ ОРГАНИЗАЦИИ

Влацкая И.В., канд. техн. наук, доцент, Чернышев М.С.
Оренбургский государственный университет

С появлением бизнес-процессов возникает потребность в управлении механизмами с помощью упорядоченной документации. Рост предприятия увеличивает количество документации. В то же время, если не заниматься документами своевременно, то они начнут накапливаться и есть вероятность потерять важный документ или не исполнить вовремя поручение. Для того чтобы избежать подобных проблем применяют электронный документооборот.

Документооборот – это движение документов с момента их создания или получения до завершения исполнения, отправки адресату или передачи в архив. [1]

Электронный документооборот – это единый механизм движения документов, созданных с помощью компьютерных средств, как правило, подписанных электронной цифровой подписью, а также способ обработки этих документов с помощь различных электронных носителей. [2]

Задачи, решаемые системами электронного документооборота:

1 систематизацию и регламентацию работы с документами;
2 подготовка документов по шаблонам;
3 ведение номенклатуры дел организации;
4 автоматизацию учёта документов, в том числе:
   a. классификацию документов по различным критериям;
   b. регистрацию документов по заданным шаблонам и алгоритмам;
   c. учёт сроков хранения;
   d. помещение документов в дела и разбивка дел на тома;
5 автоматизацию поиска документов;
6 электронную рассылку документов;
7 автоматизацию процедур коллективной работы с документом:
   a. разработка проекта документа;
   b. согласование документа;
   c. экспертиза документа;
   d. исполнение документа;
8 обеспечение защиты от несанкционированного доступа и искажения или удаления информации.

Перед каждой организацией стоит свой набор задач и, соответственно, нет необходимости в излишнем функционале.

Системы электронного документооборота принято классифицировать по следующим типам в зависимости от особенностей:

1) Универсальные системы электронного документооборота:

- небольшой функционал, относительно остальных видов систем;

- неадаптированность под конкретику организации;
- доступность и простота в установке;
- недорогие по стоимости, относительно других видов систем электронного документооборота;
- техническая поддержка в период действия лицензии;
- чаще всего реализует общий документооборот без поддержки электронной подписи.

2) Индивидуальные системы электронного документооборота

- максимальная персонификация системы электронного документооборота;
- дополнительные траты на переобучение сотрудников и закупку оборудования;
- высокая стоимость;
- затраты времени на разработку, развертывание и внедрение больше, чем у остальных видов систем.

3) Комбинированные системы электронного документооборота

- полностью подходит для обеспечения потребностей организации;
- затраты на разработку, установку и введение в эксплуатацию снижаются;
- базовые модули позволяют быстро освоить систему и обучить персонал;
- может взаимодействовать с другими программными продуктами;
- заказчик получает полное право на программный продукт.

Основные требования, которые выдвигают организации к системам электронного документооборота являются: серверная и клиентская операционная системы, СУБД, возможность интеграции со сторонними продуктами, стоимость и наличие сертификата ФСЭК России. Так же выдвигаются требования к функционалу, который реализует система: делопроизводство, общий документооборот, управление договорной документацией, электронный архив, управление проектами, работа с документами СМК, электронная подпись.

Рассмотрим пример системы электронного документооборота реализующий обмен юридически значимой и общей документацией внутри предприятия. В данном случае необходимо наличие цифровой подписи. Согласно ФЗ №63 выделяют три типа подписи: простая, усиленная неквалифицированная, усиленная квалифицированная. В данном случае подписи имеют разную юридическую силу в зависимости от места применения и особенностей обработки. К примеру усиленная квалифицированная подпись в любой ситуации будет иметь юридическую силу, за исключением случаев в которых удастся доказать компрометацию подписи до момента ее использования. В случае с усиленной неквалифицированной подписью, она имеет юридическую силу в рамках органи-

зации, в случае документооборота с контрагентами необходимо заключать соглашение рассматривающее порядок формирования обработки подписи и организацию хранения подписанной документации. Существенным отличием квалифицированной и неквалифицированной подписей является то, что усиленную подпись необходимо приобретать в специальных удостоверяющих центрах, сертифицированных ФСБ России. Неквалифицированную же подпись можно сгенерировать силами IT отдела, или так же приобрести в определенных организациях, но в данном случае наличие сертификата ФСБ России у организации не обязательно.

На данный момент усиленную неквалифицированную подпись применяют во многих сферах деятельности. Последние года данный вид подписи активно применяется в сфере государственных заказов, на этапе подачи заявок на участие в тендере. Обратим внимание, что итоговые документы по заключению договоров по государственным заказам подписываются только с применением усиленной квалифицированной подписи. Нормальной практикой является приобретение небольшого количества ключей усиленной квалифицированной подписи для руководителей, их заместителей и бухгалтерии, а на всю остальную организацию - усиленную неквалифицированную. Так же данный вид подписи применяется для обращений граждан в государственные органы через портал государственные услуги.

Так же существует проблема организации хранения подписанной документации. Срок действия сертификатов подписей ограничен, а в некоторых случаях согласно ФЗ №125 «Об архивном деле», срок хранения документов может достигать 75 лет, но в большинстве случаев не превышает 10, то существует необходимость хранения временных меток подписи, а также основной информации по сертификатам подписей. Для решения данной проблемы в базе данных необходимо хранить дату подписи документа, а также добавлять данную метку к самой подписи, а для ключей использовать PKI, где так же будут храниться даты действия и отзыва сертификатов ключей. Другим путем решения проблемы является перенос документов для длительного хранения на бумажные носители и заверение их печатями и подписями ответственных должностных лиц.

Очень важным является выбор алгоритма электронной подписи. Разные алгоритмы имеют разную стойкость в зависимости от длины ключей, разную основу и разные хэш-алгоритмы. Более молодыми являются алгоритмы на основе эллиптических кривых. Они имеют большую вычислительную стойкость и, как следствие, меньшую рекомендуемую длину ключа. Не маловажным параметром является используемая хэш-функция, к примеру алгоритм SHA1 и SHA512 имеют разную крипто стойкость и вероятность коллизии. Так же на

примере SHA1 можно сказать о возможности практического взлома, но на данный момент это займет пять миллиардов лет. NIST прогнозирует возможность реализации практического взлома в ближайшие 5-10 лет, из-за этого NIST планирует полностью отказаться от данного алгоритма хэширования.

Таблица 1 - сравнение алгоритмов электронной подписи

Название	Дата создания	Основа	Рекомендуемый алгоритм хэширования	Рекомендуемая длина ключей
DSA	1991	Вычислительная сложность взятия логарифма в конечных полях	SHA	2048 бит
EGSA	1985	Вычисление дискретных логарифмов в конечном поле	SHA	1024 бит
RSA	1989	Факторизация больших чисел	SHA/MD5	2048 бит
ГОСТ Р 34.10-2012	2012	Операции в группе точек эллиптической кривой, определенной над конечным простым полем.	ГОСТ Р 34.11.2012	512 бит

Из таблицы можно сделать однозначный выбор в пользу алгоритма ГОСТ, Он новее всех остальных, имеет большую крипто стойкость при меньшей рекомендуемой длине ключей, что существенно упростит задачу создания пары ключей.

Стандарт ГОСТ Р 34.10-2012 разработан центром защиты информации и специальной связи ФСБ России с участием Открытого акционерного общества «Информационные технологии и коммуникационные системы». Утвержден 7 августа 2012 года и заменил собой стандарт ГОСТ Р 34.10-2001.

Данный стандарт содержит описание процессов формирования и проверки электронной подписи, реализуемой с использованием операций в группе точек эллиптической кривой, определенной над конечным простым полем.

Криптографическая стойкость данной схемы цифровой подписи основывается на сложности решения задачи дискретного логарифмирования в группе точек эллиптической кривой, а также на стойкости используемой хэш-функции. Алгоритмы вычисления хэш-функции установлены в ГОСТ Р 34.11-2012. [3]

Схема формирования и проверки подписи представлена на рисунке 1.

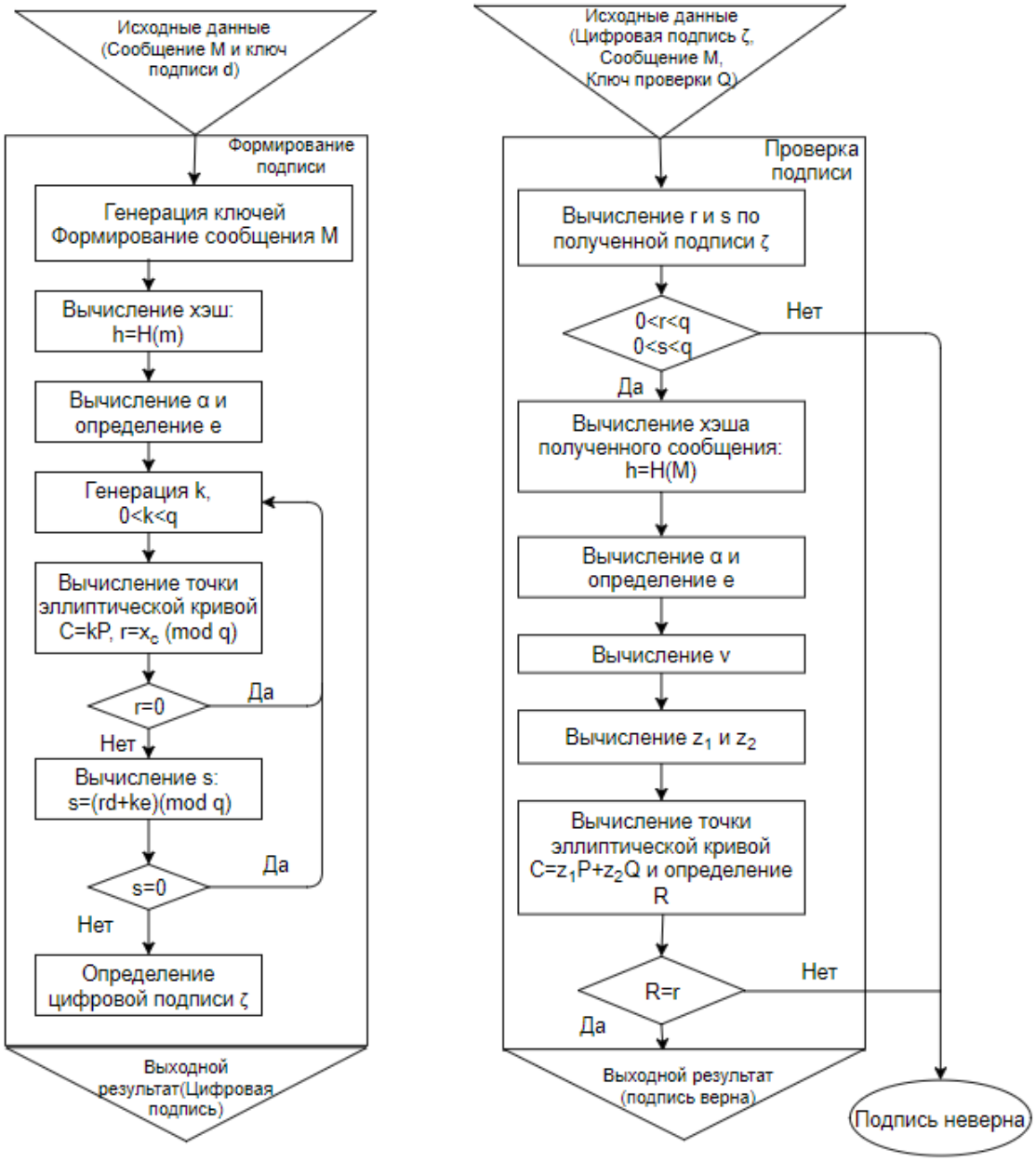

Рисунок 2 – Процесс формирования и проверки подписи (взято из ГОСТ Р 34.11-2012)

Для любой системы существует ряд угроз информационной безопасности. Данные угрозы представлены в банке данных угроз сформированным ФСТЭК России.

УБИ.030: Угроза использования информации идентификации/аутентификации, заданной по умолчанию. [4]

Данная угроза заключается в том, что у программы и ее компонентов существуют учетные записи по умолчанию, которые применяются для начально настройки системы. Например, у базы данных при ее создании могли использоваться стандартные поля admin/admin. Единственным вариантом защиты от

данной угрозы является изменение этих заданных по умолчанию данных или же использовать при начальной настройке надежные варианты, а не простые и удобные.

УБИ.067: Угроза неправомерного ознакомления с защищаемой информацией. [4]

Данная угроза заключается в возможности неправомерного случайного или преднамеренного ознакомления пользователя с информацией, которая для него не предназначена. В данном случае не предусматривается использование каких-то специальных средств. Угроза заключается в возможности просмотра с монитора пользователя или чтение отпечатанных документов. Стоит отметить что данная угроза применима ко всей документации, как юридически значимой, так и нет. Защита в данном случае является физическое разделение рабочего пространства работников, а также контроль за печатаемой документацией на сетевых принтерах.

УБИ.074: Угроза несанкционированного доступа к аутентификационной информации. [4]

Угроза заключается в возможности извлечения паролей из оперативной памяти или паролей, хранящихся в файлах в открытом виде. Защитой является шифрование участков операционной памяти в которых хранится пароль, с использованием криптопотоков, а также отказ от хранения паролей в открытом виде. Особое внимание стоит обратить на разграничение прав доступа, сетевую защиту и физическую защиту рабочих станций, ведь без доступа к ним реализация данной угрозы невозможно.

УБИ.084: Угроза несанкционированного доступа к системе хранения данных из виртуальной и (или) физической сети. [4]

Данная угроза реализуется с помощью специальных программных средств, способных эксплуатировать слабости технологий, использованных при построении системы хранения данных. Например, в случае использования ftp-сервера или его модификаций, необходимо применять надежные наборы логина и пароля, запретить анонимный вход, разрешить доступ к административной части только с определенных IP адресов.

УБИ.086: Угроза несанкционированного изменения аутентификационной информации. [4]

В случае взлома рабочей почты, злоумышленник может изменить аутентификационную информацию для системы. Оптимальным методом защиты от подобной угрозы является смена этой информации только по запросу и после подтверждения администратором личности, звонок по рабочему телефону или

личном присутствии. Обратим внимание, что в случае проникновения злоумышленника в саму организацию и доступу к рабочему телефону данный метод не может гарантировать стопроцентную защиту.

УБИ.116: Угроза перехвата данных, передаваемых по вычислительной сети. [4]

Так называемая атака человек посередине. Решается путем шифрования трафика. Более того реализация данной угрозы без физического внедрения в сеть практически не реально.

Рассмотрим принцип работы предполагаемой системы. В системе присутствуют три типа пользователей: администратор, руководитель, сотрудник. Администратор осуществляет управление пользователями, отделами и ключами. Руководитель может создать документ, отправить документ в другой отдел и осуществлять поиск, создавать юридически значимый документ, получить юридически значимый документ, отправить и извлечь документ (обычный и юридически значимый) из архива, осуществлять поиск документов, отправить документ в другой отдел или внутри отдела. Перед отправкой юридически значимого документа осуществляется его подпись. После получения юридически значимого документа можно проверить его достоверность, так же руководитель имеет права на управление отделом, который находится в его подчинении. Сотрудник может создать документ, отправить документ в архив, получить из архива, отправить документ в другой отдел или внутри отдела и осуществлять поиск.

Список литературы

1 *Романов, Д.А. Правда об электронном документообороте / Д.А. Романов, Т.Н. Ильина, А.Ю. Логинова – Москва: ДМК Пресс, 2008. – 224 с. – ISBN 5-94074-171-1.*

2 *Бухгалтерия, учет и отчетность: Электронный документооборот, электронная отчетность. [Электронный ресурс]. – Режим доступа: http://www.klerk.ru/rubricator/elektronnyj-dokumentooborot-elektronnaja-otchetnost/.*

3 *"ГОСТ Р 34.10-2012. Национальный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи" (утв. и введен в действие Приказом Росстандарта от 07.08.2012 N 215-ст).*

4 *Банк данных угроз безопасности информации: ФСТЭК России. [Электронный ресурс]. – Режим доступа: http://bdu.fstec.ru/threat (дата обращения 20.12.2017).*

ПРИМЕНЕНИЕ КЛАСТЕРНОГО АНАЛИЗА ДЛЯ РАЙОНИРОВАНИЯ ЗОН СЕЙСМИЧЕСКОЙ АКТИВНОСТИ

Влацкий В.В., отдел геоэкологии УрО РАН, г. Оренбург

Задача анализа сейсмической активности состоит в определении района, временного отрезка, магнитуды возможных сейсмических событий. Рассмотрение методов классификации, описанных прикладной статистикой, показало, что имея такую задачу, как задача классификации узлов сейсмической активности, наиболее подходящим оказывается метод кластерного анализа. Как отправная точка, было рассмотрено два вида признаков – геофизические признаки и статистические признаки. Статистические признаки основаны на данных с сейсмических станций. Геофизические признаки основаны на экспертных данных о геоморфологических узлах. Применение кластерного анализа для рассмотрения 10 случайных узлов и прилегающих районов. В классифицированных районах рассматривались события, произошедшие за два предыдущих месяца. В результате, в районе 80 % узлов, отнесенных к уровню слабой сейсмической активности, в следующие полгода событий не наблюдалось. В районе 75% узлов, отнесенных к уровню сильной сейсмической активности, произошло одно или более событие. Более полные результаты представлены в табл. 1 и 2.

Таблица 1.Результаты классификации узлов на ретроспективных данных

№ узла	Классифицированный уровень активности	Количество событий за последующие полгода
1	Высокая активность	2
2	Средняя активность	1
3	Низкая активность	0
4	Высокая активность	2
5	Низкая активность	0
6	Низкая активность	0
7	Высокая активность	0
8	Высокая активность	3
9	Низкая активность	0
10	Низкая активность	1

Таблица 2. Результаты проверки качества классификации по уровням

Уровень сейсмической активности	Количество классифицированных узлов	Количество узлов, в районе которых произошли события	Процент от общего числа узлов	Количество узлов, в районе которых не произошли события	Процент от общего числа узлов
Низкий	5	1	20%	4	80%
Средний	1	1	100%	0	0%
Высокий	4	3	75%	1	25%

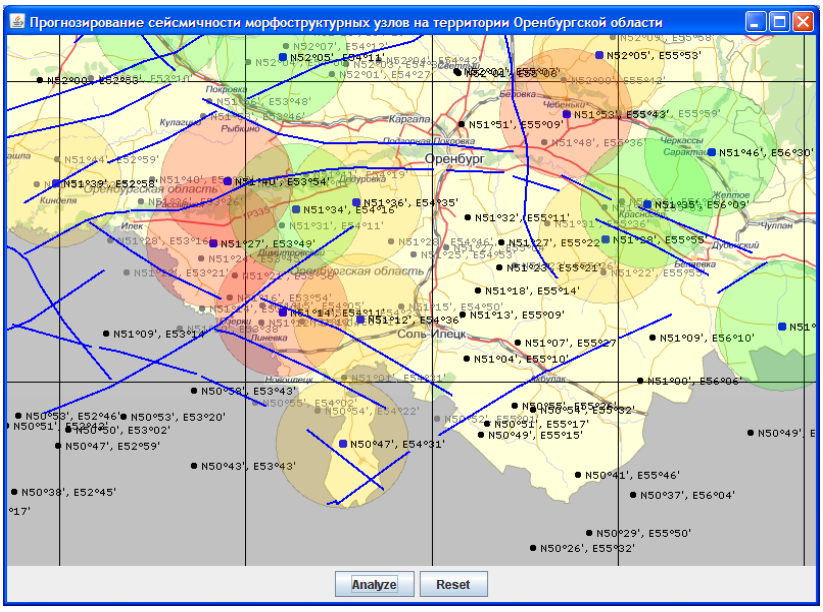

Рисунок 1- Результат классификации морфоструктурных узлов на месторождениях УВ в Южном Предуралье

На карте отображена координатная сетка, тектонические нарушения, морфоструктурные узлы, события и их координаты до проведения классификации. Узлы и зоны вокруг них выделены различным цветом, в зависимости от класса, в который они попали после классификации.

По данным пятилетних наблюдений за сейсмичностью на месторождениях УВ в Южном Предуралье, выявлено, что в пределах 5 км от разломов плотность событий составляет 0,00263 ед./км^{2}год. В полосе на расстоянии от 5 до 10 км от разлома она уменьшается на 23 % - до 0,00203. За пределами 10 км

от разлома количество событий уменьшается в 2 – 3 раза в сравнении с их количеством в зонах ближе 5 км и близко к средней плотности сейсмических событий во всей контролируемой сетью сейсмических станций территории нефтегазоносного Южного Предуралья, равной 0,0008 ед./км^{2}год. В зоне разломов, составляющей 1 % контролируемой сейсмическим мониторингом территории Южного Предуралья, происходит около 30 % всех событий.

За пределами разрабатываемых месторождений УВ на удалении от них более чем 15 км выделившаяся энергия сейсмических событий менее 10^4 Дж/км^2 ·год и заметного влияния на суммарную выделившуюся энергию в регионе не оказывает.

В расчете на тысячу квадратных километров за год в зоне планетарно-тектонической трещиноватости происходит около 9 событий с выделением сейсмической энергии на некоторых участках до 10^9 Дж/км^2·год, а в среднем в зоне разломов выделяется $7{,}03\cdot10^6$ Дж/км^2·год. На всей контролируемой сейсмической сетью территории в расчете на тысячу квадратных километров за год происходит 2-3 события с выделением сейсмической энергии до $1{,}14\cdot10^6$ Дж/км^2·год. Это в 7 раз меньше среднего ее выделения в зоне разломов и более чем в 1000 раз больше, чем среднее выделение энергии при сейсмических событиях внутри блоков на расстоянии более 15 км от разломов.

Для выявления и исследования закономерностей распределения сейсмической активности в районах разработки месторождений углеводородов построены полосовые и полигональные буферные зоны с различным шагом.

Таблица 3. Анализ сейсмической активности в районе месторождений углеводородов в зависимости от расстояния до разломов за 2008 – 2015 гг.

Расстояние до разлома, км	Площадь зоны, км^2	ство событий в среднем за год	% от общего числа событий	событий 10-3 ед./км2 в год	Сумм. выделившаяся энергия, Дж/год	Плотн. выделившейся энергии, Дж/км2·год
0 – 5	3532	16	27	4,4	$3{,}69\cdot10^{10}$	$1{,}04\cdot10^7$
5 – 10	3444	10,7	17	2,9	$2{,}40\cdot10^{10}$	$0{,}69\cdot10^7$
10 - 15	3494	5	7	1,2	$0{,}66\cdot10^{10}$	$0{,}20\cdot10^7$
15 – 20	4010	3,8	7	0,9	$<10^9$	$<10^6$
Южное Предуралье	661706	57,7	100	0,8	$7{,}56\cdot10^{10}$	$0{,}11\cdot10^6$

Используя эти данные о расстояниях от сейсмических событий до разломов, построена гистограмма распределения, которая представлена на рис. 8, математическое ожидание M=3,43 км и среднеквадратическое отклонение S=30 км.

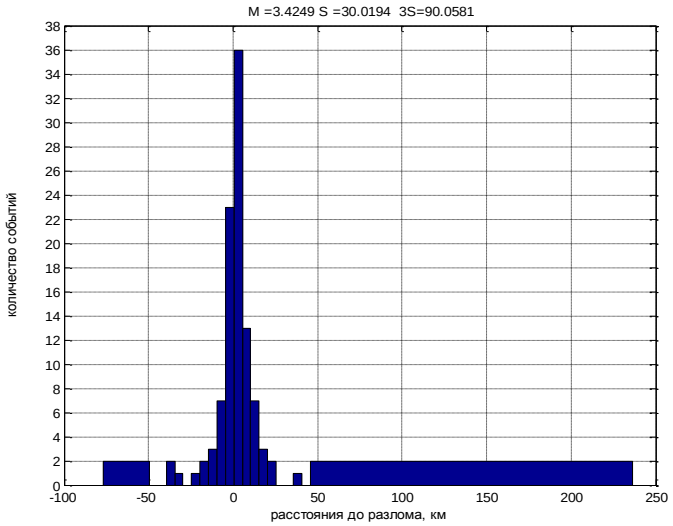

Рисунок 2- Гистограмма распределения всех сейсмических событий каталога в зависимости от расстояния до разломов.

Анализ распределения сейсмической активности недр показывает, что основная часть сейсмических событий располагается в районах интенсивно разрабатываемых месторождений углеводородов (табл. 4). Территории, удаленные от зон техногенных нарушений (центральная и восточная части Предуральского краевого прогиба, юго-восток Прикаспийской синеклизы и др.) имеют значительно меньшую частоту сейсмических событий и вероятно они вызваны естественными тектоническими процессами. События, произошедшие в зонах техногенных нарушений геологической среды (добыча нефти и газа и др.) происходят более часто и имеют более сложную природу, и их, по-видимому, следует относить к техногенным или природно-техногенным.

Таблица 4. Плотность зарегистрированных событий и выделившейся сейсмической энергии в районе месторождений УВ

Расстояние до месторождения км	Площадь, км^2	Событий в год	% от общего числа событий	Плотность событий, ед./км^{2}год	Сумм. выделившаяся энергия, Дж/год	Плотн. выделившейся энергии, Дж/км^2·год
В контуре месторождений	3582	9,7	17	0,0027	$1{,}01\cdot10^{10}$	$2{,}81\cdot10^{6}$
0-5	3129	4,7	8	0,0015	$0{,}93\cdot10^{10}$	$2{,}96\cdot10^{6}$

5-10	4360	6,7	12	0,0015	$0,56 \cdot 10^{10}$	$1,29 \cdot 10^{6}$
Южное Предуралье	661706	57,7	100	0,0008	$7,56 \cdot 10^{10}$	$0,11 \cdot 10^{6}$

Выявлено, что в контурах месторождений нефти и газа плотность событий составляет в среднем 0,0027 ед./км^2 в год. В полосовой зоне ограниченной расстояниями до 10 км от месторождения она уменьшается на 44 % до 0,0015. За пределами 10 км от месторождений количество событий уменьшается в 3-4 раза в сравнении с их количеством в контуре месторождений с плотностью 0,0008 ед./км^2 в год. Следовательно, на 1,6 % территории в контуре месторождений и 10 км вокруг них происходит более 35 % всех событий на контролируемой сейсмическим мониторингом общей территории Южного Предуралья.

Список литературы

1. *Нестеренко М.Ю., Никонорова О.А. Распознавание геодинамических неустойчивых зон районов эксплуатируемых нефтегазовых месторождений //Устойчивое развитие территорий: управление природными, техногенными, пожарными, биолого-социальными и экологическими рисками: материалы международной научно-практической конференции 5-7 октября 2011 года. – Оренбург: Издательский центр ОГАУ, 2011. – 236 с. – С.185 – 188.*
2. *Bradley, P. Scaling clustering algorithms to large databases [Текст] / P. Bradley, U.M. Fayyad, C.A. Reina. – Proc. 4th Int. Conf. Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, Calif., 1998. - pp.9-15.*
3. *Котов, А. Кластеризация данных [Текст] / А. Котов, Н.С. Красильников. – СПб.: БХВ-Петербург, 1998. –36 с.*

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ И ПРОТОКОЛЫ, ПОСТРОЕННЫЕ НА ПЛАТФОРМЕ НЕКОММУТАТИВНЫХ ГРУПП

Кайманова Е.А.
Оренбургский государственный университет

В настоящее время наиболее распространены криптосистемы и протоколы с открытым ключом. Данные алгоритмы и протоколы являются ассиметричными. Криптографические системы с открытым ключом в настоящее время широко применяются в различных сетевых протоколах, в частности, в протоколах TLS и его предшественнике SSL (лежащих в основе HTTPS), в SSH. Также используется в PGP, S/MIME.

Начало асимметричным шифрам было положено в работе «Новые направления в современной криптографии» Уитфилда Диффи и Мартина Хеллмана, опубликованной в 1976 году. Метод стал известен как обмен ключами Диффи - Хеллмана, был первым опубликованным практичным методом для установления разделения секретного ключа между заверенными пользователями канала. В 1977 году учёными Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом из Массачусетского технологического института был разработан алгоритм шифрования, основанный на проблеме о разложении на множители. Система была названа по первым буквам их фамилий (RSA — Rivest, Shamir, Adleman). RSA стал первым алгоритмом, пригодным и для шифрования, и для цифровой подписи [1] . Так же примерами ассиметричных алгоритмов и протоколов являются алгоритмы: DSA, EDSA, ГОСТ Р 34.10-2012, McEliece, схема Эль-Гамаля. Они основаны на теории чисел, следовательно, зависят от структуры абелевых групп. Развитие вычислительной техники сделало эти методы восприимчивыми к атакам, так же ожидание разработки квантового компьютера, для которого решение «трудных» задач станет возможным за полиномиальное время, привело к исследованиям некоммутативных групп, как основы для построения криптографических примитивов. Эта линия исследования получила название некоммутативная алгебраическая криптография. Основы некоммутативной криптографии изложены в монографии А. Мясникова, В. Шпильрайна и А. Ушакова.

Новые исследование так же связаны с использованием новых трудных задач, сложность которых была бы сверхполиномиальна и в случае применения квантового вычислителя.

Некоммутативная криптография предполагает исследование общих алгебраических приемов для построения криптосистем, изучение потенциальных алгебраических платформ (групп, колец) для реализации методов и схем, криптоанализ и анализ безопасности систем.

Для обоснования возможности использования группы рассматриваются следующие аспекты: группа должна быть хорошо изучена; должна существовать эффективно вычислимая нормальная форма для элементов группы, т.е.

проблема слов в группе должна быстро (за линейное или квадратичное время) решаться детерминированным алгоритмом; должен существовать способ сокрытия элементов группы такой, что было бы невозможно восстановить; группа должна быть группой суперполиномиального (т. е. экспоненциального) роста [2].

Среди первых попыток использования некоммутативных групп в криптографии были схемы Аншеля-Аншела-Голдфелда и Ко-Ли и др. Авторы примерно в одно и то же время предложили использовать некоммутативные группы как основы для криптосистем с открытыми ключами. Платформа для данных схем является группа кос Артина. Группа кос достаточно хорошо изучена, в них можно эффективно выполнять вычисления разного толка. В то же время существуют трудноразрешимые проблемы, дающие возможность построения стойких криптосистем.

Позже несколько других некоммутативных структур как группы Томпсона, полициклические группы, группы Григорчука и матричные группы были идентифицированы как потенциальные кандидаты на шифровальные заявления. Многие из них описаны в монографии А. Мясникова, В. Шпильрайна и А. Ушакова [5].

В основу криптосистемы кладется одна из сложных математических проблем. К основным «трудным » задачам, на основании которых построены криптографические протоколы и алгоритмы с открытым ключом на некоммутативных группах относятся: проблема равенства, проблема сопряжения, проблема факторизации и декомпозиции, проблема вхождения, проблема изоморфизма.

Для обеспечения достаточной стойкости алгоритмов и протоколов криптографии с открытым ключом требуется положить в основу вычислительно трудные задачи, для которых сложность решения имела бы сверхполиномиальную сложность как при решении на компьютерах обычного типа, так и при решении на квантовых компьютерах. В качестве базовой задачи была предложена задача нахождения сопрягающего элемента в некоммутативных группах кос и проблема одновременного поиска множества сопряжений.

Группы кос крайне эффективны при обеспечении трудоёмких вычислительных процессов. Благодаря этому, различными группами исследователей были предложены протоколы с преобразованием на данных группах. Криптография на группе кос позволяет реализовать два протокола обмена ключами: протокол Аншеля-Аншеля-Гольдфельда; протокол обмена ключами К. Н. Ко, аналогичный алгоритму Диффи-Хеллмана.

В протоколе Аншеля-Аншеля-Гольдфельда в качестве открытого ключа принимается два набора кос $\{p_1,\ldots,p_l\},\{q_1,\ldots,q_m\}$ где $p_i,q_j \in B_n$ для $1\le i\le l$ и $1\le j\le m$. Секретный ключ u, принадлежащий участнику обмена А, состоит

из *l* нитей. Аналогично секретный ключ *v*, принадлежащий участнику обмена B, состоит из m нитей. Обмен происходит следующим образом:

1. A генерирует косу $s = u(p_1, \ldots, p_l)$, и использует ее, чтобы сгенерировать сопряженные $q_1' = sq_1s^{-1}, \ldots q_m' = sq_ms^{-1}$; пересылает $q_1', \ldots q_m'$
2. B генерирует косу $r = v(q_1, \ldots, q_m)$, и использует ее, чтобы сгенерировать сопряженные $p_1' = rp_1r^{-1}, \ldots, p_l' = rp_lr^{-1}$; пересылает $p_1', \ldots p_l'$
3. A вычисляет $t_A = su(p_1', \ldots, p_l')^{-1}$.
4. B вычисляет $t_B = rv(q_1', \ldots, q_m')r^{-1}$.

Искомый ключ $t_A = t_B$

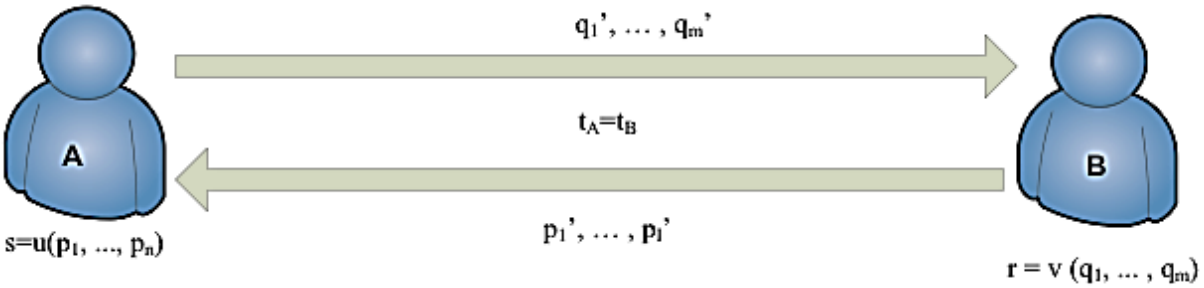

Рис. 1. Схема протокола Аншеля-Аншеля-Гольдфельда

Такая последовательность действий требует $2m + 2l + 2$ операций умножения и *m+l* операций нахождения обратных кос. Протокол основывается на проблеме одновременного поиска множества сопряжений.

Протокол, который предложен К.Н. Ко, базируется на протоколе Диффи-Хеллмана. Здесь, открытый ключ *p* это определённая коса в группе B_n. Секретный ключ, принадлежащий A, представляет собой косу *s* из подгруппы LB_n, а секретный ключ B — косу r из подгруппы RB_n. Обмен ключами происходит следующим образом:

1. A и B договариваются о выборе открытого ключа $p \in B_n$;
2. A генерирует сопряжение $p' = sps^{-1}$ пересылает его B;
3. B генерирует сопряжение $p'' = rpr^{-1}$ пересылает его A;
4. A вычисляет $t_A = sp''s^{-1}$;
5. B вычисляет $t_B = rp'r^{-1}$;

Искомый ключ $t_A = t_B$

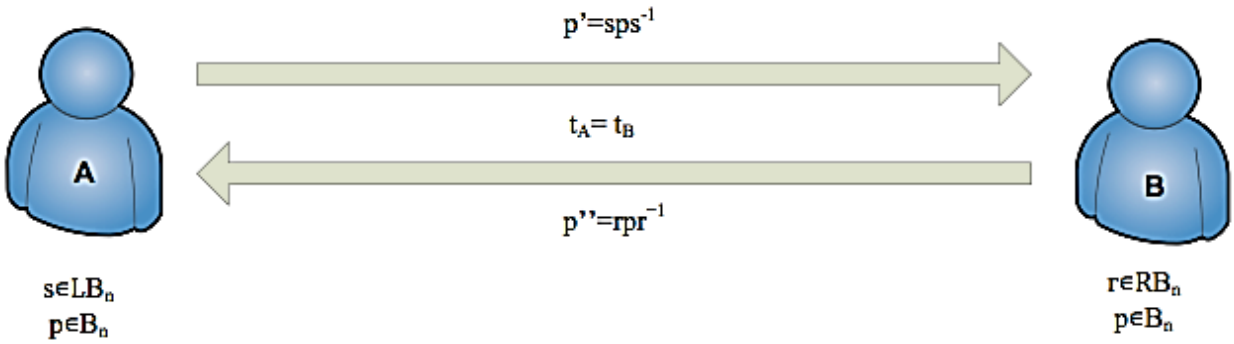

Рис.2. Схема протокола, предложенного К.Н.Ко

Такой протокол требует 8 операций умножения кос и 2 операции нахождения обратной косы. Протокол основывается на проблеме сопряжения кос.

К основным характеристикам криптографических систем, базирующихся на группе кос относятся:

Входящее сообщение, бит	$p \cdot l \cdot \log(n)$
Зашифрованное сообщение, бит	$4p \cdot n \cdot \log(n)$
Скорость зашифрования, операции	$O\left(p^{2n} \log(n)\right)$
Скорость расшифрования, операции	$O\left(p^{2n} \log(n)\right)$
Длина персонального ключа, бит	$\frac{1}{2} p \cdot n \cdot \log(n)$
Длина открытого ключа, бит	$3p \cdot n \cdot \log(n)$
Сложность атаки «грубая сила»	$\left(\left(\frac{n}{2}\right)!\right)^p = \exp\left(\frac{1}{2} p \cdot n \cdot \log(n)\right)$

где p - каноническая длина, n - индекс косы.

Таким образом, группы кос крайне эффективны при обеспечении трудоёмких вычислительных процессов. Представленные протоколы позволяют решить такие задачи информационной безопасности, как обеспечение конфиденциальности хранимой информации, обеспечение конфиденциальности передаваемой информации, обеспечение целостности хранимой информации, обеспечение целостности передаваемой информации, обеспечение подлинности информации; аутентификация пользователей, обеспечение анонимности пользователей и другие.

Рассмотренные криптографические системы показывают, что разработка алгоритмов, использующих группы кос является перспективным направлением в развитии современной криптографии.

Список литературы

1. *Глухов М. М. К анализу некоторых систем открытого распределения ключей, основанных на неабелевых группах. Матем. вопр. криптогр., 2010.*
2. *Молдовян Д.Н. Примитивы криптосистем с открытым ключом: конечные некоммутативные группы четырехмерных векторов. Журнал Информационно-управляющие системы, Выпуск№ 5 / 2010*
3. *Паршина Д. А., Митяева И. А., Горбенко И. Д. Анализ криптографических систем в группах КОС / Прикладная радиоэлектроника: наук. техн. журнал, 2012. – Том 11. № 2. — С. 210–215*
4. *Anshel I.,.Anshel M,.Goldfeld D, An Algebraic Method for Public Key Cryptography, Math.Res.Lett, 6, 1999, 287-291 Springer Verlag*
5. *Myasnikov A.G., Shpilrain V. and Ushakov A., Group-Based Cryptography Advanced Courses in Mathematics, CRM Barcelona, 2007*
6. *Cha J.C., Ko K.H., Lee S.J., Han J.W., Cheon J.H., An efficient implementation of braid groups, AsiaCrypt 2001, Springer Lect. Notes in Comput. Sci., 144–156.*

МАТЕМАТИЧЕСКИЕ ЗАДАЧИ С ЭКОНОМИЧЕСКИМ СОДЕРЖАНИЕМ В СОВРЕМЕННОМ ОБРАЗОВАНИИ

Колобов А.Н., канд. техн. наук, доцент
Оренбургский государственный университет

Сегодня школьному образованию уделяется огромный интерес. Предоставление учащимся крепких знания является одной из основных задач данного образования.

Сильная ценность содержится в способности понимать и решать задачи с экономическим содержанием, так как их можно отнести к жизненным ситуациям.

Многие учащихся не могут самостоятельно справиться со сложностью таких задач, так как в младших классах на эту тему отводится незначительно малое количество часов.

С помощью исследований обнаруживается, что большое количество учащихся встречают трудности. Они не могут ориентироваться в задачах с экономическим содержанием, то есть это банковские взносы и кредиты.

Следовательно, предпочтительно смотреть на данную тему ежедневно, принимая во внимание, что такие задачи есть в повседневной жизни. Помимо этого, не стоит забывать, что, окончив школу, выпускники стремятся поступить в различные учебные заведения. А для этого необходимы знания. Так же сдача ЕГЭ не обходится без задач с экономическим содержанием.

Ценной работой учителя математики является подготовка выпускников к экзамену. В этот период приходится вспоминать даже элементарные формулы, потому что не все учащиеся их помнят. Задачи с экономическим содержанием — это в основном задачи на проценты. Впервые школьники познакомились с ними в 5 классе, что говорит о возможности про них забыть.

Следовательно, данные задачи имеют место быть. Представления кредита и вклада возникли еще в далеком прошлом, когда возникло определение долга. Выплаты были необходимы, как и займы. Таким образом, в математике зарождались проценты. Вскоре они появились и в других науках, как, например, химия и физика.

Для того, чтобы ввести определение понятия «задачи с экономическим содержанием», следует сначала разобрать эти задачи и выяснить из чего они состоят. Данные задачи в основном на вклады или кредиты, курсы валют, а также торгово-денежные отношения и на простые и сложные проценты. В большинстве случаев такие задачи решаются с помощью введения понятия «процент».

Термин «процент» имеет латинские корни, что практически обозначает «100». История данного понятия берет свое начало с Европы. Именно в этой части света в 15 веке была введена десятичная система счисления.

В литературе есть предположение, что в 1584 г. именно ученый из Бельгии ввел впервые данный термин, его звали Симон Стевин и именно он издал таблицы процентов. Только в 18 веке в РФ стали использовать понятие «процент». Длительный период времени этот термин представлял собой доход или ущерб, и он применялся в трейдерских и валютных сделках, то есть в торговых и денежных. Спустя некоторое время область применения процентов увеличилась, и они стали попадаться в статистике, хозяйстве, экономике, а также науке.

Есть версия происхождение данного знака. Существует теория, будто бы обозначение % имеет происхождение от слова procento, что с итальянского языка переводится как 100. Сначала было pro cento, позже сократили до cento, потом до cto и вскоре стали писать c/o, буква t превратилась в наклонную черту. Посредством дальнейшего сокращения появился знак %.

Существует еще одна теория происхождения данного знака. Если открыть учебник математики пятого класса, авторы которого являются Н.Я. Виленкин, Ф.С. Чесноков и В.И.Жохов [1], то в разделе «история математики» данное обозначение появилось вследствие опечатки в 1658 г. в книге Матьёде ла Порта «Руководство по коммерческой арифметике». Согласно оплошности наборщик напечатал знак %, вместо слово cto.

Что означает кредит и какое его происхождение?

Кредит это общественные дела, образующиеся меж субъектами финансовых отношений по предлогу изменения стоимости, это передача денежных средств в долг на определенное время и обязательный их возврат и своевременная платежеспособность.

Впервые в Российской империи термин « кредит» появился в 1703 году. В то время «кредит» означал «авторитет» и имел немецкое происхождение. Если сравнить термин «кредит» с латинским credo, то получим перевод как верую, занимаю.

Вклады. Вклад это сумма средств, которая предоставляется банку от клиента на конкретный или неконкретный срок. Чем больше срок, тем больше проценты начисляются клиенту от банка.

История вклада идет еще с 7 столетия до н.э. В древней Греции принимали средства на временное хранение, а во 2 веке до н.э. в городах Фивы, Гермонтис имелись накопительные банки, в которые собирали сборы налогов и тратили их на нужды общества.

Валюта. Курс валюты.

Термин «валюта» имеет итальянское происхождение и обозначает товар, способный исполнять функцию средств при обмене.

В узком значение валюта это денежная единица, главный элемент валютной системы государства и вселенской валютной системы.

Валютный курс – стоимость валютной единицы одного государства, переведенная в валютную единицу иного государства.

Торгово-денежные отношения появились во время кредитов и означают публичные дела, образующиеся меж людьми в ходе изготовления и продажи.

Простые проценты это начисления в конце времени 1 раз.

Сложные проценты это начисления, добавляющиеся к основной сумме долга.

Производительность труда это коэффициент деятельности работников, измеряющаяся количеством работы, потраченной за единицу времени.

Рассматриваемые задачи всегда занимали особенное место в математики. Их применение идет со времен Древнего Вавилона, в виде глиняных табличек и иных письменных источников. Длительный период запас математических сведений переходился из поколения в поколение, как задачи с имеющимся решением, встречающиеся в жизни.

Причина повышенного интереса к таким задачам состоит в том, что длительный период целью обучения детей математике было усвоение знаний, связанных с жизненными ситуациями и практическими расчетами.

Существует вторая причина повышенного интереса. Она заключается в том, что не только заимствовали древний метод обучения, но и сформировали значимые общеучебные знания, полученные с помощью разборов сюжетных задач, выделения проблемы и главного вопроса, а также проверкой полученного результата и анализа текста. Также важно приучить школьников переводить экономические задачи в вид математических действий, графиков или же уравнений с неизвестными. Задачи помогают развивать не только логические способности ученика, но и образное мышление. Стоит сказать, что они, несомненно, повышают эффективность изучения математики и других дисциплин.

Арнольд И.В. в 40-ые годы описывает изучение решению сюжетных задач как к инертному, то есть пассивному запоминанию учащимися небольшого количества типовых примеров, по которым нужно запоминать ход решений в разных случаях. Он считает, что такие задачи ничтожно полезны по сравнению с затрачиваемым напряжением мыслей. В общем счете это абсолютная беспомощность и неумение разбираться в наиболее простых задачах.

К 1950 году экономические задачи были отлично систематизированы, методы их применения в школьном курсе лучше разработаны, но реформы образования к концу 1960 году изменились.

К середине 20-го столетия в СССР немного побеждали практические подходы в применении математико-экономических задач. Для того чтобы обучить школьников, необходимо применять все изученные способы на практике, по мере возможностей. Традиционные способы решения задач стали считаться устаревшими и поэтому решили перейти к применению уравнений. Данный способ оказался наиболее научным и современным.

Именно благодаря этому данные задачи представляли немаловажную значимость в ходе обучения России, и им давалось достаточно много времени при изучении математики в школе.

Задачи с экономическим содержанием – это задачи, подразумевающие наличие терминов в области экономики, и требующие для их решения математическую модель.

Задача формулируется из таких стадий, как:

- Проблемная ситуация;
- Постановка задачи;
- Достаточность условия;
- Вид условия (словесно, действительные обстоятельства, изображение);
- Заключение.

Решить задачу означает ответить на поставленный вопрос с помощью мыслительных операций и выполнения определенного алгоритма.

Существуют два поиска решения: прямой и метод от противного. При первом проверяют доступные способы решения и смотрят, является ли это результативным применением. Метод от противного заключается в отрицании предполагаемого утверждения. Можно сказать, что это классическая логика, то есть логика, указывающая верны ли предполагаемые утверждения [2].

Сложность задачи состоит не только в постановки и условий, также ее понятия. Не все дети хорошо подготовлены к рассуждениям, поэтому одна из трудностей, это их мыслительный процесс.

Рассмотрим некоторые актуальные задачи, встречающиеся на ЕГЭ и имеющие непосредственное отношение к экономике:

- Вклады;
- Кредиты;
- Проценты;

Возникает вопрос, почему проценты? Скорее всего, потому, что все чаще во второй части ЕГЭ встречаются задачи на проценты. Обычно это задачи на сложные проценты и за правильный ответ дается 3 первичных балла, что является высокой оценкой.

Для того, чтобы решать каждую из отмеченных задач, обязательно нужно знать две главные формулы. Они доступны любому ученику, но, стоит отметить, что во многих случаях эти формулы игнорируются.

Приведем жизненную ситуацию. Человек, с зарплатой 300 тысяч, захотел купить квартиру, стоимостью 20 миллионов рублей. За год он имеет возможность откладывать на нее 3 миллиона рублей. Сколько лет потребуется для накопления нужной суммой?

Решение: $20/3 \approx 6, (6) \approx 7$ лет.

Так как его зарплата большая, то лучше отнести деньги в банк. Это не только надежно, но и выгодно. Чем больше вклады, тем больше процент начисления от банка. При 15% годовых сумма вклада увеличивается в 1,15 раза.

Вследствие чего получается, что вклады, не смотря на низкий процент банка, дают хороший результат, значительно превышающий годовой доход. Стоит заметить, что основной итоговый доход доводится на заключительные года.

Нужное в данных рассуждениях - это формула, позволяющая отыскать окончательную сумму взноса при ежегодных платежах и начисляемые проценты банка. Запишем следующее:

$$\text{Вклад} = \text{Платеж} \bullet \frac{\%^n - 1}{\% - 1},$$

где $\frac{\%^n - 1}{\% - 1}$ - сумма геометрической прогрессии,

n – количество лет.

Платеж - это сумма, которая откладывается, в зависимости от задачи

Вклад – общая сумма денег, которая окажется в конце накоплений.

Причем $k\%$ это $1 + \frac{k}{100}$.

Данную формулу следует понимать как главную формулу суммы взноса. Она способна уменьшить подсчеты в данных задачах [3].

У многих может возникнуть вопрос, к чему эти сложности, нельзя ли просто решить с помощью таблицы, как прописано во многих учебниках, посчитать отдельный год, а затем посчитать общую сумму вклада. Можно совсем забыть о сумме геометрической прогрессии и считать с помощью таблиц, так сделано в большинстве сборников по подготовке к ЕГЭ. Однако при этом, во-первых, резко увеличивается объем вычислений, а во-вторых, как следствие, увеличивается вероятность допустить ошибку.

Если есть какой-то способ упростить и сократить вычисления, то именно этим способом и надо воспользоваться.

Поговорим о кредитах. Не все хотят копить, и поэтому многие берут кредиты. Причем некоторые люди, взяв в кредит машину, квартиру и телефон, смеются над теми, у кого этого нет. Можно подсчитать, сколько люди потеряют своих денег, отдав проценты за кредит.

Сформулируем задачу. Одноклассник взял кредит в 2 миллиона рублей по ставке 20%. Срок погашения 3 месяца. Постараемся связать все в одну формулу.

$2m$ – исходная задолженность,

k – коэффициент суммы на начисление,

x – оговоренная ежемесячная сумма.

Решаем линейное выражение, в ходе которого x =949, 208 тысяч рублей.

Получим вторую важную формулу на поиск процентов, кредитов и платежей.

$$\text{Кредит} \bullet \%^n = \text{Платеж} \bullet \frac{\%^n - 1}{\% - 1},$$

Эта формула позволяет решить около 80% задач с экономическим содержанием из второй части ЕГЭ по математике.

Экономические задачи это задачи, связанные с финансовой деятельностью, и, именно поэтому они есть в школьном курсе.

Ученик должен иметь представление не только о финансах, но и о грамотной их растрате. Это нужно, чтобы в последующем правильно управлять личными финансами, понимать, что такое банковское дело и инвестиции в финансовом рынке, разбираться в вопросах страхования и различать мошенничество и финансовые пирамиды, которые на Российском рынке эксперты оценивают порядком 120, действующих в Российской Федерации.

Развитие школьного финансового образования так же необходима, как трактовка проблем его содержания [4].

Если ученик сможет говорить с преподавателем «на одном языке», то есть понимать его мысли и правильно отвечать на вопросы, то и наиболее эффективнее закрепятся его знания. При взаимопонимании учитель старается разбирать большинство тех задач, которые остались непонятными или неудачно раскрытыми.

Следовательно, предпочтительно смотреть на данную тему ежедневно, принимая во внимание, что такие задачи есть в повседневной жизни.

Список литературы

1. Виленкин, Н.Я. Математика. 5 класс: учеб.для учащихся общеобразоват. учреждений / Н.Я. Виленкин, В.И. Жохов, С.И. Шварцбурд, А.С. Чесноков – 31-е изд., стер. – М.: Мнемозина, 2013. – 280 с.: ил.

2. Колобов, А.Н. Об изучении векторной геометрии в современной школе. / А.Н. Колобов, И.В. Прояева // Мир науки, культуры, образования. Международный научный журнал. - 2017 г.- № 4.- с.199-203.

3. Колобов, А.Н. Применение интерактивных технологий в процессе подготовки к олимпиаде по математике. / А.Н. Колобов, И.В. Прояева // Мир науки, культуры, образования. Международный научный журнал. - 2017 г.- № 6.- с.169-175.

3. Колобов А.Н. Информационные технологии в обучении на современном этапе. / А.Н. Колобов // Университетский комплекс как региональный центр образования, науки и культуры [Электронный ресурс] : материалы Всерос. науч.-метод. конф., Оренбург 1-4 февр. 2017 г. / Оренбург. гос. ун-т. – Электрон. дан. – Оренбург : ОГУ, 2017. – с. 3110-3113. 1 электрон. опт. диск (CD-ROM). – ISBN 978-5-7410-1639-8.

ОЦЕНКА ЗНАЧИМОСТИ МАТЕМАТИКО-ЭКОНОМИЧЕСКИХ ЗАДАЧ В ОБРАЗОВАНИИ

Колобов А.Н., канд. техн. наук, доцент
Оренбургский государственный университет

Существуют два значения математической науки: фактическое, связанное с формированием и использованием инвентаря, требуемое человеку в его работе, и внутреннее, то есть духовное. Духовное назначение согласовано со способом постижения и мышления. Математика необходима для понятия основ устройства, применения нынешней технологии [1].

В ходе изучения в запас приемов и навыков непосредственным образом вводится индукция и дедукция, исследование и обобщение, формируются способности излагать, приводить аргументы, а так же развивать логику.

Второе значение это творческая сторона мышления.

При усвоении школьного курса, одни учащиеся пользуются только своим уровнем подготовки, другие же добиваются наибольших итогов, с помощью способности логически и творчески рассуждать, включать интуицию и смекалку. Длительный период формированием данных способностей интеллекта учебное заведение пренебрегала, или объединяла их ключевым способом к получению обучающимися простых умений и навыков.

Из-за перехода к рыночным отношениям и независимой финансовой работы людей, роль интеллекта увеличилась, таким образом, сейчас необходимо водить рассудительный и обдуманный стиль жизни.

В структуру интеллекта входят: способность быстро и рассудительно решать появляющиеся проблемы, осторожность, практичность, находчивость, инициативность и экономичность,

Находчивость выражается тогда, когда человек способен найти несколько решений в трудной жизненной ситуации. Суть в том, что какой бы вопрос не стоял, всегда готов найти подходящее решение. Из каждой проблемы находчивый человек найдет выход [2].

Задачи с экономическим содержанием хорошо развивают логическое мышление учеников, их возможности и умение найти решение. Такие задачи способствуют развитию экономичности и рассудительности.

Экономичность, как особенность состоит в том, что владеющий ей человек, в состоянии выйти из определенных условий с минимальными расходами и издержками.

Рассудительность- способность посмотреть в будущее, и, предугадав результаты своих действий, конкретно определять их итог.

Способность быстро справляться со своими задачами – это динамическая оценка интеллекта, проявляющаяся в затраченном на это время.

Задачи с экономическим содержанием нельзя причислить к легким задачам, так как они не легкие в усвоении. Такие задачи учащиеся рассматривают

еще в 5-6 классе, но будучи уже старшеклассниками, они сдают ЕГЭ по математике, одно задание в котором является как раз данная задача.

Конечно, учитываются возрастные рамки, и сложность таких задач со временем возрастает.

Вопросы, относящиеся к экономическим задачам, дают возможность продемонстрировать обучающимся, что знания, полученные в ходе решения, применяются в обыденности.

При обучении этой темы, подростки знакомятся с различными методами решения задачи, к тому же диапазон образцов шире. Обучающийся овладевает разными методами мышления, обогащая собственные приемы и методы. Также он имеет возможность пользоваться тем способом, который ему кажется наиболее подходящим [3].

В России хоть и нет культуры финансовой грамотности, все равно о ней должен знать каждый. Настал момент, когда мы вышли в большую экономику и поэтому некоторые люди не очень хорошо понимают, что делать. Тем более, когда кредиты, вклады и реклама доступны на каждом шагу. Многие хотят приобрести недвижимость, технику или что-то еще сразу, поэтому берут кредиты. Иногда под довольно таки большой процент.

Поговорим о некоторых ошибках учителей, и так:

Погрешность 1. Пропуск шага оценивания данных в заданиях. «Прочитайте условие задания и решите, кто будет показывать решение у доски?» – это многократно наблюдается практически на всех уроках. И незамедлительно наступает выполнение текстовой задачи с экономическим содержанием. Шаг оценивания не хватает во многих учебных пособиях. Причем стоит заметить, что учитель не редко проводит данный шаг. «Учащиеся буквально недавно решали очень схожую задачу. Для чего проводить оценивание, если задание практически подобное?». Данное высказывание возможно оспорить. Возможно, осуществление шага необходимо не всем. В любом классе есть учащиеся с более свернутым и понятливым только для них шагом. Такие ученики мгновенно его выполняют, следовательно, быстрее находят решение и его правильно оформляют. Позже записывают ответ. Часто учитель помогает тем, у кого возникают трудности при решении. Главный фактор заданий в том, что они опираются на искомые величины и условия. Для этого и существует оценка задания. Чтобы ученики быстрее научились решать без помощи учителя, он увеличивает нагрузку, дает дополнительные задачи.

Погрешность 2. Пропуск шага нахождения решения.

Пропуск такого шага развивает неверное представление действительности, следовательно, к началу препятствий во время индивидуальной работы.

Привычным представляется такой случай, когда учитель вызывает тех, кто понимает задачи и сможет без затруднений решить их. Хотя при обучении база учителя ориентирована на тех учеников, у которых выявлены трудности при индивидуальной работе [4].

Ученикам, которые непосредственно быстро и без учителя решают задания, нужно давать задачи с повышенной трудностью. Это содействует росту их способностей.

Погрешность 3. Пропуск шага процесса решения.

Для чего важен данный шаг? На этом шаге устанавливаем, выполняется ли проверка. Говоря другими словами, подставляем под условие полученный результат. Стоит вопрос, что из решения является для учащихся полезным, и что они могут извлечь для дальнейшего решения иных задач. Данный вопрос допускает полагать, что навык решения задач ведет к увеличению умений решать задания.

Погрешность 4. Спутывание шагов оценки и нахождения решения.

Дабы это устранить, необходимо безошибочно понимать, что мы хотим получить на шаге, независимо какой он по счету.

Установка шага оценки решения – найти все обладающие звенья среди искомых и данных величин.

Установка шага нахождения решения – подобрать способ решения и создать его план. Цели этих шагов непохожие, то есть спутывать их запрещено.

На шаге оценки условия задания:

1. частично разделяем условие задания;
2. устанавливаем, что описывает в условии действие;
3. определяем, что дано, а что необходимо находить;
4. вводим взаимосвязи между ними [5].

На шаге нахождения решения определяем, что возможно выяснить по условию задания, и пригодится ли это дальше.

Конец этого шага состоит в сформировании плана.

Погрешность 5. На шаге оценки данных устанавливаются не все сцепления среди величин.

Нужно стремиться закрепить как можно больше связей. Для чего это необходимо? Не заметив одну из них, возможно лишиться:

а) условие для построения равенства;

б) вероятность показать 1 значение через другое;

в) обеспечить иные методы решения.

Преподаватель не должен направлять обучающихся к собственному решению: необходимо проанализировать все ответы, слушать и обговаривать их.

В результате были замечены трудности при решении финансовых задач, встречающиеся в школьном курсе математики. Так как отдается немного времени на освоение и их понимание, следует, что и ориентироваться учащимся в них довольно сложно.

Ученик должен иметь представление не только о финансах, но и о грамотной их растрате. Это нужно, чтобы в последующем правильно управлять личными финансами, понимать, что такое банковское дело и инвестиции в финансовом рынке, разбираться в вопросах страхования и различать мошенниче-

ство и финансовые пирамиды, которые на Российском рынке эксперты оценивают порядком 120, действующих в Российской Федерации.

Список литературы

1. Колобов, А.Н. О значении компьютерных технологий и математического моделирования в образовании бакалавров. / А.Н. Колобов, Т.М. Зубкова //Вестник Оренбургского государственного университета. – 2014. – №2. – С.118-124.

2. Колобов, А.Н. Применение интерактивных технологий в процессе подготовки к олимпиаде по математике. / А.Н. Колобов, И.В. Прояева // Мир науки, культуры, образования. Международный научный журнал. - 2017 г.- № 6.- с.169-175.

3. Атанасян, Л.С. Алгебра (базовый и профильный уровни).10 - 11кл. / Л.С. Атанасян, Бутузов В.Ф., Кадомцев С.Б – М.: Просвещение, 2013.

4. Колобов, А.Н. Об изучении векторной геометрии в современной школе. / А.Н. Колобов, И.В. Прояева // Мир науки, культуры, образования. Международный научный журнал. - 2017 г.- № 4.- с.199-203.

5. Колобов А.Н. Компьютерные технологии и высшее образование. / А.Н. Колобов // Университетский комплекс как региональный центр образования, науки и культуры [Электронный ресурс] : материалы Всерос. науч.-метод. конф., Оренбург 3-5 февр. 2016 г. / Оренбург. гос. ун-т. – Электрон. дан. – Оренбург : ОГУ, 2016. – с. 2499-2501. 1 электрон. опт. диск (CD-ROM). – ISBN 978-5-7410-1385-4.

ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В ПОДГОТОВКЕ БУДУЩЕГО БАКАЛАВРА ПРИ ИЗУЧЕНИИ МАТЕМАТИЧЕСКИХ ДИСЦИПЛИН

Максименко Н.В., Смирнова Е.Н.
Оренбургский государственный университет

Информационная подготовка будущих специалистов строится таким образом, чтобы она могли служить базой для формирования основ информационной культуры будущего бакалавра, при этом профессиональная деятельность должна находить свое отражение в решении конкретных прикладных задач с помощью современных информационных средств, таких как:

- использование электронных учебников;
- обучающие мультимедиа системы;
- программы контроля и самоконтроля знаний;
- использование информационных технологий в организации и проведении научных исследований.

В последнее время одной из основных проблем, над которой мы работаем, является роль информационных технологий в формировании профессионально-деловых качеств бакалавра при изучении математических дисциплин. Главными направлениями решения этой проблемы являются:

1. компьютеризация учебного процесса;
2. новое в информационных технологиях обучения;
3. информационная культура как составная профессиональной культуры специалиста;
4. роль электронных учебников в образовании студентов;
5. организация самостоятельной работы студентов с использованием ПК;
6. опыт проведения компьютерного контроля знаний;
7. эффективность использования мультимедийных технологий в учебном процессе.

В современных условиях компьютерное обучение целесообразно и возможно строить как личностно ориентированное, т.е. принимать во внимание психологические возможности студентов, а также специально предусматривать и создавать условия для развития их личности. На каждом этапе освоения информационных технологий преподавателем активизируются, мобилизуются имеющиеся у обучающихся личностные ресурсы, мотивы и интересы, способности и умения, а также специально развиваются, формируются новые качества, востребованные на более высоком уровне применения компьютера.

Важно не только приспосабливать бакалавра к новым информационным технологиям, но и обеспечивать адаптацию информационных технологий к потребностям и возможностям, запросам и способностям пользователей. Будущие специалисты должны обладать системой знаний и умений, позволяющих гра-

мотно использовать информационные технологии в профессиональной деятельности.

Информатизация образования развивается в рамках трех последовательных этапов:

- освоение информационных технологий как новой составляющей содержания высшего образования;

- использование информационных технологий как учебного средства при изучении существующих дисциплин вуза;

- разработка новых учебных дисциплин, ориентированных на обновленное содержание, цели и методы подготовки студентов в условиях становления информационного общества и органически использующих новейшие педагогические технологии для достижения этих целей.

Студенты используют информационные технологии в самостоятельной и исследовательской работе. Это решение и оформление контрольных, курсовых работ, поиск информации в Интернете для подготовки рефератов по различным дисциплинам, участие в олимпиадах и научно-исследовательской работе.

При слуховом восприятии закрепляются 15% языковой информации, при зрительном – 25% визуальной информации, слыша и видя одновременно, человек запоминает 65% информации, которая ему сообщается, поэтому очень важно применение мультимедийных технологий.

Использование мультимедийных технологий преследует, в основном, две цели. Первая – облегчить усвоение и запоминание учебного материала. Мультимедийные технологии в учебном заведении должны стать объектом для изучения, для того, чтобы будущий бакалавр мог оптимально их использовать.

Обеспечение необходимого уровня информационной культуры специалиста не может быть целью только одной учебной дисциплины, необходимо внедрение современных информационных технологий во все специальные дисциплины, что требует определенного уровня профессиональной подготовки преподавательского состава, его знакомства с потенциальными возможностями этих технологий, умением использовать эти возможности в своей практической и научной деятельности. Этот момент является весьма актуальным и педагогически значимым, так как студенты на деле, то есть в процессе учебно-тренировочных занятий, проведения научных исследований и т.д., должны видеть и на себе испытать преимущества и возможности современных информационных технологий.

Использование компьютерных сетей позволяет:

- получать доступ к самым разнообразным источникам информации, к отраслевым базам данных в области экономики, науки, образования, культуры, а также к правительственным, университетским, общественным, коммерческим базам и региональным хранилищам информации;

- принимать участие в электронных конференциях;

- получать информацию из различных районов земного шара по интересующей проблеме;

- общаться с коллегами, специалистами, работающими в самых разнообразных областях;

- связываться с международной образовательной системой;

- иметь доступ к электронным архивам программного обеспечения для персональных компьютеров.

В результате становится возможным решение следующих задач.

1. Индивидуализация процесса обучения. Например, компьютер позволяет осуществлять обучение по специальной авторской программе.

2. Налаживание действенной обратной связи (как в случае отрицательных, так и положительных ответов на вопрос). При этом компьютеры могут взять на себя рутинную, но трудоемкую работу по проверке знания таблиц, умений осуществлять математические и логические операции и др., давая возможность преподавателю заниматься творческой работой.

3. Увеличение скорости усвоения студентами материала. Компьютер осуществляет селекцию информации и представляет ее в удобной (графической или звуковой) форме. Среди причин, затрудняющих развитие готовности бакалавров к профессиональной деятельности на основе информационных технологий, выделяются следующие:

- отсутствие теоретически обоснованной системы непрерывной информационной подготовки специалиста,

- усиление теоретической подготовки в ущерб практической,

- недостаточное использование в учебном процессе новых информационных технологий.

Внедрение информационных технологий создает предпосылки для интенсификации учебного процесса. Они позволяют широко использовать на практике психолого-педагогические разработки, обеспечивающие переход от механического усвоения знаний к овладению умением самостоятельно приобретать новые знания.

Мы стараемся активно внедрять компьютерные технологии обучения. Разработан и применяется сборник тестов по математическим дисциплинам. Это позволяет оперативно и непредвзято проводить контроль знаний, умений и навыков студентов при их подготовке к отдельным занятиям, в конце учебного семестра, а также при сдаче экзаменов.

Опыт использования программированного контроля знаний, особенно с использованием персональных компьютеров, позволяет выделить его позитивные моменты, а именно:

- повышается объективность оценивания знаний студента;

- изменяется роль преподавателя, который освобождается от функции «наказания», связанной с выставлением оценок. Преподаватель перестает быть источником негативных эмоций, а приобретает роль консультанта;

- улучшается психологическая атмосфера в учебных группах, понятие «любимчиков» автоматически теряет смысл;

- резко возрастает оперативность получения результатов оценивания по сравнению с другими методами (устным и письменным опросом);

- ликвидируется возможность подсказки и списывания.

Информатизация образования в конечном итоге должна обеспечить доступность получения знаний и информации, развитие интеллектуальных и творческих способностей личности, повышение квалификации и оперативное изменение сферы деятельности каждого человека в течение активного периода жизни, а также необходимые условия для реализации опережающего образования и повышения эффективности дистанционных форм обучения.

Новые информационные технологии предъявляют повышенные требования к уровню квалификации (информационно-технической подготовленности) педагогических и руководящих работников вузов, который в значительной степени определяют прогресс в данном направлении. Позитивное влияние новых информационных и коммуникационных технологий на качество российского образования заключается в создании условий для повышения творческого и интеллектуального потенциала обучаемого за счет самоорганизации, стремления к знаниям, умения взаимодействовать с компьютерной техникой и самостоятельно принимать ответственные решения; интеграции современных электронных средств обучения с традиционными средствами обучения.

Список литературы

1. *Максименко, Н.В. Использование информационных технологий в подготовке будущего специалиста / Н.В. Максименко // Современные информационные технологии в науке, образовании и практике: материалы X Всерос. науч.-практ. конф. – Оренбург: ООО ИПК "Университет", 2012. – С. 464-465. – ISBN 978-5-4417-0097-9.*

ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ В ШКОЛЬНОМ КУРСЕ ГЕОМЕТРИИ

Никульшина А.А.
Оренбургский государственный педагогический университет

Геометрические преобразования — одни из основных идей современной математики. Они лежат как в основе определения геометрии, так и в основе классификации её отдельных разделов. Достаточно вспомнить определение геометрии, данное Феликсом Клейном (1849—1925) в его знаменитой Эрлангенской программе [1, с.108].

В Германии долгое время существовал обычай, согласно которому кандидат на замещение профессорской должности должен был выступить перед Ученым Советом с лекцией на свободно выбранную им тему; на основании этой лекции Ученый Совет делал заключение о возможности допущения данного лица к профессуре. Такая лекция Ф. Клейном (известным немецким математиком) была прочитана в 1872 г. в г. Эрлангене (Германия) и впоследствии получила название Эрлангенской программы Клейна.

Ф. Клейном впервые были сформулированы принципы теоретико-группового построения геометрии. Геометрия — это наука, изучающая свойства фигур, инвариантные относительно некоторой группы преобразований. В самом деле, проективная геометрия — геометрия проективной группы, аффинная — аффинной, а школьная геометрия — геометрия группы движений и подобий [2].

Широки практические приложения геометрических преобразований. Теория подобия проникла в физику и стала основой физического эксперимента. Она нашла приложение и в технике. В современной науке и технике широкое применение находит обобщенное понимание геометрического подобия и моделирования явлений. Геометрические преобразования имеют и большое воспитательное значение, с ними входят в геометрию диалектика, движение [3].

Если в науке идея геометрических преобразований завоевала всеобщее признание, то вопрос о целесообразности изучения геометрических преобразований в школе оставался открытым до недавнего времени. В настоящее время уже не стоит вопрос: изучать геометрические преобразования или нет. Вопрос в другом: как изучать.

Прежде чем заняться изучением геометрических преобразований в школе, дадим некоторые общие определения. Они носят несколько абстрактный характер, так как относятся к множествам, природа элементов которых для нас пока безразлична.

Определение 1. Отображением f множества M в множество M' называется такое правило, при котором каждому элементу m множества M соответствует единственный элемент m' множества M'. Элемент m' называется образом

элемента т, а элемент m называется прообразом элемента m' при отображении f.

$$f: M \to M'$$

$$f: m \to m' \quad m' = f(m)$$

Если при отображении f каждый элемент m' множества M' является образом по крайней мере одного элемента m множества M, то говорят, что множество M отображается на множество M'.

Определение 2. Отображение f множества M на множество M' называется взаимно однозначным, если разным элементам множества M соответствуют разные элементы множества M'.

Определение 3. Взаимно однозначное отображение f множества M на себя называется преобразованием множества M.

В геометрии мы также занимаемся отображением одного множества в другое, только элементами множества M являются точки плоскости или пространства и тогда говорят о геометрическом отображении одного множества в другое.

Определим геометрическое преобразование плоскости.

Определение 4. Пусть элементами множества M являются все точки плоскости. Взаимно однозначное отображение множества точек плоскости на себя называется геометрическим преобразованием плоскости.

Теперь посмотрим, как же определить геометрическое преобразование плоскости в школе. С этой целью обратим внимание на определение числовой функции, данное в курсе «Алгебра и начала анализа» по учебнику Н. Я. Виленкина [4].

Пусть X — числовое множество. Отображение, сопоставляющее каждому числу x из X некоторое число у, называют *числовой функцией,* заданной на X. Множество X называют областью определения функции f.

Тогда по аналогии естественно определить геометрическое преобразование следующим образом:

Пусть X — множество всех точек плоскости. Геометрическим преобразованием плоскости называется отображение этой плоскости, которое каждой точке плоскости ставит в соответствие некоторую точку этой же плоскости; при этом

1) различным точкам плоскости A и B соответствуют различные точки A' и B',

2) область определения и область значений совпадают с X.

Примечание. Если в школе не дано определение отображения одного множества на другое, то можно несколько упростить определение, а именно.

Пусть X — множество всех точек плоскости. Известно правило, которое каждой точке плоскости ставит в соответствие некоторую точку этой же плоскости, при этом

1) различным точкам плоскости A и B соответствуют различные точки A' и B',

2) область определения и область значений совпадают с X.

Тогда говорят, что задано геометрическое преобразование плоскости.

Сопоставляя геометрическое преобразование и числовую функцию, видим, что геометрическое преобразование и числовая функция — две модели общего понятия отображения одного множества на другое. Эти модели отличаются природой области определения и области значений, способом задания соответствия. Совершенно очевидно, что методика изучения геометрических преобразований должна быть ориентирована на подчеркивание идеи функции, которая играет здесь объединяющую роль, устраняя традиционную изолированность геометрии, поэтому преобразования в геометрии желательно изучать по тому же плану, что и функции в алгебре.

Методическая схема изучения геометрических преобразований

1. Определение.
2. Способы задания.
3. Свойства.
4. Применение к доказательству теорем и решению задач.

Список литературы

1. *Мацуо Комацу. Многообразие геометрии: Пер. с японского— М.: Знание, 1981.*

2. *И. В. Прояева, А. Д. Сафарова. Организация самостоятельной работы студентов по курсу «Преобразования плоскости и проективная геометрия», Издательство ОГПУ, Оренбург 2016.*

3. *И. В. Прояева, А. Д. Сафарова. Организация самостоятельной работы студентов по подготовке к ГИА курсу «Геометрия», Издательство ОГПУ, Оренбург 2016.*

4. *Виленкин Н.Я., Ивашев-Мусатов О.С., Шварцбурд С.И. Алгебра и начала математического анализа. 11 класс. Углубленный уровень 18-е изд., стер. - М.: 2014.*

РАСПРЕДЕЛЁННЫЕ ВЫЧИСЛЕНИЯ НА ОСНОВЕ DOCKER SWARM И TENSORFLOW ДЛЯ КЛАСТЕРОВ

Очередько О.О., Полежаев П.Н.
Оренбургский государственный университет

В настоящее время стало весьма актуально использование нейронных сетей для решения большого количества разнообразных задач. Их область применения не ограничена, нейронные сети используют для прогнозирования, распознавания образов, для анализа данных и кластеризации, принятия решений и управления, аппроксимации и оптимизации и т.д. Поскольку нейронная сеть для получения результата выполняет значительное количество векторных и матричных операций, наиболее подходящим способом её реализации является использование параллельного программирования на графических процессорах. Также в последние годы достаточно часто используют контейнеры для развертывания приложений, которые можно легко развернуть на распределенных системах, позволяя системе быстро масштабироваться и оставаться работоспособной при отказе отдельных машин или приложения.

Для управления Docker контейнерами необходим инструмент, с помощью которого можно создавать и управлять кластерами Docker на узлах, как единой виртуальной системой. Кластеризация является важной особенностью для контейнерных технологий, так как она создает совместную группу систем, которые могут обеспечить избыточность. Чаще всего в качестве такого инструмента выступает Docker Swarm.

Docker Swarm – это группа машин, на которых запущен Docker, при этом они соединены в единый кластер [1]. Функции управления кластерами и оркестровки (распределение контейнеров, управление кластером, и возможность добавления дополнительных машин), встроенные в Docker Engine, построены с использованием swarmkit. Swarmkit — это отдельный проект, который реализует уровень оркестровки Docker и используется непосредственно в Docker.

Swarm состоит из нескольких машин, которые могут быть физическими или виртуальными. При этом они работают либо как менеджеры (для управления членством и делегирования полномочий), либо как рабочие (которые выполняют службы swarm), либо выполняют обе роли. При создании сервиса можно определить её оптимальное состояние (количество реплик, доступных для сети и ресурсов хранения, порты, предоставляемые сервисом для внешнего мира, и многое другое). Docker поддерживает необходимое состояние. Например, если рабочий узел становится недоступным, Docker распределяет задачи этого узла на других узлах. Задача — это запущенный контейнер, который является частью службы swarm и управляется менеджером.

Одним из ключевых преимуществ сервисов swarm над автономными контейнерами является то, что можно изменить конфигурацию службы, включая сети и тома, к которой она подключена, без необходимости вручную переза-

пуска службы. Docker обновит конфигурацию, остановит задачи обслуживания с устаревшей конфигурацией и создаст новые, соответствующие желаемой конфигурации.

Для развертывания приложений на нескольких компьютерах с использованием графических процессоров NVIDIA используют Nvidia-docker. Nvidia-docker по существу является оберткой обычного Docker, которая прозрачно создает контейнер с необходимыми компонентами для выполнения кода на графическом процессоре [2].

Ещё до не давнего времени Nvidia-Docker не поддерживал режим Swarm. Но в конце декабря 2017 года вышла новая версия Docker 17.12.0-ce, в которой добавлена поддержка режима изоляции службы Swarm [3]. Теперь пользователь может настроить демон Docker так, чтобы графические процессоры были видны в Docker Swarm. Для этого необходимо выполнить следующие шаги:

1. Создать переопределение для конфигурации dockerd, изменив время выполнения по умолчанию и добавив ресурсы графического процессора. Флаги ресурсов можете сгенерировать следующим образом:

nvidia-smi -a | grep UUID | awk '{print "--node-generic-resource gpu="substr($4,0,12)}' | paste -d' ' -s

sudo systemctl edit docker

[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// --default-runtime=nvidia < ресурсы, приведенные выше >

2. Раскомментировать swarm-ресурс в /etc/nvidia-container-runtime/config.toml

3. Перезапустить демон Docker, создать swarm и создать новый сервис, запрашивающий графические процессоры, например:

docker service create -t -generic-resource "gpu = 1" ubuntu bash

Однако в некоторых случаях может возникнуть ошибка, если в файле конфигурации указан аргумент «node-generic-resource». Данную ошибку разработчики обещают исправить в ближайшее время. По этой причине в настоящее время рациональнее всего реализовывать распределённые вычисления для нейронных сетей через кластер TensorFlow с распределением части вычислений по графическим ускорителям и процессорам вычислительных узлов.

Кластер TensorFlow представляет собой набор «задач» («tasks»), которые участвуют в распределенном выполнении графа TensorFlow [4]. Каждая задача связана с сервером TensorFlow, который содержит «master» для создания сеан-

сов и «worker» для выполнения операций на видео ускорителях. Кластер также можно разделить на одно или несколько «заданий» («job»), где каждое задание содержит одну или несколько задач.

Для создания нового кластера необходимо во всех задачах запустить сервер TensorFlow, а затем выполнить следующие действия:

а) создать tf.train.ClusterSpec, который описывает все задачи в кластере. Он создаётся одинаковым для всех задач;

б) создать tf.train.Server, передав конструктору tf.train.ClusterSpec и определив локальную задачу с именем задания и индексом задачи.

Объект tf.train.Server содержит набор локальных устройств, набор подключений к другим задачам в tf.train.ClusterSpec и tf.Session, которые могут использовать их для реализации распределенных вычислений. Каждый сервер является членом определенного именованного задания и имеет индекс задачи в этом задании. Сервер может взаимодействовать с любым другим сервером в кластере.

Чтобы поместить операции в конкретный процесс, необходимо использовать функцию tf.device для определения места выполнении операции: на процессоре или на графическом процессоре.

Обычно в системе есть несколько вычислительных устройств. В TensorFlow поддерживаются такие типы устройств как CPU и GPU. В случае если операция будет назначена на CPU, а в системе имеются устройства обоих типов, то GPU будет присвоен приоритет [5].

Чтобы определить каким устройствам назначены операции, необходимо создать сеанс с параметром конфигурации log_device_placement, установленным в True.

Если необходимо, чтобы определенная операция выполнялась на некотором определённом устройстве, а не на том, которое было автоматически выбрано, необходимо указать это устройство в tf.device (например, tf.device('/cpu:0')). Таким образом, все операции в этом контексте будут иметь одинаковое назначенное устройство.

По умолчанию TensorFlow отображает память всех графических процессоров (с учетом CUDA_VISIBLE_DEVICES). Это делается для ещё более эффективного использования относительно ценных ресурсов памяти GPU на ускорителях за счет сокращения фрагментации памяти.

В некоторых случаях желательно, чтобы процесс выделял только подмножество доступной памяти или только увеличивал использование памяти. TensorFlow предоставляет два параметра конфигурации сеанса для его управления:

а) первый — это параметр allow_growth, который выделят такое количество памяти GPU, сколько необходимо для времени выполнения размещения: он начинает выделять достаточно малое количество памяти, и по мере того, как сеансы запускаются, расширяется область памяти GPU, необходимая процессу TensorFlow;

б) второй метод – это параметр per_process_gpu_memory_fraction, который определяет долю общего объема памяти, которую должен выделять каждый видимый графический процессор. Например, можно указать, чтобы TensorFlow выделял 50% общей памяти для каждого GPU, если прописать config.gpu_options.per_process_gpu_memory_fraction = 0.5.

Если нужно запустить TensorFlow на нескольких графических процессорах, то вы можете построить свою модель в режиме multi-tower. Для этого сначала создаётся массив со всеми необходимыми устройствами, а затем в цикле для каждого устройства задаются операции в tf.device().

Для того чтобы отследить производительность и другие характеристики графических процессоров в режиме реального времени необходимо воспользоваться командой «watch nvidia-smi», запускаемой в консоли. Она дает информацию о температуре графических процессоров, текущей используемой памяти, напряжении GPU, о том, какие задачи используют различные графические процессоры и др.

Таким образом, из-за некоторых ограничений Nvidia-Docker в Docker Swarm, рациональнее реализовывать распределённые вычисления для нейронных сетей через кластер TensorFlow с распределением части вычислений по графическим ускорителям и процессорам вычислительных узлов. Однако в скором времени, после исправления всех ошибок при использовании графических процессоров в Docker Swarm станет возможным создавать и быстро разворачивать кластеры на Docker машинах.

Исследование выполнено при финансовой поддержке Правительства Оренбургской области и РФФИ (проекты №17-47-560046, №16-29-09639 и №18-07-01446), Президента Российской Федерации в рамках стипендии для молодых ученых и аспирантов (СП-2179.2015.5).

Список литературы

1. JSwarm mode overview [Электронный ресурс] / Docker Documentation // URL: https://docs.docker.com/engine/swarm/ (дата обращения: 9.11.2017).

2. Очередько О.О., Полежаев П.Н. Сравнительный анализ обучения нейронной сети с использованием виртуальных машин и контейнеров NVidia-Docker // Современная техника и технологии: проблемы, состояние и перспективы: Материалы VII Всероссийской научно-практической конференции с международным участием 27-28 октября 2017 г. / Под ред. к.т.н., доцента С.А. Гончарова; к.ф-м.н., доцента Е.А. Дудник / Рубцовский индустриальный институт. – Рубцовск, 2017. – С. 65-72.

3. Docker CE release notes [Электронный ресурс] / Docker Documentation // URL: https://docs.docker.com/release-notes/docker-ce/ (дата обращения: 9.01.2018).

4. Distributed TensorFlow [Электронный ресурс] / TensorFlow // URL: https://www.tensorflow.org/deploy/distributed (дата обращения: 10.11.2017).

Using GPUs / TensorFlow [Электронный ресурс] // URL: https://www.tensorflow.org/tutorials/using_gpu (дата обращения: 10.

АРХИТЕКТУРА ПРОТОТИПА АВТОНОМНОЙ СИСТЕМЫ ОБЕСПЕЧЕНИЯ КИБЕРБЕЗОПАСНОСТИ И КАЧЕСТВА ОБСЛУЖИВАНИЯ ПРОГРАММНО-УПРАВЛЯЕМОЙ ИНФРАСТРУКТУРЕ МУЛЬТИОБЛАЧНОЙ ПЛАТФОРМЫ

Парфёнов Д.И. канд. техн. наук, Дедюрин В.В., Шардаков В.М.
Оренбургский государственный университет

В настоящее время доля использования технологии облачных вычислений для размещения приложений и сервисов в крупных коммерческих и государственных организациях, в том числе на промышленных предприятиях (в областях электроэнергетики, машиностроения, добычи и переработки полезных ископаемых и т.п.) постоянно растет [1-3]. При этом в виртуальную инфраструктуру переносят не только публичные ресурсы организации, но и сервисы, отвечающие за критически важные бизнес процессы, требующие обеспечения заданного качества обслуживания (QoS), а так же необходимого уровня информационной безопасности. Инфраструктура традиционных центров обработки данных (ЦОД) не позволяет в полной мере обеспечить гибкое управление сетевыми и вычислительными ресурсами [4-5]. Это в свою очередь негативно сказывается на параметрах, влияющих на работу облачных приложений и сервисов [6-8].

В рамках настоящего исследования разработан прототип автономной системы обеспечения кибербезопасности и качества обслуживания. Предложенное решение построено на базе современных подходов, используемых при организации виртуальной программно-управляемой инфраструктуры мультиоблачной платформы. В частности для организации базовой инфраструктуры среды передачи данных предлагаемого решения выбрана программно-конфигурируемая сеть (Software-defined networking, SDN). В свою очередь для эффективного использования сетевых ресурсов внутри построенной инфраструктуры использован подход, основанный на виртуализации сетевых функций (Network function virtualization, NFV).

Разработанный прототип является модульным и масштабируемым решением, что позволяет интегрировать его в состав любой системы управления облачными вычислениями. Прототип включает в себя следующий ряд программных компонентов:

1) Модуль глубокого анализа данных, который осуществляет сбор необходимой для принятия решений о фильтрации трафика информации на сетевых и вычислительных узлах мультиоблачной платформы.

2) Модуль контроллера сетевой безопасности, который на основе алгоритма межсетевого экранирования осуществляет управление правилами доступа к ресурсам в сетевой среде мультиоблачной платформы.

3) Модуль обеспечения качества обслуживания в своей работе использует алгоритм самоорганизации управления адаптивной маршрутизацией сетевого трафика в программно-конфигурируемой сети для управления потоками данных приложений и сервисов в мультиоблачной платформе.

4) Модуль управления инфраструктурой мультиоблачной платформы, осуществляет размещение приложений и сервисов в сетевой среде мультиоблачной платформы.

Разработанные модули адаптированы для работы контейнеров на базе Docker, что позволяет быстро разворачивать их в сетевой среде мультиоблачной платформы. Архитектура предлагаемого решения представлена на рисунке 1.

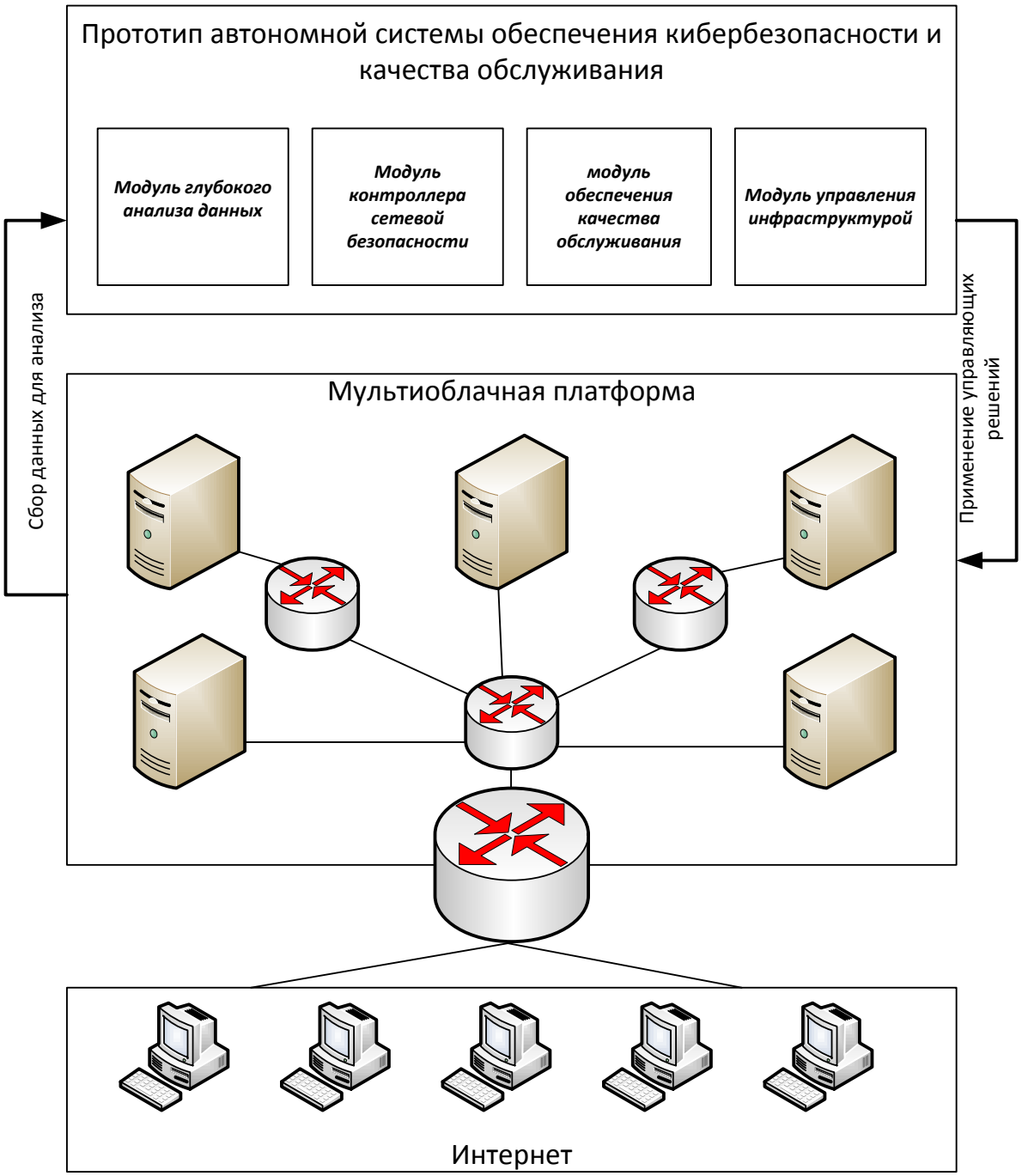

Рисунок 1 – Архитектура прототипа автономной системы обеспечения кибербезопасности и качества обслуживания программно-управляемой инфраструктуре мультиоблачной платформы

Для исследования разработанного прототипа на базе Оренбургского государственного университета построена экспериментальная площадка, включающая в себя два виртуальных ЦОД. В первом виртуальном ЦОД была развернута мультиоблачная платформа (целевая среда кибератаки), построенная на базе OpenStack. Внутри мультиоблачной платформы запущен набор типовых о приложений и сервисов (цели кибератаки) характерных для корпоративных пользователей. Для реализации сценария самой кибератаки на базе второго виртуального ЦОД были определены два сегмента сети. В состав первого сегмента входили легитимные пользователи, отправляющие запросы к целевым приложениям в рабочем режиме. Во втором сегменте была развернуты вычислительные узлы (атакующие агенты), реализованные на базе виртуальных машин и генерирующие вредоносный трафик, направленный к приложениям мультиоблачной платформы.

Для исследования особенностей работы прототипа в экспериментальном исследовании использовались потоки вредоносного трафика различной интенсивности. Кроме того для оценки эффективности предлагаемого решения в плане обеспечения качества обслуживания поступающих от легитимных пользователей проводилась оценка нарушений требований QoS и измерялось время отклика приложений и сервисов внутри облачной системы. Сопоставление результатов полученных при работе прототипа (Активный режим) проводилось с и типовыми модулями обычной облачной системы OpenStack (Пассивный режим). Результаты экспериментального исследования представлены в таблице 1

Таблица 1 - Результаты экспериментального исследования

№ эксперимента	Скорость поступления вредоносного трафика, Гбит/с.	Время отклика приложений и сервисов внутри облачной системы, мс		Процент нарушений требований QoS, %	
		Активный режим	Пассивный режим	Активный режим	Пассивный режим
1	0,10	45	90	0,01	0,10
2	0,20	48	120	0,20	0,20
3	0,30	50	150	0,50	30
4	0,40	60	180	1,80	25
5	0,50	65	220	2,50	30

Экспериментальные исследования показали, что разработанный прототип системы позволяет не только существенно сократить время отклика приложе-

ний и сервисов в сети мультиоблачной платформы при проведении кибератак, но и поддерживать заданное качество обслуживания на требуемом уровне.

В дальнейшем планируется исследовать работу прототипа на предмет ресурсоемкости, а также поведение при различных типах кибератак.

Исследование выполнено при финансовой поддержке РФФИ (проекты 16-37-60086, 16-07-01004, 18-07-01446) и гранта Президента Российской Федерации для государственной поддержки молодых российских ученых - кандидатов наук (МК-1624.2017.9).

Список литературы

1 Болодурина И.П., Парфёнов Д.И. Управление потоками данных в высоконагруженных информационных системах, построенных на базе облачных вычислений [Текст] / И.П. Болодурина, Д.И. Парфёнов // Системы управления и информационные технологии. – 2015. - № 1.1. – С. 111-118.

2 Bolodurina I.P., Parfenov D.I. Dynamic routing algorithms and methods for controlling traffic flows of cloud applications and services [Текст] / Bolodurina I.P., Parfenov D.I. // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. – 2017. - Т 6. - № 2. – С. 84-98.

3 Пальчевский, Е.В. Разработка системы обнаружения низкоактивного несанкционированного сетевого трафика / Е.В. Пальчевский, А.Р. Халиков // Перспективные информационные технологии. Изд-во: «СНЦ РАН», Самара, 2017. – С. 266-269.

4 Rafique M., Chen P., Huygens C., Joosen W. Evolutionary algorithms for classification of malware families through different network behaviors //Proceedings of the 2014 conference on Genetic and evolutionary computation. ACM, 2014. С. 1167–1174.

5 Bakhareva N.F., Polezhaev P.N., Ushakov Yu.A., Shukhman A.E. SDN-based firewall implementation for large corporate networks // Proceedings of 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT 2017), 2017. – P. 313-318.

6 Частикова, В.А. Обнаружение DDoS-атак на основе нейронных сетей с применением метода роя частиц в качестве алгоритма обучения / В.А. Частикова, К.А. Власов, Д.А. Картамышев // Фундаментальные исследования. № 8-4. Изд-во: «Академия Естествознания», Пенза, 2014. – С. 829-832.

7 Коржов, В. Современные DDoS-атаки / В. Коржов // Журнал сетевых решений LAN. № 9. Изд-во: «Открытые системы», Москва, 2016. – С. 55-57.

8 Юнг, Й.Ф. Защита от DDoS-атак из облака / Й.Ф. Юнг // Журнал сетевых решений LAN. № 5. Изд-во: «Открытые системы», Москва, 2014. – С. 63-65.

ПРОТОТИП СИСТЕМЫ УПРАВЛЕНИЯ ОБЛАЧНЫМИ РЕСУРСАМИ ДЛЯ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ, ПОСТРОЕННЫХ НА БАЗЕ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ

Полежаев П.Н.
Оренбургский государственный университет

В настоящее время облачные решения получили широкое распространение в сфере информационных технологий. С точки зрения провайдера весьма актуальным является предоставление облачных услуг максимально эффективным образом. В рамках данной работы было предложено соединить облачные технологии с программно-конфигурируемыми сетями (Software Defined Networking, SDN) и с технологией виртуализации сетевых функций (Network Function Virtualization, NFV). В результате были разработаны алгоритмы [1, 2], обеспечивающие эффективное планирование виртуальных машин, назначение на них контейнеров облачных приложений, также были созданы алгоритмы проактивной и реактивной маршрутизации потоков данных для программно-конфигурируемых сетей.

Данные алгоритмы были реализованы в виде отдельных модулей прототипа системы управления облачными ресурсами (см. рисунок 1).

Сервер контроллера SDN (Software Defined Networking – программно-конфигурируемой сети) представляет собой контроллер Ryu. Для него были разработаны следующие программные модули:

а) Модуль межсетевого экрана для облачных систем – с помощью OpenFlow реализует блокировку трафика на уровнях L2-L4 с поддержкой контроля состояния для протоколов TCP и UDP. Алгоритм работы данного модуля описан в [3].

б) Модуль сбора информации о состоянии программно-конфигурируемой сети облачной системы – реализует алгоритм сбора информации о топологии и состоянии сети [4], который основывается на сочетании использования протоколов LLDP, SNMP и OpenFlow в качестве источников информации для получения сведений о состоянии сети. Собираемая информация активно применяется другими модулями прототипа.

в) Модуль построения коммуникационных схем сервисов на основе статистики их взаимодействия – реализует алгоритм [1], который анализирует файл Docker Compose облачного приложения с целью выявления его отдельных микросервисов и собирает статистику об их сетевой активности с помощью счетчиков OpenFlow.

г) Модуль реактивной и проактивной маршрутизации потоков данных – реализует генетический алгоритм маршрутизации потоков данных для реактив-

ного случая и алгоритм Дейкстры для проактивного [2]. В первом случае проактивная маршрутизация выполняется до запуска облачного приложения после определения назначенных виртуальных машин. Генетический алгоритм решает оптимизационную задачу по максимизации оценки степени соблюдения требований к QoS – максимальной гарантированной задержки и минимальной гарантированной пропускной способности.

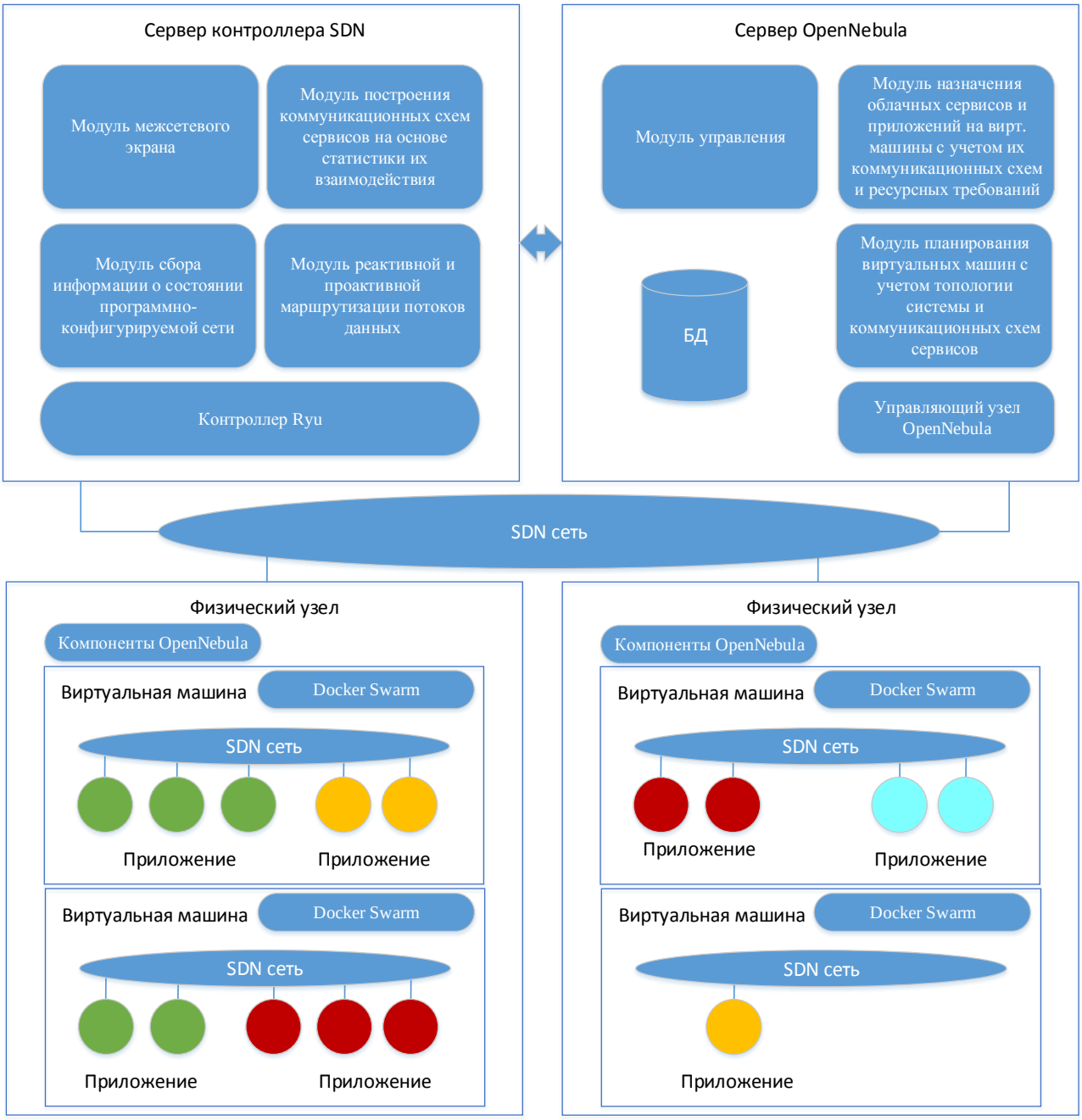

Рисунок 1 – Логическая архитектура прототипа системы управления

Во втором случае реактивная маршрутизация выполняется для динамически возникающего трафика, о котором нет информации заранее. Здесь было предложено использовать модифицированный вариант алгоритма Дейкстры [2].

Сервер OpenNebula представляет собой управляющий узел для системы управления облачной системы OpenNebula, которая была выбрана в качестве основы для реализации данного прототипа. Он включает в себя следующие модули:

а) Модуль управления – предоставляет простой Web-интерфейс для работы конечных пользователей с прототипом. С его помощью можно запустить облачное приложение и управлять им.

б) БД – база данных прототипа, содержащая информацию остальных модулей. Реализована с помощью СУБД PostgreSQL.

в) Модуль назначения облачных сервисов и приложений на виртуальные машины с учетом их коммуникационных схем и ресурсных требований. Он реализует разработанный генетический алгоритм, оптимизирующий целевую функцию – линейную свертку оценок использования вычислительных и коммуникационных ресурсов [2]. Алгоритм оптимизирует всю целевую функцию путем подбора оптимального отображения микросервисов облачного приложения на существующие и новые запускаемые виртуальные машины (за счет свободных ресурсов физических серверов). При этом для получения коммуникационной оценки каждый раз вызывается модуль проактивной маршрутизации потоков данных, который прокладывает все необходимы маршруты с учетом требований к QoS.

г) Модуль планирования виртуальных машин с учетом топологии системы и коммуникационных схем сервисов, запускаемых внутри виртуальных машин. Данный модуль заменяет модуль OpenNebula и решает две задачи: запуск группы виртуальных машин для назначения компонентов запускаемого облачного приложения (с помощью генетического алгоритма) [2], запуск одиночной виртуальной машины для размещения компоненты (микросервиса) облачного приложения (с помощью жадного алгоритма) [1].

На каждом физическом узле, используемом для развертывания облачных приложений и сервисов, были установлены обслуживающие компоненты OpenNebula. Через них происходит запуск виртуальных машин, в которых функционирует Docker Swarm. С помощью Docker Swarm происходит развертывания облачных приложений, состоящих из компонент (микросервисов), описываемых в виде файлов Docker Compose.

На рисунке 1 различные облачные приложения изображены в виде кругов внутри виртуальных машин одного цвета. Облачные приложения могут быть распределены по нескольким виртуальным машинам, которые в общем случае могут выполняться на нескольких узлах.

Связность приложений и инфраструктурных компонент обеспечивается с помощью SDN.

Прототип был развернут на базе оборудования Оренбургского государственного университета, его физическая архитектура изображена на рисунке 2.

Экспериментальный сегмент облачной системы располагается в телекоммуникационной стойке и включает в себя: сетевое хранилище данных NetGear ReadyNAS, два OpenFlow коммутатора HP 3500, два OpenFlow коммутатора NetGear 7224, сервер OpenNebula Aquarius Server T50, сервер контроллера SDN, (Intel Xeon, 4 ядра, 32 Гб оперативной памяти), 8 вычислительных узлов и шлюз для подключения к сети Интернет.

Коммутаторы NetGear 7224 и H33500 соединены в топологию кольца. Достоинством такой структуры является то, что каждый узел физически соединен со всеми остальными, что обеспечивает высокую степень избыточности. Если какой-либо канал выходит из строя, то существует резервный маршрут, позволяющий передать данные в пункт назначения.

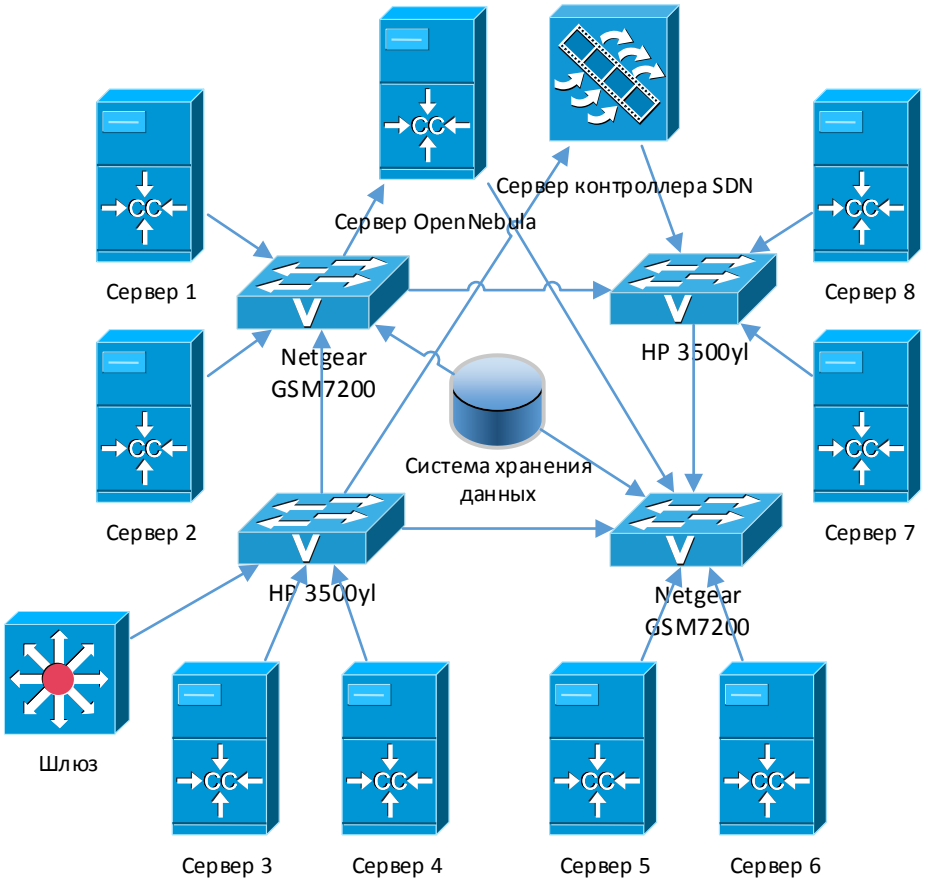

Рисунок 2 – Физическая архитектура прототипа системы управления

Сервера OpenNebula и контроллера OpenFlow подключены к двум разным коммутаторам с целью обеспечения избыточности и надежности. Серверы 1-8 используются для развертывания виртуальных машин и облачных приложений.

В экспериментальном исследовании участвовали облачные приложения следующих типов:

а) Neural Network – нейросетевое приложение в процессе обучения на наборе данных CIFAR-10 [5].

б) QSS – приложение имитационного моделирования.

в) Media Converter – приложение автоматической конвертации мультимедийных файлов.

Все входные данные, необходимые для работы данных приложений, размещались во внешних docker volume, которые располагались в системе хранения данных.

Экспериментальное исследование сервисов производилось в смешанном режиме. Для всех типов облачных приложений генерировался собственный экспоненциальный поток заявок на их запуск с типовыми значениями интенсивности $\lambda_{NN} = 0.01 \cdot L$, $\lambda_{QSS} = 0.1 \cdot L$ и $\lambda_{Media} = 0.02 \cdot L$ для соответствующих серви-

сов. Здесь $\cdot L \in [1,100]$ – коэффициент загрузки, позволяющий исследовать эффект масштабирования. С целью автоматической генерации потока заявок было написано отдельное вспомогательное приложение.

Экспериментальное исследование проводилось для различных значений коэффициента загрузки L с определенным дискретным шагом. Для каждого конкретного значения L проводилось по 20 повторений эксперимента, получаемые значения метрик усреднялись.

Сравнение производилось между разработанным прототипом и обычной облачной системы. В качестве последней использовался прототип, запущенный в минимальном режиме – стандартный планировщик виртуальных машин OpenNebula, стандартный модуль назначения, алгоритм маршрутизации OSPF, SDN не используется.

На рисунке 3 представлены графики зависимости средней загруженности вычислительных ядер серверов от коэффициента загрузки. Прототип демонстрирует увеличение средней загруженности вычислительных ядер на 3-5 % в зависимости от коэффициента загрузки.

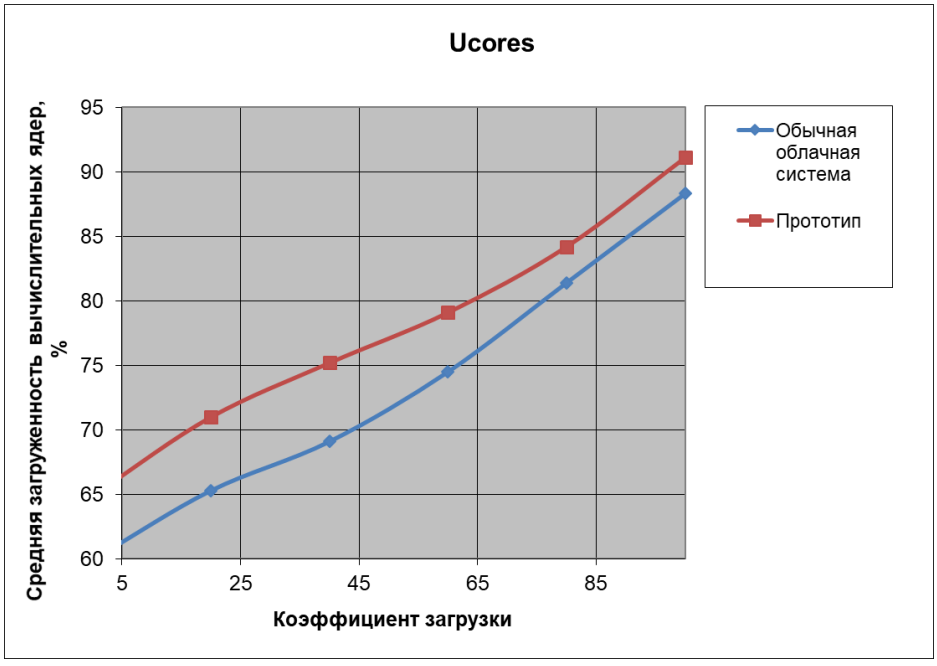

Рисунок 3 – Графики зависимости средней загруженности вычислительных ядер серверов от коэффициента загрузки

Данные значения согласуются с результатами, полученными ранее с помощью симулятора облачной системы и программно-конфигурируемой сети – с увеличением коэффициента загрузки растет значение показателя, также демон-

стрируется улучшение в сравнении со стандартными решениями. Показатели улучшения того же порядка, что и на симуляторе.

На рисунке 2 представлены графики зависимости процента нарушений требований QoS от коэффициента загрузки. Здесь видно, что прототип показывает меньше потерь, что также согласуется с результатами, полученными с помощью симулятора.

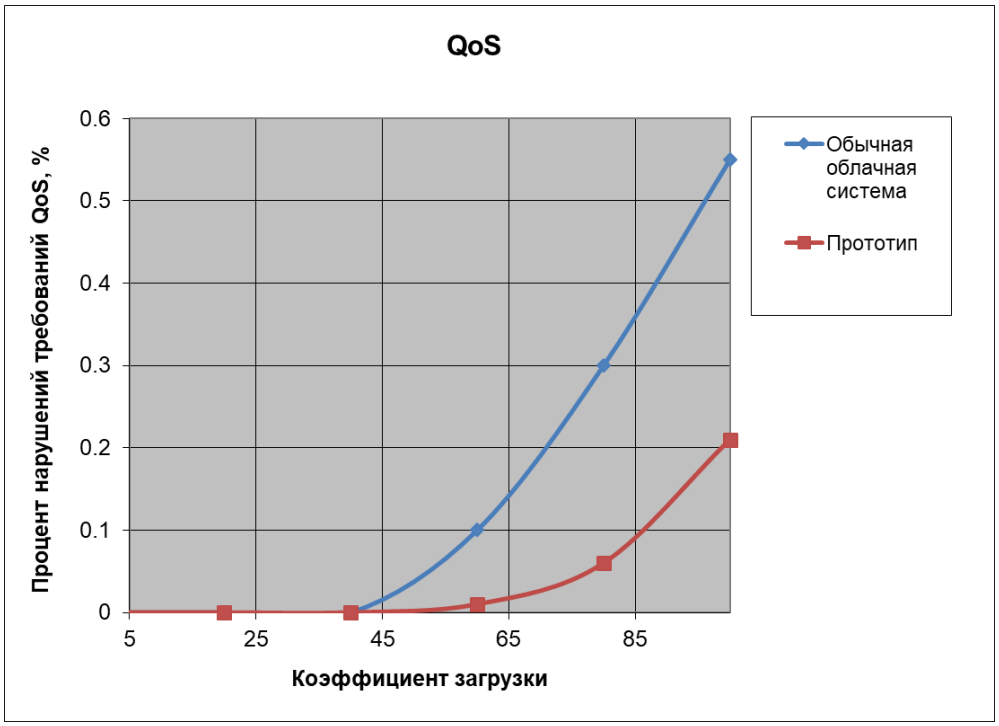

Рисунок 4 – Графики зависимости процента нарушений требований QoS от коэффициента загрузки

Разработан прототип системы управления облачными ресурсами для центров обработки данных, построенных на базе программно-конфигурируемых сетей. Описаны его логическая и физическая архитектуры. Проведенные экспериментальные исследования продемонстрировали его эффективность.

Созданный прототип может быть использован владельцами публичных и приватных облачных ЦОД, а также компаниями-разработчиками облачных ЦОД под ключ. Использование данного прототипа, включающего все разрабатываемые алгоритмические решения, позволит компаниям увеличить эффективность использования физических ресурсов, а также повысить качество предоставляемых услуг. Последний фактор очень важен для конечных пользователей облачных сервисов.

Работа выполнена при поддержке Президента Российской Федерации, стипендия для молодых ученых и аспирантов (СП-2179.2015.5).

Список литературы

1 Полежаев П.Н. Решение задач планирования виртуальных машин и сбора статистики о работе облачных приложений // Университетский комплекс как региональный центр образования, науки и культуры [Электронный ресурс]: материалы Всероссийской научно-методической конференции; Оренбург. гос. ун-т. - Электрон. дан. - Оренбург: ОГУ, 2017. – С. 3178-3184

2 Полежаев П.Н. Создание эффективных алгоритмов функционирования облачных систем // Университетский комплекс как региональный центр образования, науки и культуры [Электронный ресурс]: материалы Всероссийской научно-методической конференции; Оренбург. гос. ун-т. - Электрон. дан. - Оренбург: ОГУ, 2017. – С. 3185-3193

3 Bakhareva N.F., Polezhaev P.N., Ushakov Yu.A., Shukhman A.E. SDN-based firewall implementation for large corporate networks // Proceedings of 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT 2017), 2017. – P. 313-318.

4 Ушаков Ю.А., Полежаев П.Н., Бахарева Н.Ф., Коннов А.Л. Сбор и обобщение информации с сетевых устройств и виртуальных сетевых модулей в рамках сегмента сети NFV // Интеллект. Инновации. Инвестиции, 2017. - №10. – С 82-87.

5 Порохненко Ю.С., Полежаев П.Н. Классификация изображений набора данных CIFAR-10 с помощью нейронных сетей // Компьютерная интеграция производства и ИПИ-технологии: материалы VIII Всероссийской научно-практической конференции. – Оренбург, 2017. – С. 299-303.

ИСПОЛЬЗОВАНИЕ MAPREDUCE И HDFS ДЛЯ ХРАНЕНИЯ БОЛЬШИХ ДАННЫХ

Полежаев П.Н., Порохненко Ю.С.
Оренбургский государственный университет

В настоящее время существует проблема хранения большого объёма данных, которые могут быть необходимы для анализа сетевого трафика, поступают с различных физических экспериментальных установок или используются для обучения и эксплуатации нейронных сетей. Для решения этой проблемы существуют модель MapReduce и файловая система Hadoop Distributed File System (HDFS).

MapReduce – это программная среда для удобного написания приложений, которые обрабатывают огромное количество данных в параллельном режиме на больших кластерах (тысячи узлов) надежным, отказоустойчивым способом [1]. MapReduce разбивает входные данные на независимые фрагменты, которые обрабатываются полностью параллельно. Работа MapReduce состоит из двух шагов, называемых map и reduce, как и основополагающие функции фреймворка. На первом шаге происходит предварительная обработка данных, один из узлов (master node) разделяет все входные данные и передаёт их другим узлам (worker node). На них происходит создание наборов «ключ-значение». На втором шаге происходит обработка полученных блоков данных, объединяются все промежуточные значения, связанные одним ключом, после чего рабочие узлы возвращают обработанные блоки главному узлу, на их основе формируется решение поставленной задачи. При этом все вызовы функций на разных узлах проходят независимо, и функции выполняются параллельно. Фреймворк заботится о планировании задач, контролирует их и повторно выполняет неудавшиеся задачи.

MapReduce состоит из одного главного планировщика JobTracker и одного подчиненного TaskTracker, по одному на каждый узел кластера. Ведущий отвечает за планирование задач на подчиненных устройствах, мониторинг и повторное выполнение неудачных задач. Планировщики на узлах выполняют задачи по указанию мастера.

Как правило, все вычисления и хранение данных происходят на одном узле, то есть фреймворк MapReduce и распределенная файловая система работают на одном и том же узле. Такая конфигурация позволяет инфраструктуре эффективно планировать задачи на узлах, где уже присутствуют данные, что приводит к очень высокой совокупной пропускной способности в кластере.

Стандартный подход к работе с объемными данными – это использование базы данных и большого количества дисков. Главная проблема таких носителей заключается в том, что со временем время поиска улучшается медленнее, чем скорость передачи. Поиск — это процесс перемещения головы диска в определенное место на диске для чтения или записи данных. Он характеризует ла-

тентность операции с диском, тогда как скорость передачи соответствует пропускной способности диска. Если в шаблоне доступа к данным преобладает поиск, потребуется больше времени для чтения или записи больших частей набора данных, чем для потоковой передачи.

С другой стороны, для обновления небольшой доли записей в базе данных чаще всего используется B-Tree (структура данных, используемая в реляционных базах данных, которая ограничена скоростью, с которой он может выполнять поиск). B-Tree менее эффективен, чем MapReduce, который использует сортировку слиянием для пересоздания базы данных, когда необходимо обновить большой объём данных. Во многих отношениях MapReduce можно рассматривать как дополнение к системе управления реляционной базой данных (РСУБД). Фреймворк хорошо подходит для решения задач, в рамках которых необходимо анализировать весь набор данных в пакетном режиме. СУРБД применяется для точечных запросов или обновлений, где набор данных проиндексирован для обеспечения небольшого времени поиска и обновления. MapReduce подходит для приложений, в которых данные записываются один раз и читаются много раз, тогда как реляционная база данных применяется для наборов данных, которые постоянно обновляются.

В таблице 1 представлена сравнительная характеристика РСУБД и MapReduce.

Таблица 0 – Сравнительная характеристика РСУБД и MapReduce

Параметр	РСУБД	MapReduce
Размер данных	Гигабайты	Петабайты
Доступ	Интерактивный и пакетный	Пакетный
Обновления	Многоразовые чтение и запись	Одноразовая запись, многоразовое чтение
Операции	ACID	-
Структура	Основана на записи	Основана на чтении
Целостность	Высокая	Низкая

Работа с нейронными сетями подразумевает частое чтение данных и редкое обновление, поэтому MapReduce подходит для решения задачи распределения данных, предназначенных для обучения и эксплуатации нейронных сетей.

Hadoop HDFS – это распределенная файловая система, предназначенная для работы на кластерах из огромного количества узлов [2]. Она имеет много общего с существующими распределенными файловыми системами, однако отличия от них значительны. HDFS отличается высокой отказоустойчивостью и предназначена для развертывания на стороннем оборудовании. HDFS обеспечивает высокопроизводительный доступ к данным приложения и подходит для приложений с большими наборами данных. Первоначально HDFS была построена как инфраструктура для проекта веб-поиска Apache Nutch, сейчас является

частью проекта Apache Hadoop Core. Hadoop разработан в рамках идеи MapReduce и считается одной из главных составляющих технологии Big Data.

Экземпляр HDFS может состоять из сотен или тысяч серверных машин, каждая из которых хранит часть данных файловой системы. Тот факт, что существует огромное количество компонентов и каждый из них имеет нетривиальную вероятность отказа, означает, что какой-то компонент HDFS всегда нефункционален, поэтому обнаружение ошибок и быстрое автоматическое восстановление являются основной архитектурной целью HDFS.

Приложения, работающие на HDFS, нуждаются в потоковом доступе к своим наборам данных. Они не являются приложениями общего назначения, которые обычно запускаются на обычных файловых системах. HDFS больше подходит для пакетной обработки, основное внимание в фреймфорке уделяется высокой пропускной способности доступа к данным. Набор стандартов для обеспечения совместимости операционных систем и переносимости прикладных программ на уровне исходного кода (POSIX) накладывает множество жестких требований, которые не нужны для приложений, предназначенных для HDFS, поэтому некоторые требования были смягчены для увеличения пропускной способности данных и обеспечения потокового доступа к данным файловой системы.

Приложения, использующие HDFS, работают с большими наборами данных. Типичный размер файла HDFS – от 100 Гб до нескольких петабайт. Таким образом, HDFS настроен на поддержку больших файлов и должен обеспечивать высокую совокупную пропускную способность, масштабироваться до сотни узлов в одном кластере, а также поддерживать десятки миллионов файлов. Вычисление, которое выполняет приложение, намного эффективнее, если оно выполняется вблизи данных, с которыми оно работает. Это особенно актуально, когда размер набора данных огромен, т.к. минимизируется перегрузка сети и увеличивается общая пропускная способность системы. В связи с этим лучше переносить вычисления на узел, где находятся данные, а не перемещать данные туда, где выполняется приложение. HDFS подходит приложений, которые находятся максимально близко к узлу, на котором находятся данные. Hadoop разработан таким образом, чтобы быть легко переносимым с одной платформы на другую. Это облегчает широкое внедрение HDFS в качестве платформы для большого набора приложений.

Hadoop не является первой распределенной системой для хранения и анализа данных, но он обладает некоторыми особенностями, которые отличает её от других похожих систем. С развитием HDFS различия между реляционными базами данных и системами Hadoop уменьшаются. Реляционные базы данных начали включать некоторые идеи из Hadoop, системы HDFS, такие как Hive, становятся более интерактивными и обладают такими функциями, как индексы и транзакции, которые делают их более похожими на традиционные РСУБД.

Основным различием между Hadoop и РСУБД является структура наборов данных, с которыми они работают. Структурированные данные, которые

имеют определенный формат, например, документы XML или таблицы базы данных, которые соответствуют определенной схеме — это область РСУБД. Менее структурированные, например, электронная таблица, в которой структура представляет собой сетку ячеек, которые могут содержать любую форму данных, как обычный текст, так и данные изображения, – это область HDFS. Hadoop хорошо работает с неструктурированными или полуструктурированными данными, поскольку он предназначен для интерпретации данных во время обработки. Это обеспечивает гибкость и позволяет избежать дорогостоящей фазы загрузки данных в РСУБД, поскольку в Hadoop это всего лишь копия файла.

HDFS имеет архитектуру master/slave. Кластер HDFS состоит из одного Namenode, главного сервера, который управляет пространством имен файловой системы и регулирует доступ к файлам клиентами, и нескольких Datanodes, которые располагаются на каждом узле в кластере и управляют хранилищем, прикрепленным к узлам, на которых они работают. HDFS предоставляет пространство имен файловой системы и позволяет сохранять пользовательские данные в файлах. Внутри файл разбивается на один или несколько блоков, и эти блоки хранятся в наборе Datanodes. Namenode выполняет операции с пространством имен файловой системы, такие как открытие, закрытие и переименование файлов и каталогов. Он также определяет отображение блоков в Datanodes. Datanodes отвечают за обслуживание запросов на чтение и запись от клиентов файловой системы. Datanodes также выполняют создание, удаление и репликацию блока по команде Namenode. На рисунке 1 представлена архитектура HDFS.

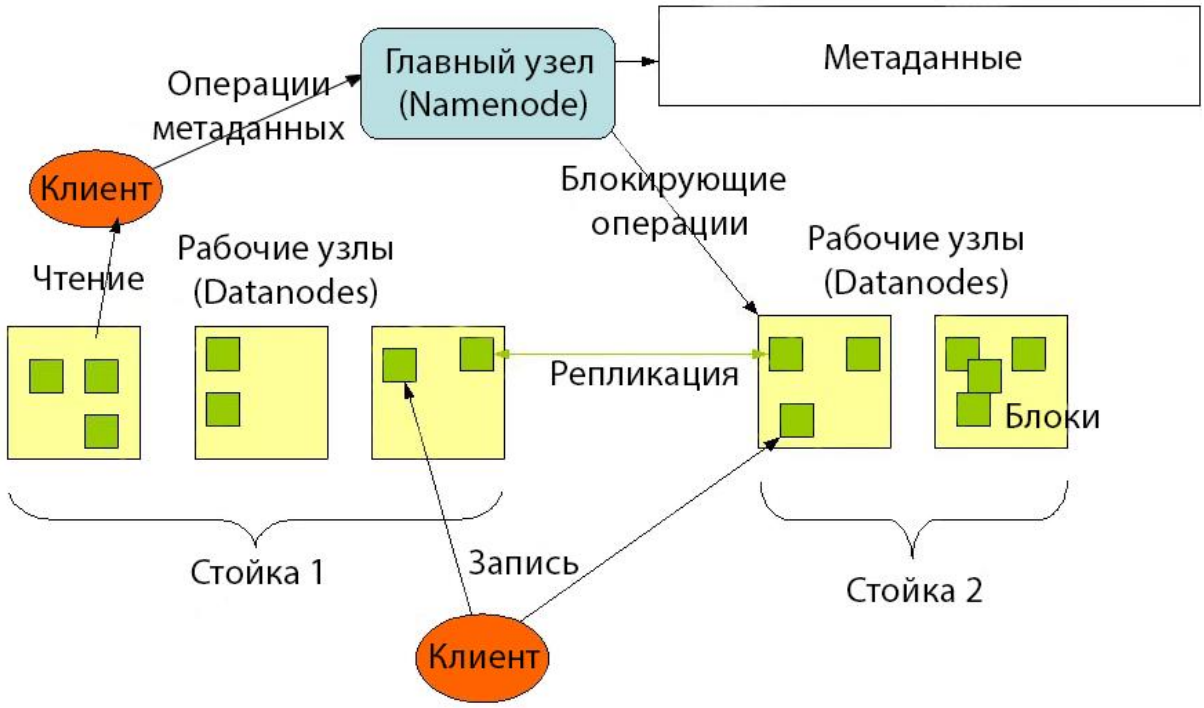

Рисунок 0 – Архитектура HDFS

Типичное развертывание HDFS состоит из выделенного компьютера, на котором работает только Namenode. Каждая из других машин в кластере запускает один экземпляр Datanode. Архитектура не исключает запуск нескольких Datanodes на одном компьютере, но в реальном развертывании такое происходит редко. Существование одного Namenode в кластере значительно упрощает архитектуру системы, т.к. он является арбитром и репозиторием для всех метаданных HDFS. Система разработана таким образом, чтобы пользовательские данные никогда не проходили через Namenode. Главный узел поддерживает пространство имен файловой системы. Любые изменения в пространстве имен файловой системы записываются в Namenode, помимо этого он хранит информацию о количестве копий файла (коэффициент репликации). Приложение может указывать количество реплик файла, которое должен поддерживать HDFS.

HDFS поддерживает традиционную иерархическую файловую организацию. Пользователь или приложение могут создавать каталоги и хранить файлы внутри этих каталогов. Иерархия пространства имен файловой системы похожа на большинство других существующих файловых систем; можно создавать и удалять файлы, перемещать файл из одного каталога в другой или переименовывать файл. В настоящий момент HDFS еще не реализует квоты пользователей или разграничение доступа и не поддерживает жесткие ссылки или программные ссылки, однако, архитектура HDFS не исключает возможности реализации этих функций.

HDFS хранит каждый файл в виде последовательности блоков; все блоки в файле, кроме последнего блока, имеют одинаковый размер. Каждый блок реплицируется для отказоустойчивости, при этом размер блока и коэффициент репликации настраиваются для каждого файла. Коэффициент репликации указывается во время создания файла и может быть изменен позже. Файлы в HDFS записываются один раз и имеют строго одного автора. Namenode принимает все решения относительно репликации блоков. Он периодически получает от каждого узла в кластере список всех блоков и ошибок (Blockreport) и ответ (Heartbeat), получение которого подразумевает, что Datanode работает правильно.

Размещение реплик имеет решающее значение для надежности и производительности HDFS. Оптимизация размещения реплик отличает HDFS от большинства других распределенных файловых систем. Размещение реплик повышает надежность, доступность и пропускную способность сети. Чтобы минимизировать глобальное потребление полосы пропускания и латентность чтения, HDFS пытается удовлетворить запрос на чтение от ближайшей к читателю реплики. Если существует реплика на той же стойке, что и считывающий узел, то эта реплика предпочтительна для удовлетворения запроса на чтение. Если кластер HDFS охватывает несколько центров обработки данных, то ре-

плика, которая находится в локальном центре обработки данных, предпочтительнее любой удаленной реплики.

Каждый Datanode периодически отправляет сообщение Heartbeat в Namenode. Нахождение узлов в разных сетях может привести к тому, что поднабор Datanodes потеряет связь с Namenode. Namenode обнаруживает это из-за отсутствия сообщения Heartbeat и отмечает не ответивший узел как мертвый и не передает к нему никаких новых запросов на чтение и запись к ним. Любые данные, зарегистрированные в мертвом Datanode, больше не доступны для HDFS. Отключение узла может привести к тому, что коэффициент репликации некоторых блоков упадет ниже их заданного значения. Namenode постоянно отслеживает, какие блоки необходимо реплицировать, и при необходимости инициирует репликацию. Потребность в повторной репликации может возникнуть по многим причинам: узел может стать недоступным, реплика может стать поврежденной или коэффициент репликации файла должен быть увеличен.

Типичный размер блока, используемый HDFS, составляет 64 МБ. Таким образом, файл HDFS разделяется на фрагменты по 64 МБ каждый, и, если возможно, каждый фрагмент будет находиться на другом Datanode.

Когда файл удаляется пользователем или приложением, он не сразу удаляется из HDFS. Вместо этого HDFS сначала переименовывает его в файл в каталоге /trash. Файл может быть восстановлен быстро, пока он остается в этом каталоге. По истечении срока его жизни Namenode удаляет файл из пространства имен HDFS. Удаление файла приводит к освобождению блоков, связанных с файлом.

Hadoop MapReduce – это фреймворк для программирования распределённых вычислений в рамках модели MapReduce. Разработчику приложения необходимо реализовать базовый обработчик. Фреймворк позволяет создавать задания как с базовыми обработчиками, так и с обработчиками, написанными без использования Java. Также, в состав дистрибутивов Hadoop входят реализации различных конкретных базовых обработчиков и свёрток, наиболее типично используемых в распределённой обработке. Модуль Hadoop MapReduce реализован поверх YARN.

Оценка производительности – это количественная основа любого исследования компьютерных систем. Для того, чтобы набор эталонов был уместным, его рабочие нагрузки должны представлять собой важные приложения целевой системы и быть достаточно разнообразным, чтобы демонстрировать диапазон поведения целевых приложений.

Для HDFS и MapReduce существует набор тестов HiBench, разработанный Intel для тестирования систем с помощью стресс-тестов [3]. HiBench состоит из набора программ Hadoop, которые помогают оценить структуру Hadoop с точки зрения скорости, пропускной способности, использования системных ресурсов и шаблонов доступа к данным. Он состоит из 10 различных рабочих нагрузок: Micro Benchmarks (сортировка, подсчёт количества слов, TeraSort, улучшенный DFSIO), веб-поиск (индексирование Nutch, PageRank), машинное

обучение (байесовская классификация, кластеризация методом K-средних) и аналитические запросы (Hive Join, Hive Aggregation).

Для тестирования кластера из 3 узлов, которые представляют собой виртуальные машины с ОС Ubuntu, расположенные в одной сети, были выбраны: подсчёт количества слов, индексирование Nutch и кластеризация методом K-средних. В кластере один узел является главным, остальные два узла – рабочие. В таблице 2 представлена информация об объёмах данных для каждого тестирования.

Таблица 2 – Информация об эксперименте

Тест	Подсчёт количества слов	Индексирование Nutch	Кластеризация методом K-средних
Объём данных	60 Гб	8.4 Гб	66 Гб

На рисунках 2-4 представлены графики использования центрального процессора, памяти и дискового ввода-вывода на временной шкале для каждой рабочей нагрузки.

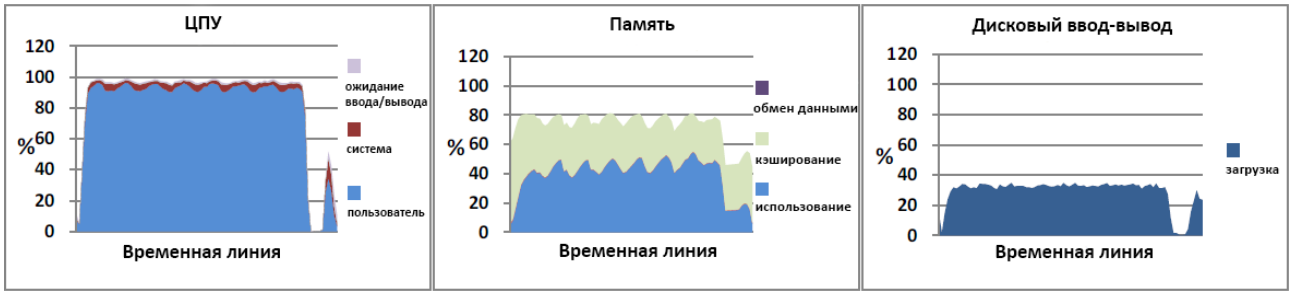

Рисунок 2 – Результаты для подсчёта количества слов

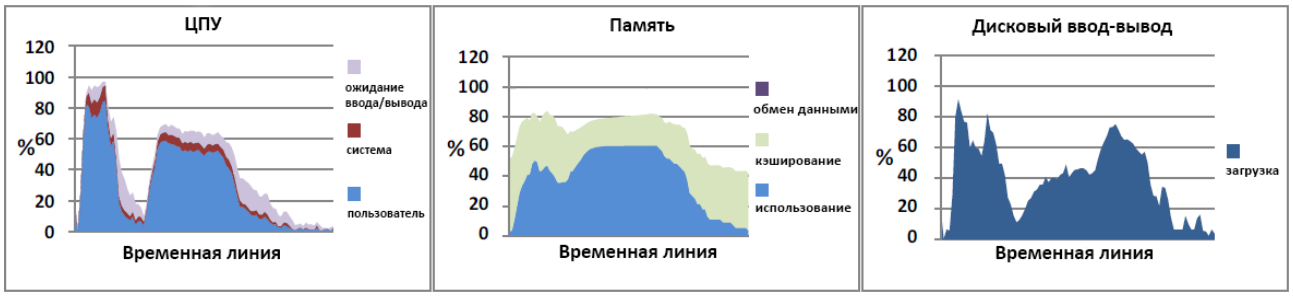

Рисунок 3 – Результаты для индексирования Nutch

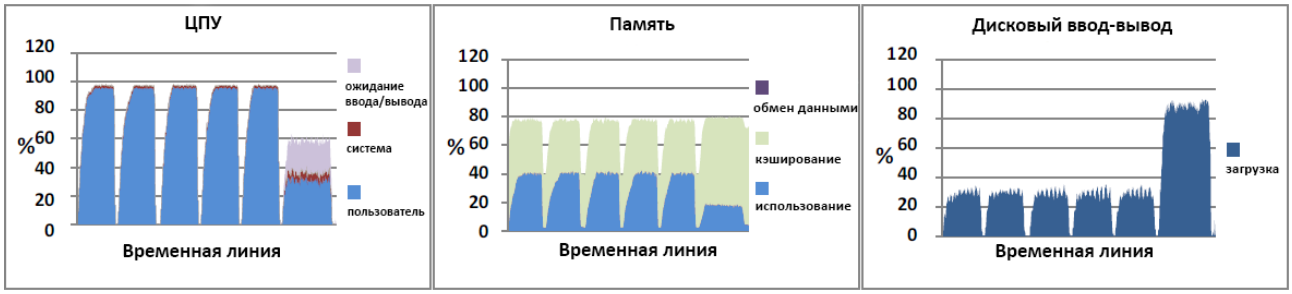

Рисунок 4 – Результаты для кластеризации методом K-средних

Рабочая нагрузка при подсчёте количества слов в основном связана с высокой загрузкой центрального процессора и незагруженным дисковым вводом-выводом, как показано на рисунке 2. Кроме того, ожидается, что поведение практически не изменится при тестировании на больших кластерах.

Как показано на рисунке 3, индексирование Nutch связано с высокой загрузкой процессора во время этапа Map, и с большим количеством операций ввода-вывода на диск (с примерно 60% использованием процессора).

Рабочая нагрузка кластеризации методом K-средних занимает большую часть времени на итерациях работы при вычислении центров кластеров. Как показано на рисунке 4, вычисление центроидов в методе K-средних в основном приводит к высокой нагрузке процессора. С другой стороны, задача кластеризации приводит к большому количеству операций ввода/вывода, в основном из-за записи результатов.

Таким образом, модель MapReduce и файловая система HDFS решают проблему хранения больших объёмов данных. Их можно эффективно применять для работы с данными, необходимыми для обучения и эксплуатации нейронных сетей.

Исследование выполнено при финансовой поддержке Правительства Оренбургской области и РФФИ (проекты №17-47-560046, №16-29-09639 и №18-07-01446), Президента Российской Федерации в рамках стипендии для молодых ученых и аспирантов (СП-2179.2015.5).

Список литературы

1. White, T. Hadoop: The definitive guide [Текст] // O'Reilly Media, Inc., 2012.- 756 с.

2. Borthakur, D. The hadoop distributed file system: Architecture and design [Текст] // Hadoop Project Website. – 2007. – Т. 11. – №. 2007. – С. 21.

3. Huang S. et al. Hibench: A representative and comprehensive hadoop benchmark suite [Текст] // Proc. ICDE Workshops. – 2010 – 325 с.

МЕТОДИКА ОЦЕНКИ КАЧЕСТВА ГИПЕРССЫЛОЧНЫХ УЧЕБНЫХ ПОСОБИЙ МОДУЛЬНОГО ТИПА

Ряполова Е.И. , канд. пед. наук, доцент
Оренбургский государственный университет

Системный подход является методологической базой разработки методик для оценки качества гиперссылочных учебных пособий, а также других программных средств используемых в процессе обучения студентов, как модели построения эффективного обучения. Возможность его применения к педагогическим объектам основывается на системности как важном качестве объективной действительности [2].

Процесс обучения с использованием гиперссылочных учебные пособия эффективно функционирует, и развиваться в современных условиях информатизации, опираясь на творческую направленную деятельность студентов, саморазвитие и критическое мышление.

Эффективность процесса обучения студентов состоит в мотивационной, психолого-педагогической и методической подготовке студентов к самоуправлению своей деятельностью.

Разработанная модель модульного обучения с использованием гиперссылочного учебного пособия осуществлялась с позиции построения педагогической системы. Структура и содержание педагогической системы рассматриваются в работах: В.П.Беспалько, Ю.Г. Татура, Л.Я.Терещенко, А.М. Кутеповым и другими [1]. Авторы выделяют следующие элементы системы: цели подготовки специалиста, учащиеся, содержание подготовки, дидактические процессы как способы осуществления задач педагогического процесса, преподаватели и опосредующие их педагогическую деятельность технические средства обучения, организационные формы педагогической деятельности [1].

Взяв за основу данные положения и принципы модульности, разработана модель модульного обучения с использованием гиперссылочного учебного пособия (рисунок 1). Цель разработки модели выступает повышении эффективности подготовки будущих инженеров.

Данная модель модульного обучения с использованием гиперссылочного учебного пособия, для студента [2]:

-обеспечивает предоставление теоретического материала по данной дисциплине;

-способствует систематической самостоятельной подготовке студентов к лекционным, практическим и лабораторным занятиям;

-содержит необходимый учебно-методический материал для самостоятельного изучения дисциплины;

-обеспечивает обратную связь с преподавателем;

-осуществляет системный поэтапный контроль и самоконтроль студента.

Рисунок 1 – Модель модульного обучения с использованием гиперссылочного учебного пособия

Для преподавателя данная модель модульного обучения с использованием гиперссылочного учебного пособия:

-способствует эффективной организации аудиторной и самостоятельной работы студентов;

-включает средства автоматизированного контроля знаний студентов;

-предоставляет результаты студентов для дальнейшего анализа и корректировки;

-предоставляет возможность усовершенствования учебно-методического материла изучаемой дисциплины с учетом специфики преподаваемой дисциплины и ростом информационного прогресса;

-позволяет строить индивидуальный маршрут обучения, как для каждого студента, так и для группы в целом.

В ходе анализа литературы нами выделен состав системы диагностики обучающих программ (рисунок 2).

Название системы	Подсистемы	Элементы
Диагностика качества гиперссылочных учебных пособий	Обучающая программа Учебное пособие	Теоретический блок Практический блок Контрольный блок Справочный блок
	Исследователь	Преподаватель Разработчик обучающей программы
	Обучаемые	Студенты Учащиеся
	Блок диагностики качества учебных пособий (обучающих программ)	Оценочный блок Методики оценки Блок результатов оценки

Рисунок 2 - Модель состава системы для диагностики качества обучающих программ

Модель системы для диагностики качества обучающих программ состоит из четырех элементов взаимосвязанных между собой (рисунок 3).

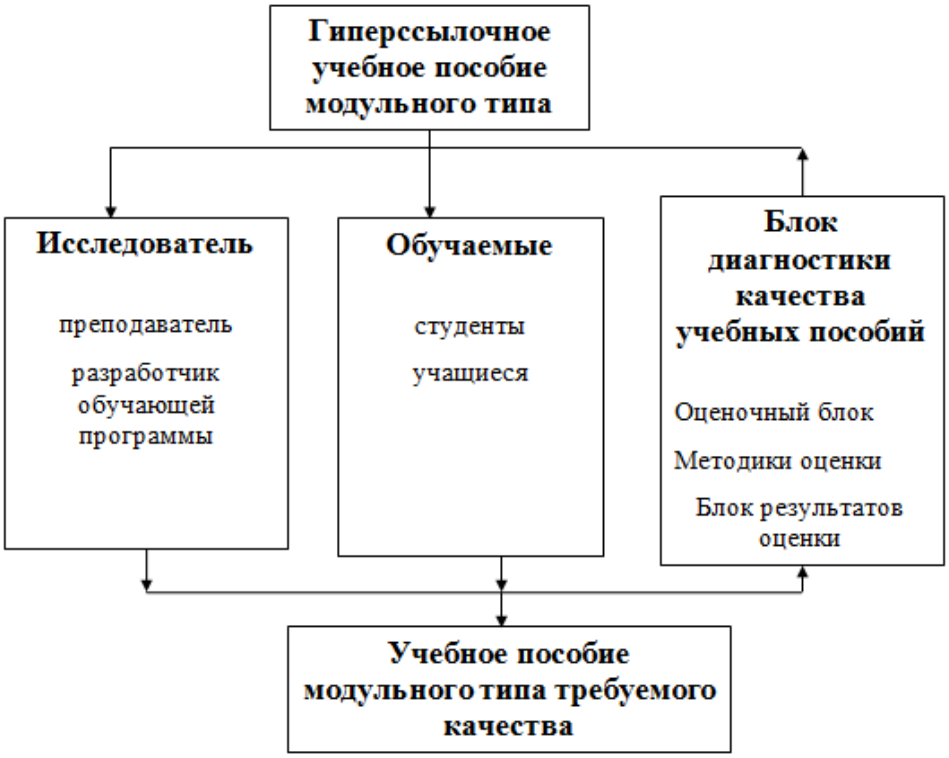

Рисунок 3 – Структурная схема системы диагностики гиперссылочных учебных пособий

В литературе существуют различные определения показателей качества разработки гиперссылочных учебных пособий и единого подхода не прослеживается. Описываются различные формы, методы, модели построения учебного процесса обладающие определенной эффективностью. Для проведения целостной диагностики гиперссылочных учебных пособий определим показатели их качества. Выделенные показатели встречаются во многих работах исследователей в данной области, но интерпретации показателей различны. Определим показатели по значимости: программно-техническая реализация гиперссылочного учебного пособия, методическая обоснованность, психолого-эргономические аспекты. Каждый из предложенных показателей представлен более подробно и определен математически.

Таким образом, в ходе анализа литературы предложена методика построения модульной программы с использованием гиперссылочного учебного пособия, а так же позволил нам выделить наиболее эффективную методику оценки качества гиперссылочных учебных пособий модульного типа. Анализ модульных программ позволил выделить его компоненты. Модульное формирование курса с использованием гиперссылочного пособия дает возможность осуществлять перераспределение времени, отводимого учебным планом на его изучение, более полно удовлетворить потребности творческой личности студента.

Список литературы

1.Беспалько, В.П. Системно-методическое обеспечение учебно-воспитательного процесса подготовки специалистов: Учебно-методическое пособие / В.П. Беспалько, Ю.Г, Татура.- М.: Высшая школа, 1989.-144с..

2. Ананьева, Е.И. Модульное обучение студентов как педагогическая проблема / Ананьева , Е.И. . - Вестник №4.-Оренбург: ГОУ ОГУ.-2006.-168с.

ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ ПРЕДСКАЗАТЕЛЬНОГО МОДЕЛИРОВАНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ УРОВНЯ ПОДГОТОВКИ УЧИТЕЛЯ ИНФОРМАТИКИ

Симченко Н. Н., канд. пед. наук, доцент
Оренбургский государственный университет

В современном обществе накоплено большое количество данных, которые подвергаются анализу. Кроме того, в огромной объеме информации, которую человек не в силах исследовать самостоятельно, содержатся знания. Для обнаружения «скрытых» знаний применяется специальные методы автоматического анализа, при помощи которых приходиться практически добывать знания из «завалов» информации. За этим направлением закрепился термин «добыча данных (Data Mining)» или «интеллектуальный анализ данных», который применяется для реализации масштабных аналитических проектов в бизнесе, маркетинге, интернете, телекоммуникациях, промышленности, геологии, медицине, фармацевтике и других областях.

Предсказательное моделирование основано на подходе Data Mining, что делает его наиболее полезным в ситуациях, когда:

- пользователь имеет дело с многомерной проблемой: есть множество факторов, оказывающих влияние на объект анализа;
- в данных имеются пропуски или неверно заполнены поля;
- не совсем понятно, подходят ли имеющиеся данные для анализа (первичная оценка данных);
- требуется быстрый наглядный результат, поскольку пользователь не владеет навыками настройки модели и ее интерпретации;
- решение нужно «день-в-день»;
- желательно проанализировать все имеющиеся данные (без лимита на число переменных).

Довольно часто смоделировать ту или иную ситуацию с использованием обычных вычислительных средств крайне сложно, например, если необходимо сопоставлять слишком много критериев или данных между собой. Тут на помощь и приходят методы компьютерного моделирования. Процесс моделирования начинается с «загрузки» в компьютер необходимых данных. На основе этих сведений создается и настраивается компьютерная модель – виртуальный образ ситуации. Затем в дело вступают эксперты, которые формулируют к образу всевозможные сценарии его развития. Это не значит, что они предсказывают саму ситуацию, – они лишь определяют условия, в которых ситуация будет предположительно развиваться.

Для того, чтобы подчеркнуть, что целью компьютерного моделирования является предсказание (прогноз, оценка) характеристик проектируемого объекта, в последнее время часто используется термин «предсказательное моделирование». Компьютерные системы предсказательного моделирования

(называемые также системами поддержки принятия инженерных решений) вместе с компьютерными системами проектирования давно используются для автоматизации труда инженера-проектировщика и повышения качества принимаемых решений. В рамках данного исследования была поставлена задача проанализировать зависимость уровня подготовки учителя от места проживания, категории, стажа работы и образования.

Для этого была спроектирована структура данных, основанная на результатах тестирования, которое было проведено для определения уровня предметной составляющей профессиональной компетенции учителя информатики. Для контроля знаний в тест были включены вопросы двух уровней сложности: низкий и высокий.

В качестве выходного определен атрибут –Уровень подготовки учителя, показывающий уровень подготовки учителя, принимающий значение «высокий», если учитель, верно ответил на вопросы высокого уровня сложности и значение «низкий», если учитель ответил верно на вопросы на распознавание, которые представлены тестами с выборочными ответами. Содержание тестовых вопросов отражало все содержательные линии школьного предмета «Информатика» в полном соответствии с ФГОС полного (среднего) образования. 60% заданий аналогичны заданиям части «В» Единого государственного экзамена по информатике, 40% заданий предусматривали проверку теоретических знаний учителя по темам, которые недостаточно полно представлены в ЕГЭ, например, компьютерным сетям. Для обширного контроля знаний в тест были включены вопросы двух уровней сложности: низкий и высокий.

Низкий уровень – вопросы на распознавание, которые представлены тестами с выборочными ответами. Они представлены заданиями с кратким ответом

Высокий уровень – вопросы на воспроизведение знаний и на применение при решении нетиповой или измененной задачи. Они представлены заданиями с кратким ответом (задания на вычисление определенной величины; задания на установление правильной последовательности, представленной в виде строки символов по определенному алгоритму) и в виде заданий с открытым (развернутым) вариантом ответа.

Для анализа зависимости уровня подготовки учителя от места проживания, категории, стажа работы и образования, были определены атрибуты:

1) Район/Город – место проживания учителя, возможные значения: областной центр, город, районный центр, село.

2) Категория – уровень квалификации педагога. Возможные значения: нет, вторая, первая, высшая.

3) Образование – уровень подготовки. Возможные значения: профильное (педагогическое высшее (учитель информатики), непрофильное-педагогическое высшее (учитель другой дисциплины, педагог-психолог и другое).

4) Стаж работы. Возможные значения6: >20, <5, 5-10, 11-20.

В качестве выходного определен атрибут –Уровень, показывающий уровень подготовки учителя, он будет принимать значение «высокий», если учитель, верно ответил на вопросы на воспроизведение знаний и на применение при решении нетиповой или измененной задачи и принимать значение «низкий», если учитель ответил верно на вопросы на распознавание, которые представлены тестами с выборочными ответами. Возможные значения: низкий, высокий.

Для построения дерева решений был рассмотрен следующий фрагмент хранилища данных, полученный экспериментально (таблица 1).

Таблица 2.1 – Хранилище данных

Образование	Район/Город	Категория	Стаж	Уровень
Профильное	Город	Высшая	>20	высокий
Профильное	Город	Высшая	>20	высокий
Профильное	Город	Высшая	<5	высокий
Профильное	Город	Первая	5-10	высокий
Профильное	Город	Нет	5-10	высокий
Непрофильное	Город	Первая	<5	низкий
Профильное	Город	Первая	11-20	высокий
Непрофильное	Город	Первая	<5	низкий
Профильное	Город	Первая	5-10	высокий
Непрофильное	Город	Первая	<5	низкий
Непрофильное	Город	Нет	<5	низкий
Непрофильное	Город	Нет	<5	низкий
Непрофильное	Город	Вторая	<5	низкий
Профильное	Областной_центр	Высшая	>20	высокий
Непрофильное	Областной_центр	Первая	<5	низкий
Непрофильное	Районный_центр	Нет	<5	низкий
Профильное	Районный_центр	Первая	5-10	низкий
Непрофильное	Районный_центр	Нет	<5	низкий
Профильное	Село	Первая	5-10	высокий

Для проведения предсказательного моделирования был выбран алгоритм ID3. Алгоритм строит такое решающее дерево, в котором с каждым узлом ассоциирован атрибут, являющийся наиболее информативным среди всех атрибутов, еще не рассмотренных на пути от корня дерева. В качестве меры информативности обычно используется теоретико-информационное понятие энтропии. Для того, чтобы получить более оптимальное дерево принятия решений, нужно на каждом шаге выбирать атрибуты, которые «лучше всего» характеризуют целевую функцию [1].

Предположим, что имеется множество A из n элементов, m из которых обладают некоторым свойством S. Тогда энтропия множества A по отношению к свойству S вычисляется по формуле (1):

$$H(A,S) = -\frac{m}{n} log_2 \frac{m}{n} - \frac{n-m}{n} log_2 \frac{n-m}{n} \quad (1)$$

То есть энтропия зависит от пропорции, в которой разделяется множество. По мере возрастания этой пропорции от 0 до ½ энтропия тоже возрастает, а после ½ – симметрично убывает.

Если свойство S – не бинарное, а может принимать s различных значений, каждое из которых реализуется в m_i случаях, то энтропия обобщается естественным образом, формула (2):

$$H(A,S) = -\sum_{i=1}^{s} \frac{m_i}{n} \log \frac{m_i}{n} \quad (2)$$

Понятие энтропии тесно связано с теорией информации. Грубо говоря, энтропия – это среднее количество битов, которые требуются, чтобы закодировать атрибут S у элемента множества A. Если вероятность появления S равна ½, то энтропия равна 1, и нужен полноценный бит; а если S появляется не равновероятно, то можно закодировать последовательность элементов множества A более эффективно.

При выборе атрибута для классификации нужно выбрать его так, чтобы после классификации энтропия стала как можно меньше (свойство S в данном случае – значение целевой булевской функции). Энтропия при этом будет разной в разных потомках, и общую сумму нужно считать с учетом того, сколько исходов осталось в рассмотрении в каждом из потомков. Общепринятое в теории деревьев принятия решений определение выглядит следующим образом:

Предположим, что множество A элементов, некоторые из которых обладают свойством S, классифицировано посредством атрибута Q, имеющего q возможных значений. Тогда прирост информации (information gain) определяется формулой (3).

$$Gain(A,Q) = H(A,S) - \sum_{i=1}^{q} \frac{|A_i|}{|A|} H(A_i,S) \quad (3)$$

где A_i – подмножество элементов множества A, на которых атрибут Q имеет значение i.

Практическое применение классической реализации ID3 сталкивается с рядом проблем, характерных для моделей, основанных на обучении вообще и деревьев решений в частности. Основными из них являются переобучение и наличие пропусков в данных.

Например, если данные содержат шум, то число уникальных значений атрибутов увеличится, а в крайнем случае для каждого примера обучающего множества значения атрибутов окажутся уникальными.

Следуя логике ID3, можно предположить, что при разбиении по такому атрибуту будет создано количество узлов, равное числу примеров, так как в каждом узле окажется по одному примеру. После этого каждый узел будет объявлен листом, и дерево даст число правил, равное числу примеров обучающего набора.

Энтропия при этом будет разной в разных потомках, и общую сумму нужно считать с учетом того, сколько исходов осталось в рассмотрении в каждом из потомков. Кроме энтропии применяется величина *Gain,* показывающая количество информации, которое получают благодаря некоторому атрибуту. Алгоритм ID3 использует эту величину для оценки информативности атрибута при построении решающих деревьев, что позволяет получать деревья минимальной высоты.

Далее была произведена программная реализация алгоритма ID3, выбран язык программирования C#, среда разработки Microsoft Visual Studio 2012. Для построения дерева принятия решений разработан рекурсивный статический метод create_tree.

```
class item_tree
{
   public attributes a;
   public int num_attr;
   public int obuch = 0;
   public string res;
   public item_tree[] children;
   public bool[] enter=new bool[count];
}
```

На рисунке 1 продемонстрирована работа программы: выбор атрибута «Образование» в качестве корневого и перебор ветвей полученного узла.

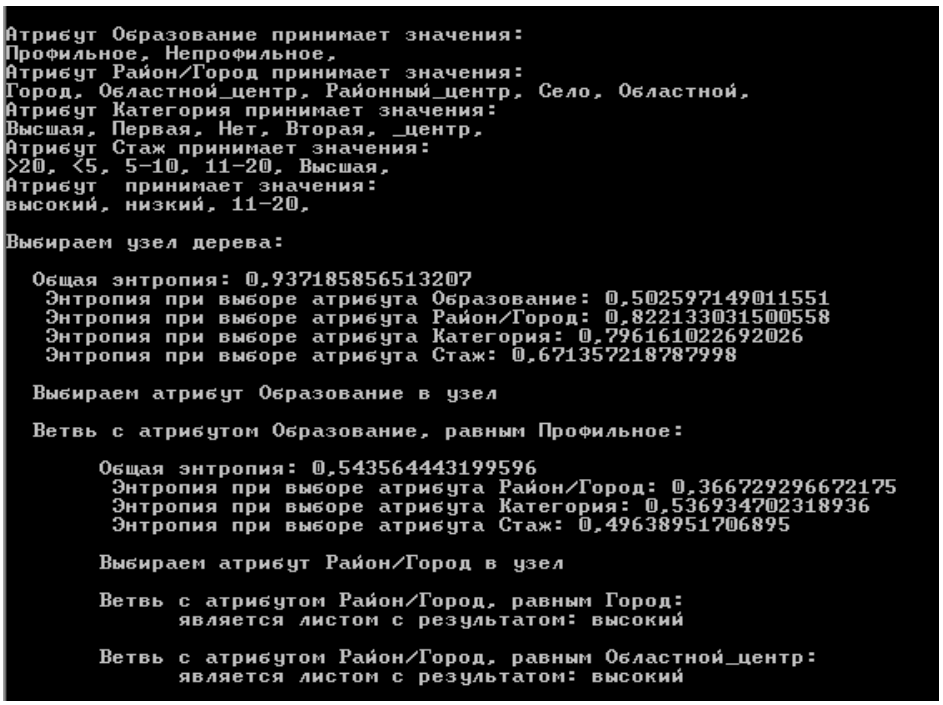

Рисунок 1 – Выбор атрибута, наиболее уменьшающего энтропию

По окончании построения дерева принятия решений, можно спрогнозировать уровень, определив район/город, категория, стаж. Результаты такого прогноза представлены на рисунке 2.

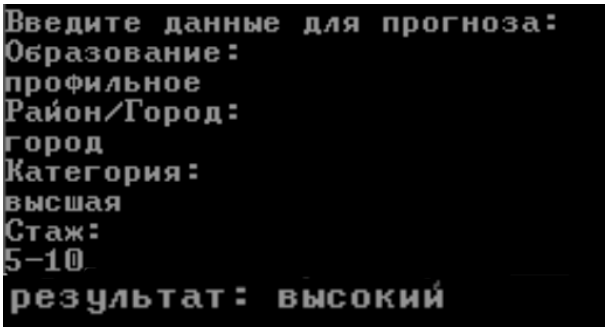

Рисунок 2– Результат прогноза

В результате работы программы на приведенном в таблице 1 фрагменте хранилища данных, было построено дерево принятия решений. Анализ построенного дерева принятия решений, показывает, что на уровень подготовки учителя наибольшее влияние оказывает образование (профильное/непрофильное). Проанализировав ветвь «Профильное», было выявлено, что наиболее информативным является атрибут «Стаж», если категория «Высшая», то «Уровень» – «Высокий», если «Вторая», то «Низкий».

Таким образом был сделан вывод, что для того, чтобы уровень подготовки учителя был высоким необходимы средства повышения квалификации, т.к. меняющаяся ситуация в системе общего образования

формирует новые образовательные потребности педагогов. Учитель постоянно находится между практикой и теорией, наращивая свой опыт преимущественно практическими умениями. Любая педагогическая работа – это практическая деятельность. Часто бывает так, что между теоретическими знаниями и практическими умениями продолжает сохраняться серьёзный разрыв. Преодолеть этот разрыв в современной школе можно средствами профессиональной переподготовки. Человека с современным мышлением, способного успешно самореализоваться в жизни, могут только педагоги, обладающие высоким профессионализмом. Повышение квалификации помогает учителю избавиться от устаревших взглядов, делает его более восприимчивым к внешним изменениям, что в конечном итоге повышает его конкурентоспособность.

Список литературы

1. *Вагин, В. Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин., Е. Ю. Головина, А. А. Загорянская, М. В. Фомина // Москва.: ФИЗМАТЛИТ, 2004. – 704 с. -*
2. *Вьюгин, В.В. Математические основы теории машинного обучения и прогнозирования / В. В. Вьюгин. – Москва: МЦНМО, 2013. — 390 с.*
3. *Осипов, Г. С. , Методы искусственного интеллекта / Г .С. Осипов.– Москва: Физматлит, 2011. - 296 с.*

ИССЛЕДОВАТЕЛЬСКАЯ ДЕЯТЕЛЬНОСТЬ В КОМПЕТЕНТНОСТНОЙ СТРУКТУРЕ НАПРАВЛЕНИЯ ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

Тарасова Т.Н., канд. пед. наук, доцент
Оренбургский государственный университет

Научно-исследовательская деятельность является одним из основных видов профессиональной деятельности, к которым готовятся выпускники, освоившие программы академического бакалавриата и академической магистратуры направления 01.03.02 Прикладная математика и информатика [1], [2]. Перечень профессиональных задач и профессиональных компетенций, соответствующих научно-исследовательской профессиональной деятельности, предусмотренных федеральными государственными образовательными стандартами, формируют систему требований к результатам образовательного процесса и основные направления его организации в подсистеме, призванной обеспечить формирование исследовательской компетентности выпускников.

ФГОС высшего образования по направлению 01.04.02 Прикладная математика и информатика (уровень магистратуры)	ФГОС высшего образования по направлению 01.03.02 Прикладная математика и информатика (уровень бакалавриата)
Профессиональные задачи, относящиеся к научно-исследовательскому виду профессиональной деятельности	
изучение новых научных результатов, научной литературы или научно-исследовательских проектов в области прикладной математики и информатики в соответствии с тематикой проводимых исследований;	изучение новых научных результатов, научной литературы или научно-исследовательских проектов в соответствии с профилем объекта профессиональной деятельности;
исследование систем методами математического прогнозирования и системного анализа	изучение информационных систем методами математического прогнозирования и системного анализа;
разработка и применение современных высокопроизводительных вычислительных технологий, применение современных суперкомпьютеров в проводимых исследованиях;	изучение больших систем современными методами высокопроизводительных вычислительных технологий, применение современных суперкомпьютеров в проводимых исследованиях;
построение математических моделей и исследование их аналитическими методами, разработка алгоритмов, методов, программного обеспечения,	исследование и разработка математических моделей, алгоритмов, методов, программного обеспечения, инструментальных средств по тематике про-

инструментальных средств по тематике проводимых научно-исследовательских проектов;	водимых научно-исследовательских проектов;
составление научных обзоров, рефератов и библиографии по тематике проводимых исследований;	составление научных обзоров, рефератов и библиографии по тематике проводимых исследований;
	участие в работе научных семинаров, научно-тематических конференций, симпозиумов;
подготовка научных и научно-технических публикаций.	подготовка научных и научно-технических публикаций.
Профессиональные компетенции, соответствующие научно-исследовательскому виду профессиональной деятельности	
ПК -1 способность проводить научные исследования и получать новые научные и прикладные результаты самостоятельно и в составе научного коллектива	ПК -1 способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям
ПК – 2 способность разрабатывать и анализировать концептуальные и теоретические модели решаемых научных проблем и задач	ПК -2 способность понимать, совершенствовать и применять современный математический аппарат
	ПК – 3 способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности

Структура требований ФГОС рассматривает исследовательскую компетентность как составную часть профессиональной компетентности, призванную обеспечивает ее эффективность. В российской системе образования, традиционно, в перечне ключевых компонентов исследовательской компетенции, выделяют четыре группы:

- когнитивный компонент, который рассматривается как совокупность знаний и понятий, необходимых для постановки и решения исследовательских задач в сфере профессиональной деятельности;

- мотивационный компонент, включающий смыслы, которые исследовательская деятельность имеет для конкретного человека;

- ориентировочный компонент, понимаемый как совокупность умений, обеспечивающих выявление потребности в каких-то знаниях и построение образа того, как оно может быть получено в существующих условиях;

- операционный или технологический компонент – это совокупность умений субъекта выполнять исследовательские действия, необходимые для решения исследовательских задач [3].

Таким образом, образовательная программа направления должна обеспечить, наряду с когнитивным и технологическим компонентами, реализующимися, в основном, через содержание образования, мотивационный и ориентировочный, для реализации которых необходимы специальные формы организации образовательного процесса.

При рассмотрении активной научно-исследовательской деятельности как организованного управляемого процесса, предоставляющего студентам возможность с помощью доступных им на определенном этапе обучения знаний, умений и навыков участвовать в создании (моделировать), анализировать и преобразовывать (видоизменять на фоне изменяющихся профессиональных задач) объекты профессиональной деятельности, проявляя при этом активность, способность самостоятельно принимать решения и нести за них ответственность, критично оценивая результаты своих действий [4], актуализируется проблема поиска эффективных подходов, которые бы позволили включить студентов в научно-исследовательскую деятельность, как одна из основных проблем развития исследовательских компетенций.

Очевидно, что формирование исследовательской компетенции возможно лишь при органическом соединении учебного процесса с научно - исследовательской деятельностью. В российской системе высшего образования различают два вида исследовательской деятельности студентов: учебно-исследовательскую (УИРС) и научно-исследовательскую (НИРС), которые образуют систему усложняющихся моделей объектов профессиональной деятельности. При выполнении учебных исследований формируются элементарные исследовательские умения и навыки. В отличие от учебно-исследовательской работы, научные исследования предполагают более высокий уровень изысканий, претендующий на объективную общественную значимость предполагаемых и полученных результатов. Тем не менее, несмотря на различия в формах организации и уровне исследовательской значимости, УИРС И НИРС совместно обеспечивают формирование исследовательской компетентности обучающихся. Кроме того, эти формы обеспечивают взаимосвязь обучения и научного исследования: процесс формирования исследовательских компетенций с содержанием исследовательской деятельности. В этом единстве реализуются системная мотивация учения и научно-исследовательской деятельности, творческий характер образовательного процесса в вузе, научное сотрудничество преподавателей и студентов [5].

УИРС организуется на основе учебного плана направления. При этом особого внимания заслуживает введение студента-первокурсника в такой вид деятельности. В формировании исследовательской компетенции важная роль отводится специально организованным средствам подготовки студентов к научно-исследовательской деятельности.

В образовательной программе, разработанной и реализуемой кафедрой прикладной математики Оренбургского государственного университета, первое знакомство студентов с теорией и практикой исследовательской деятельности, происходит в рамках учебной дисциплины «Введение в специальность». Один из разделов курса посвящен введению в научно-исследовательскую работу, которая рассматривается как способ повышения квалификации и ускорения карьерного роста. Содержание раздела представляет собой знакомство с инструментами и методами, применяемыми современным математиком-прикладником, научный характер его профессиональной деятельности; влияние научного образования математика-прикладника на возможность получения работы и последующую карьеру. В разделе рассматриваются основные способы получения научной квалификации, целесообразность и возможность продолжения образования в магистратуре, аспирантуре и докторантуре. Студенты узнают об основных видах исследовательских работ, выполняемых студентом в вузе, о научных, практических и коммерческих результатах исследовательской работы студентов, о возможности получения и опубликования научного результата при выполнении курсовой, дипломной и других студенческих научных исследований. Кроме того, узнают об организационных формах студенческой исследовательской деятельности: студенческом научном обществе университета, профессиональных олимпиадах и конкурсах, научных конференциях и др. В рамках курса предусмотрена организация встреч и бесед с руководителями исследовательских проектов: заведующей кафедрой прикладной математики д.т.н., профессором И.П. Болодуриной, заведующим кафедрой геометрии и компьютерных наук факультета математики и компьютерных наук ОГУ, к.п.н., доцентом Шухманом А.Е., членами профессорско-преподавательского состава кафедры, ведущими активную исследовательскую работу в области прикладной математики и информатики, а также выпускниками, магистрантами, аспирантами и студентами старших курсов направления Прикладная математика и информатика.

Кроме того, первокурсники привлекаются к участию в работе секций традиционной студенческой научной конференции Оренбургского государственного университета, студенческого научного семинара кафедры прикладной математики и информатики, где получают опыт обсуждения проблем, формулирования вопросов, анализа представленных материалов и их презентации. Таким образом, осуществляется знакомство с реальной НИР студентов. Привлечение студентов 1-го и 2-го курсов к работе исследовательских объединений, способствует решению целого спектра образовательных задач: осознанию значимости фундаментальных математических дисциплин для будущей профессиональной деятельности, их приложений к решению конкретных задач прикладной математики и информатики, повышению мотивации к обучению в целом и к занятию исследовательской работой, в частности.

В дальнейшем за период обучения каждый студент самостоятельно выполняет ряд различных работ исследовательского характера: доклады, рефера-

ты, курсовые и выпускные квалификационные работы. Все эти специально организуемые виды исследовательских работ, относящиеся к УИРС, обеспечивают овладение студентами современными методами поиска, обработки и использования информации, освоение методов научно-исследовательской деятельности, формирование исследовательской компетенции.

Организация НИРС является одним из направлений деятельности кафедры, которое реализуется через привлечение студентов к выполнению госбюджетных исследований по отдельным разделам, к участию в хоздоговорных НИР, в научно-практических конференциях, в научных конкурсах и олимпиадах различных уровней. При этом используются различные формы: индивидуальная (публикации тезисов докладов и статей), групповая (научные семинары, защита курсовых работ и т. д.) и коллективная (научно-практические конференции, «круглые столы» и т. д.).

В частности, кафедрой прикладной математики организуется работа нескольких секций в рамках ежегодной Студенческой научной конференции ОГУ: «Управление и моделирование в социально-экономических и физических системах», «Приложения математики к решению естественнонаучных и социально-экономических задач», «Математика в естествознании и инженерных дисциплинах», «История математики, естествознания и техники» и др.

На старших курсах бакалавриата и в магистерских образовательных программах происходит слияние УИР и НИР, в исследовательской деятельности студентов усиливается и становится превалирующим прикладной характер задач. Исследовательские компетенции переходят с уровня общих исследовательских компетенций на уровень исследовательских компетенций профессионального характера. В условиях этого этапа обучения промежуточные и конечные результаты научно-исследовательской работы студентов становятся предметом обсуждения на заседаниях студенческого научного семинара кафедры.

Среди средств развития исследовательских компетенций в магистерской образовательной программе одним из наиболее эффективных является научно-исследовательская работа, организуемая как тип производственной практики, предполагающая последовательное прохождение этапов самостоятельного научного исследования. Важной частью организации такой практики является разработка системы оценочных средств и их критериев, позволяющих определить уровень сформированности исследовательских компетенций магистранта.

Важнейшей формой НИРС на протяжении всего периода обучения является активное взаимодействие научных руководителей со студентами, реализуемое не только посредством индивидуального и группового консультирования, но и входе совместной исследовательской работы в составе исследовательских коллективов. Студентка Ахмайзянова Ю. являлась исполнителем НИР «Совершенствование технологий интеграции и обработки данных при управлении университетским комплексом на основе распределенной информационной системы» (руководитель И.П. Болодурина), профинансированной Российским фондом фундаментальных исследований; магистрант Акманова Ю. явля-

ется одним из исполнителей ГБ НИР кафедры: «Информационно-математическое обеспечение реализации системного подхода в исследованиях проблем образования» (руководитель Н.В. Кулиш).

Реализуемая на кафедре прикладной математики факультета математики и информационных технологий ОГУ система организации исследовательской работы обеспечивает эффективную научно-исследовательскую деятельность студентов, единство образовательного, научного и инновационного процессов, условия формирования и развития исследовательской компетентности, способствует всестороннему развитию личности студентов, их творческих способностей.

Список литературы

1. Приказ министерства образования и науки Российской Федерации от 12.03.2015 N 228 «Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика (уровень бакалавриата)». Зарегистрировано в Минюсте России 14 апреля 2015 г. N 36844

2. Приказ министерства образования и науки Российской Федерации от 28.08.2015 N 911 «Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.04.02 Прикладная математика и информатика (уровень магистратуры)». Зарегистрировано в Минюсте России 23 сентября 2015 г. N 38968

3. Хуторской, А.В. Определение общепредметного содержания и ключевых компетенций как характеристика нового подхода к конструированию образовательных стандартов. // Доклад на отделении философии образования и теории педагогики РАО 23 апреля 2002. Центр «Эйдос». http://eidos.ru/journal/2002/0423.htm

4. Янюк, И.А. Формирование исследовательской компетенции студентов технических вузов. – Автореферат дисс. … кандидата педаг. наук – Шуя, 2010

5. Червова, А.А. Формирование исследовательских умений студентов вузов / А.А. Червова, И.А. Янюк // Наука и школа. – 2007. – № 6. – С. 11-14.

О МАТЕМАТИЧЕСКОЙ ПОДГОТОВКЕ БУДУЩИХ БАКАЛАВРОВ ПИЩЕВЫХ ПРОИЗВОДСТВ

Теплякова Г.В., канд. пед. наук, Казакова О.Н., канд. пед. наук
Оренбургский государственный университет

Реализация стандартов третьего поколения системы высшего образования современной России создает определенные предпосылки формирования компетентных работников пищевой индустрии. Ее доминанту составляют формирование знаний о способах рационального использования сырьевых, энергетических и других видов ресурсов, создание и обеспечение надежности технологий производства продуктов питания с высокими потребительскими свойствами.

Овладение соответствующими видами деятельности связывается с профессиональной мотивацией будущих бакалавров, обеспеченной на первоначальном этапе профессиональной подготовки изучением математики, закладывающей фундамент освоения компетенций и формирующей способность решать специфические отраслевые задачи [6].

Математическая подготовка в ее современном качестве оснащает будущего бакалавра аналитическими, численными методами и технологиями решения конкретных профессиональных задач. Вместе с тем в профессиональном образовании будущих бакалавров пищевых производств отсутствует целенаправленная интеграция математики и дисциплин профессионального цикла, учитывающая ее возможности в формировании профессиональной мотивации бакалавров.

Отметим, что уровень квалификационных характеристик, соответствующих уровню бакалавра технологии продукции и организации общественного питания, требует от выпускников углубленных знаний в профессиональной области, сформированности соответствующих профессиональных компетенций, необходимых для решения сложных задач в профессиональной сфере, а также наличие способности к инновациям. У будущих бакалавров должны быть сформированы навыки математического моделирования, т.е. навыки решения прикладных, инженерно-практических задач математическими методами. Данные требования зафиксированы в соответствующих общепрофессиональных компетенциях будущего бакалавра по направлению подготовки: 19.03.04 - Технология продукции и организация общественного питания:

– способность осуществлять технологический контроль соответствия качества производимой продукции и услуг установленным нормам (ОПК-3);

– способность измерять и составлять описание проводимых экспериментов, подготавливать данные для составления обзоров, отчетов и научных публикаций; владение статистическими методами и средствами обработки экспериментальных данных проведенных исследований (ПК-26) [7].

Анализ содержания программ и учебных пособий по общепрофессиональным и специальным дисциплинам показывает, что дисциплины, связанные

с будущей профессией бакалавра пищевого профиля, при изучении которых широко применяется математический аппарат, изучаются в основном на старших курсах. Однако подготовительная работа к ним для лучшего понимания и применения математического аппарата при изучении этих дисциплин должна осуществляться уже с первого курса, при изучении дисциплины «Математика».

Такая работа связана с определенными трудностями: во-первых, изучение математики приходится на первый-второй курсы, совпадающие со сложным периодом адаптации студентов к особенностям учебно-познавательной деятельности в высшей школе, включающим новые для них условия: социально-педагогические, психолого-педагогические, дидактические. В процессе адаптации у них формируются навыки и умения по рациональной организации умственной деятельности, позитивное отношение к избранной профессии и система профессионального самообразования и самовоспитания профессионально значимых качеств личности, рациональный коллективный и личный режим труда, досуга и быта. Второй причиной, как уже упоминалось выше, является то, что специальные дисциплины, на которых в полной мере можно увидеть возможности математики, изучаются, как правило, на средних и старших курсах, когда «основы» уже давно сданы и забыты. И опять же, зачастую преподаватели специальных дисциплин не уделяют должного внимания возможностям математического моделирования реальных процессов. При изучении математики, в качестве основного, особенно для студентов младших курсов, мы выбираем путь создания задачников, практикумов, ориентированных как на закрепление студентами основных вычислительных навыков, так и на формирование профессиональной компетентности будущих специалистов.

В частности, Л.В. Васяк в своем исследовании, посвященном процессу формирования профессиональной компетентности будущих инженеров в условиях интеграции математики и спецдисциплин, средствами профессионально - ориентированных задач, делает вывод о том, что положительную динамику сформированности профессиональной компетентности можно констатировать по следующим критериям: уровни овладения системой математических знаний, умений и навыков, уровни обучаемости, уровни обученности, сформированность мотивации изучения математики [1].

Овладения системой математических знаний, умений и навыков документально отражено в фонде оценочных средств (ФОС), который является приложением к рабочей программе по дисциплине «Математика» и является частью нормативно-методического обеспечения системы оценки качества освоения обучающимися образовательной программы высшего образования. В качестве оценочных средств используются *разно уровневые задачи и задания.*

Согласно положению о ФОС мы предъявляем студентам задачи и задания следующих уровней сложности:

– репродуктивного уровня, позволяющие оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов

изучения в рамках определенного раздела дисциплины (модуля);

– реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей;

– творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения [7].

По форме предъявления нами используются следующие виды оценочных средств:

– задания репродуктивного уровня (тестовые задания, несложные задания по выполнению конкретных действий; задания на установление правильной последовательности, взаимосвязанности действий; вопросы, направленные на выявление знаний основных математических понятий, формулировок и формул;

– задания реконструктивного уровня (комплексные практические контрольные задания – требующие многоходовых решений как в типичной, так и в нестандартной ситуациях и сопровождающиеся развернутым ответом. (Применяются для оценки уровня освоения компетенции – «владеть»);

– задания творческого уровня – частично регламентированные задания, имеющие нестандартное решение и позволяющие оценивать умение, интегрировать знания из различных областей, математически аргументировать собственную точку зрения. Могут выполняться в индивидуальном порядке или группой обучающихся. (Применяются для оценки уровня освоения компетенции – «владеть») [7].

Еще раз отметим, что в рабочих программах и ФОС нами отражено формирование компетенций в предметной области «Математика». Профессионально-ориентированные задания нами рассматривались в ходе проведения практических занятий по дисциплине.

В качестве примера приведем задачу, которую можно решать со студентами после изучения блока «Линейная алгебра и аналитическая геометрия».

Условие задачи: Фирма выпускает два вида мороженого: сливочное и шоколадное. Для приготовления мороженого используются два продукта: молоко и наполнители, расходы которых на 1 кг мороженого и суточные запасы даны в таблице:

Исходный продукт	Расход на 1 кг мороженого		Запас кг
	Сливочное	Шоколадное	
Молоко	0,8	0,5	400
Наполнители	0,4	0,8	365

Изучение рынка сбыта показало, что суточный спрос на сливочное мороженое превышает спрос на шоколадное не более чем на 100 кг, спрос на шоколадное не превышает 350 кг в сутки. Отпускная цена 1 кг сливочного мороженого – 16 ед., шоколадного – 14 ед. Определить количество сливочного и шоколадного мороженного, которое должна производить фирма, чтобы доход от реализации продукции был максимальным.

Задание 1 (репродуктивный уровень).

Используя условие задания, составьте неравенства, описывающие ограничения по молоку, наполнителям и спросу. Составьте выражение для суточного объема выпуска сливочного и шоколадного мороженого. Найдите целевую функцию.

Задание 2 (продуктивный уровень).

Составьте математическую модель задачи. Постройте область допустимых решений, используя графо-аналитический метод для решения составленной модели

Задание 3 (творческий уровень).

Требуется среди всех неотрицательных решений системы ограничений найти такое, при котором целевая функция принимает наибольшее значение (максимизируется).

При выполнении таких комплексных заданий учебная деятельность будущих бакалавров приобретает целенаправленный, осознанный характер, а ее организация характеризуется постановкой цели, проектированием задач собственной учебной деятельности и способов решения. Использование профессионально-ориентированных задач способствует повышению рабочего тонуса при изучении математики, а также помогает обучающимся осознать свои способности, поверить в себя, в свои силы и возможности, что способствует формированию профессиональной мотивации.

Список литературы

1. *Васяк, Л.В. Формирование профессиональной компетентности будущих инженеров в условиях интеграции математики и спецдисциплин средствами профессионально ориентированных задач: автореф. дис. ... канд. пед. наук: 13.00.08 / Л.В. Васяк. - Чита, 2007. – 22 с.*

2. *Журбенко, Л.Н. Управление многопрофильной математической подготовкой студентов технологического университета [Электронный ресурс] / Л.Н. Журбенко, С.Н. Нуриева // Educational Technology & Society. – 2007. – 10(3). – Режим доступа:* http://ifets.ieee.org/russian/depository/v10_i3/html/11_Zhurbenko.htm . - *15.03.2014*

3. *Казакова, О. Н. Организация самостоятельной работы студентов по линейной алгебре и аналитической геометрии как условие формирования профессиональных компетенций будущего экономиста [Электронный ресурс] / О. Н. Казакова, О. Н. Конюченко // Университетский комплекс как региональ-*

ный центр образования, науки и культуры. Материалы Всероссийской научно-методической конференции, 1-3 февраля 2012 г. / Оренбургский гос. ун-т. - Оренбург: ООО ИПК «Университет», 2012. - [С. 1117-1125]. - 1 электрон. опт. диск (CD-ROM). - Загл. с этикетки диска. - ISBN 978-5-4418-0022-8. - № гос. регистрации 0321200663.

4. Казакова, О. Н. Взаимодействие субъектов образовательного процесса как фактор адаптации студентов первого курса к условиям обучения в вузе: монография / О. Н. Казакова. - Оренбург, 2010. - 169 с. - ISBN 978-5-904401-02-3.

5. Теплякова, Г.В. Модель формирования профессиональной мотивации будущих бакалавров пищевого профиля в изучении математики / Г.В Теплякова // Вестник Оренбургского государственного университета. - 2013. - №2. - С. 238-242.

6. Теплякова, Г.В. Формирование профессиональной мотивации будущих бакалавров в изучении математики: монография / В.Г. Гладких, Г.В. Теплякова. – Оренбург: ОГУ, 2014. - 200 с. - ISBN 978-5-4417-0510-3.

7. Оренбургский государственный университет [Электронный ресурс]: 1999-2017, ОГУ ЦИТ. – Режим доступа: http://www.osu.ru. – 20.12.2017

ИСПОЛЬЗОВАНИЕ ИНТЕРАКТИВНЫХ ТЕХНОЛОГИЙ В ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ ПОДГОТОВКИ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ

Токарева М.А., к.т.н., доцент, Кулантаева И.А., к.п.н.
Оренбургский государственный университет

В XXI веке высшее образование выступает в качестве основополагающего компонента устойчивого развития человеческого сообщества, в котором важнейшее место отводится университетскому образованию. Новый тип экономики вызывает новые требования, предъявляемые к выпускникам вузов, среди которых все больший приоритет получают требования системноорганизованных интеллектуальных, коммуникативных, рефлексирующих, самоорганизующихся, моральных начал, позволяющих успешно организовывать деятельность в широком социальном, экономическом, культурном контекстах. Все четче вырисовывается вектор модернизации высшего профессионального образования в направлении концептуально новой парадигмы организации образовательного пространства: от школы памяти – парадигмы преподавания (передачи информации) к школе мышления и развития – парадигме научения (передаче компетентных знаний как потенциала действия). Формирующаяся в условиях модернизации российской высшей школы и в контексте Болонского процесса компетентностная парадигма ориентируется на компетенции и компетентности как ведущего критерия подготовленности современного выпускника высшей школы к нестабильным условиям труда и социальной жизни. Основной целью высшего образования становится подготовка выпускника не просто знающего, но умеющего распорядиться этим знанием, т.е. подготовка профессионала, обладающего критическим мышлением, способного среди множества решений выбирать оптимальное, аргументированно опровергая ложные; профессионала, готового к самообразованию, самоопределению, саморазвитию.

Ведущая ранее в деятельности педагога функция обучения трансформируется в задачу поддержки учения, а позиция обучаемого меняется от пассивного объекта научения, получателя готовой учебной информации, объекта обучающих воспитательных воздействий до субъекта познавательной, будущей профессиональной и социокультурной деятельности, активного субъекта учения, самостоятельно «добывающего» необходимую информацию и конструирующего необходимые для этого способы действий. Усилия преподавателя – посредника процесса учения фокусируются на создании среды, ориентированной на самостоятельность, интерактивность и продуктивность деятельности студентов, среды, обеспечивающей возможность формирования индивидуального образовательного опыта студента, продвигающегося по собственной образовательной траектории. Таким образом, инновационная составляющая образовательного процесса в новой парадигме высшего образования, которая базируется на оптимизации методов обучения, внедрении

в учебный процесс новых технологий обучения, повышающих интенсивность образовательного процесса, на активном использование информационных технологий, позволяющих студенту в удобное для него время осваивать учебный материал.

Особое значение в инновационной составляющей современного образовательного процесса имеет использование интерактивных технологий, бурное развитие и широкое внедрение которых регламентирует, в том числе, и ныне действующий Федеральный государственный образовательный стандарт высшего образования (ФГОС ВО). В частности, согласно ФГОС 3+ по направлению подготовки 44.03.01 Педагогическое образование, профиль «Информатика», 30% занятий следует проводить с помощью интерактивных технологий. Их применение создает предпосылки к интенсификации образовательного процесса, а также к разработке методик, ориентированных на развитие интеллекта, логики и внимательности обучаемого, на самостоятельное извлечение и представление знания, на продуцирование информации.

Внедрение интерактивных методов обучения является одним из направлений улучшения подготовки студентов в современном вузе и обязательное условие эффективной реализации компетентностного подхода. Формирование заявленных в ФГОС компетенций предполагает применение новых технологий и форм реализации учебной работы. Существенным подходом является обязательность перехода от информативных форм и методов обучения к активным формам и методам обучения. Также формирование подхода уровня знаний к действительному подходу и поиск возможностей соединения теоретических знаний учащихся с их практическими потребностями. Выбор современных образовательных технологий, а также активных и интерактивных методов обучения должен пересекаться с формируемыми компетенциями.

На кафедре информатики, которая является выпускающей бакалавров профиля «Информатика» направления подготовки 44.03.01 Педагогическое образование, разработано и внедрено в учебный процесс большое количество интерактивных средств обучения. Они позволяют создавать элементы интерактивной образовательной среды для организации эффективной самостоятельной работы, предоставлять новые дидактические возможности в контактной работе со студентами, использовать инновационную методику при прохождении студентами педагогической практики.

Рассматривая специфику дисциплин блока предметной подготовки, необходимо отметить, что в их подавляющем большинстве интегрируются математические и информационные составляющие. Они включают в себя достаточно большое количество формул, геометрических интерпретаций, табличных представлений алгоритмов, что подразумевает необходимость использования демонстрационных материалов для обеспечения наглядности. В частности, к таким дисциплинам относятся «Численные методы», «Исследование операций». Электронные лекции и презентации являются особенно простыми в разработке интерактивных методических материалов. Знание инструментальных сред подготовки электронных материалов, в частности, Jimdo, Adobe Authorware, Macromedia Dreamweaver, Microsoft Expression Web, NVU Kompozer, Microsoft

PowerPoint, дает возможность преподавателю лично разрабатывать наглядное представление изучаемого материала. Именно для этих целей выполнены разработки электронных курсов лекций, приведенных в списке литературы (позиции 1, 2, 3). Кроме того, электронный курс лекций по дисциплине «Исследование операций», рассчитанный для сопровождения лекционного материала в аудиторной работе, также размещен в LMS Moodle для использования в качестве опорного конспекта лекций для подготовки к промежуточному и итоговому контролю по данной дисциплине.

Для учителя информатики, компетентность которого во многом определяется способностью ориентироваться в стремительно меняющемся мире информационных технологий, умением определять, использование каких технологий в образовательном процессе будет более эффективно, делать правильный подбор программного обеспечения и, конечно же, самостоятельно его осваивать, особо актуальным является эффективная организация самостоятельной работы как средства самостоятельной учебной деятельности студентов вузов. Как показывает большой педагогический опыт, хорошим средством методической поддержки самостоятельной работы студентов являются электронные учебные пособия (ЭУП).

Использование ЭУП в образовательном процессе вуза позволяет глубоко изучить материал, ознакомиться более подробно с интересующими или трудными темами. Богатый и красочный иллюстративный материал в электронном пособии позволяет наглядно показать теоретическую информацию во всём её многообразии и комплексности. При использовании электронных учебных пособий происходит не только самовоспроизводящая деятельность студентов, но и абстрактно-логическая, что способствует лучшему осознанию и усвоению учебного материала [4]. Авторами разработаны и успешно применяются в учебном процессе ЭУП по дисциплинам «Численные методы», «Исследование операций», «Программирование» (позиции 5, 6, 7 в списке литературы). Главная страница учебного пособия «Программирование» показана на рисунке 1.

Рисунок 1 – Главная страница учебного пособия «Программирование»

Разработанные ЭУП содержат теоретический материал, снабженный большим количеством иллюстраций; задания для выполнения лабораторных работ с приведенными примерами выполнения; по каждому разделу имеется перечень контрольных вопросов, по которым студент может проверить усвояемость учебного материала. Также имеется итоговый тест по всему курсу, пройдя который студент может вернуться к недостаточно изученному теоретическому материалу. Кроме того, в пособии имеются тематические кроссворды, которые студенты разгадывают с интересом. Для того, чтобы обучающемуся было легче с определениями в ЭУП, создан раздел "Глоссарий". Студент в любое время может обратиться к данному разделу, чтобы уточнить тот или иной термин. Глоссарий содержит термины на русском и английском языках.

ЭУП может быть использовано как в качестве материалов для самостоятельной работы, так и в аудиторных занятиях. ЭУП содержит большое количество иллюстрированных примеров, что обеспечивает наглядность представленного материала, исключает поиск необходимой информации в объемных литературных источниках, дает возможность проверить свои знания посредством тестирования, интерактивного кроссворда. Все эти реализованные возможности повышают мотивацию к изучению дисциплины.

Исходя из анализа формируемых компетенций по дисциплинам учебного плана по направлению 44.03.01 Педагогическое образование, профиль «Информатика», эффективной интерактивной технологией для изучения дисциплины «Базы данных и СУБД» является case - метод. На самостоятельную работу по всему курсу отводится достаточно большой объем часов. Содержательный аспект дисциплины позволяет применить кейс-технологию. Нами раз-

работан электронный учебно-методический комплекс (ЭУМК) «Базы данных и СУБД», предлагающий для изучения 8 тем. К каждой теме привязан теоретический блок и практический блок. В теоретическом блоке содержатся лекции, представляющие собой материал в виде текста и изображений. В практическом блоке представлены case - задания и материал для их выполнения. Разработаны тесты для самоконтроля с возможностью контроля результатов.

При работе с ЭУМК «Базы данных и СУБД», размещенного в LMS Moodle, были использованы следующие интерактивные организационно-педагогические формы обучения: электронные семинары, вебинары, индивидуальные и групповые проекты по технологии wiki, дискуссии и обсуждения, совместное составление электронного глоссария, тезауруса, списка аннотированных интернет–источников, ведение блогов, участие в сетевых профессиональных сообществах. Отметим, что на начальном этапе развития информационно-познавательной самостоятельности студенты были зарегистрированы в роли обучающихся в системе Moodle, выполняя все предложенные работы преподавателя. На заключительном этапе студенты выступали уже в роли помощника преподавателя, организовывая самостоятельно деятельность своих коллег (организовывали дискуссии, форумы, оценивали выполненные работы, выкладывали дополнительный материал по темам практических работ).

В процессе работы с применением case-технологий у обучающегося вырабатываются навыки самостоятельности, стремление к формированию навыков эффективного сочетания самостоятельной работы и работы в коллективе, происходит воспитание положительного интереса к предмету, формирование исследовательских навыков, умение самостоятельно планировать свою работу. Применение кейсов должно быть методически, информационно, организационно и педагогически обоснованным и обеспеченным. Бесспорно, функциональное поле кейсов открывает широкие возможности для использования, дополняет традиционные классические методы обучения. Использование кейсов в преподавании – это ещё один шаг к интеграции российской системы образования в мировое образовательное пространство.

Интерактивные образовательные технологии успешно применяются в учебном процессе школы студентами профиля «Информатика» при прохождении производственной педагогической и преддипломной практик. В рамках выполнения выпускной квалификационной работы студенты создают свои собственные разработки компьютерных средств обучения, предназначенные для методической поддержки разделов школьного предмета «Информатика и ИКТ»; разрабатывают модели уроков с применением этих средств, а также обосновывают их использование во внеклассной работе. При разработке модели урока обязательно учитывается Государственное санитарно-эпидемиологическое нормирование Российской Федерации – СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы». Пользовательский интер-

фейс разрабатывается с учетом возрастных особенностей учащихся. В качестве примера на рисунке 2 представлена главная страница ЭУМК по информатике для учащихся 5-6 классов.

Рисунок 2 – Главная страница ЭУМК по информатике для учащихся 5-6 классов

Выполнив задания, представленные в ЭУМК по каждой теме учебной дисциплины, у обучающихся появляется возможность проверить уровень усвоения полученных знаний. После выполнения всех заданий по определенной теме, на экране отображается результат (рисунок 3). Также обучающийся может просмотреть полный отчет о выполнении заданий, нажав на кнопку «Просмотреть задания», тем самым выявить свои ошибки и при необходимости повторить выполнение заданий. Использование на занятиях ЭУМК предполагает быстрое переключение учеников от одной формы работы к другой. Например, после изучения теоретического материала, учащимся предлагается ответить на вопросы или выполнить практические задания.

Рисунок 3 – Результат выполнения заданий

Наличие графических изображений и видеоматериалов в ЭУМК облегчает восприятие сложной информации. Как показывает практика, применение ЭУМК углубляет качество приобретаемых знаний, умений и навыков, так как наличие разных видов материалов (текст, графика, видео) в ЭУМК выстраиваются в сознании учеников в качестве наглядных образов, в итоге складываются в единую систему. Созданное и апробированное в процессе прохождения педагогической практики методическое обеспечение дает возможность совершенствовать процесс формирования учебной самостоятельной деятельности студентов и становления их самостоятельно-деятельностной компетентности.

При использовании интерактивных методов обучаемый становится полноправным участником процесса восприятия, его опыт служит основным источником учебного познания. Преподаватель не даёт готовых знаний, но побуждает обучаемых к самостоятельному поиску. По сравнению с традиционными формами ведения занятий, в интерактивном обучении меняется взаимодействие преподавателя и обучаемого: активность педагога уступает место активности обучаемых, а задачей педагога становится создание условий для их инициативы. Педагог отказывается от роли своеобразного фильтра, пропускающего через себя учебную информацию, и выполняет функцию помощника в работе, одного из источников информации. Интерактивное обучение обеспечивает взаимопонимание, взаимодействие, взаимообогащение. Интерактивные методики ни в коем случае не заменяют лекционный материал, но способствуют его лучшему усвоению и, что особенно важно, формируют мнения, отношения, навыки поведения; повышают мотивацию студентов к изучению дисциплины; развивают информационные и коммуникативные компетенции.

Список литературы

1 Токарева, М.А. Электронный курс лекций по дисциплине «Исследование операций» / М.А. Токарева, А.С. Карабанова.- Оренбург: УФЭР ОГУ, № 1415 от 07.06.2017.

2 Токарева, М.А. Электронный курс лекций по дисциплине «Численные методы» / М.А. Токарева, М.М. Пирязев. - Оренбург: УФЭР ОГУ, №1463 от 10.11.2017.

3 Токарева, М.А. Электронное гиперссылочное учебное пособие «Математическое программирование» / М.А. Токарева, М.М. Пирязев. - Оренбург: УФЭР ОГУ, №1462 от 10.11.2017 г.

4 Алешкина, О. В. Применение электронных учебников в образовательном процессе [Текст] / О. В. Алешкина // Молодой ученый. – 2012. – №11. – 391 с.

5 Токарева, М.А. Электронное учебное пособие «Численные методы» / М.А. Токарева, Т.А. Черных, К.Ш. Хамитов. - М.: ФГНУ ИНИПИ РАО, ОФЭРНиО - Свидетельство о регистрации электронного ресурса № 19020 от 19.03.2013

6 Токарева, М.А. Электронное учебное пособие по дисциплине «Исследование операций» / М.А. Токарева, Т.А. Черных. - М.: ФГНУ ИНИПИ РАО, ОФЭРНиО - Свидетельство о регистрации электронного ресурса № 19915 от 05.02.2014.

7 Токарева, М.А. Электронное учебное пособие по дисциплине «Программирование» / М.А. Токарева, Т.Е. Тлегенова. - Оренбург: УФЭР ОГУ, №1214 от 14.01.2016.

МАТЕМАТИЧЕСКАЯ КОМПЕТЕНТНОСТЬ АКАДЕМИЧЕСКОГО БАКАЛАВРИАТА НАПРАВЛЕНИЯ ПОДГОТОВКИ «МАТЕМАТИКА И КОМПЬЮТЕРНЫЕ НАУКИ» КАК ОСНОВОПОЛАГАЮЩАЯ ПРОФЕССИОНАЛЬНОЙ КОМПЕТЕНТНОСТИ

Усова Л.Б., канд. пед. наук,
Шакирова Д.У., канд. пед. наук
Оренбургский государственный университет

Национальная доктрина образования в Российской Федерации — основополагающий государственный документ, устанавливающий приоритет образования в государственной политике, стратегию и основные направления его развития. Доктрина определяет цели воспитания и обучения, пути их достижения посредством государственной политики в области образования, ожидаемые результаты развития системы образования на период до 2025 года.

Доктрина признает образование приоритетной сферой накопления знаний и формирования умений, создания максимально благоприятных условий для выявления и развития творческих способностей каждого гражданина России.

Система образования призвана обеспечить: систематическое обновление всех аспектов образования, отражающего изменения в сфере культуры, экономики, науки, техники и технологий; непрерывность образования в течение всей жизни человека и подготовку высокообразованных людей и высококвалифицированных специалистов, способных к профессиональному росту и профессиональной мобильности в условиях информатизации общества и развития новых наукоемких технологий [4].

Нормативную правовую базу разработки Образовательной программы Высшего Образования (ОП ВО) составляют Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» и Федеральный государственный образовательный стандарт высшего образования (ФГОС ВО) по направлению подготовки 02.03.01 – Математика и компьютерные науки (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от от «25» августа 2014 г. № 949. В перечень профессиональных стандартов, соответствующих профессиональной деятельности выпускников, освоивших программу бакалавриата по направлению подготовки 02.03.01 – Математика и компьютерные науки входит профессиональный стандарт «Программист», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013г. № 679н, с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. № 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный №45230)

Целью ОП ВО по направлению 02.03.01 «Математика и компьютерные науки» и профилю подготовки «Алгоритмы и приложения компьютерной ма-

тематики» является развитие у студентов личностных качеств, способствующих их творческой активности, общекультурному росту и социальной мобильности: целеустремленности, организованности, трудолюбия, ответственности, самостоятельности, гражданственности, приверженности этическим ценностям, толерантности, настойчивости в достижении цели, способности принимать организационные решения в стандартных и нестандартных ситуациях и готовность нести за них ответственность, умение критически оценивать собственные достоинства и недостатки, выбирать пути и средства развития первых и устранения последних, а также формирование общекультурных и профессиональных компетенций, позволяющих выпускнику успешно работать в избранной сфере деятельности и быть постоянно востребованным на рынке труда соответствующих предприятий, компаний научно-производственных объединений, учреждений науки и образования.

Задачами ОП ВО по данному направлению подготовки являются обеспечение условий для:

— получения полноценного и качественного фундаментального образования в области математики и компьютерных наук;
— профессиональной компетентности в области математики и компьютерных наук;
— привития навыков работы на ЭВМ, применения стандартных алгоритмических языков, использование математических методов и программного обеспечения для решения прикладных задач в различных сферах профессиональной деятельности;
— формирования и укрепления потребности в приобретении новых знаний;
— выбора студентами индивидуальной программы образования.

Область профессиональной деятельности выпускников, освоивших программу бакалавриата по данному направлению подготовки включает: научно-исследовательскую деятельность в областях, использующих математические методы и компьютерные технологии; решение различных задач с использованием математического моделирования процессов и объектов, программного обеспечения; разработку эффективных методов решения задач естествознания, техники, экономики и управления; программно-информационное обеспечение научной, исследовательской, проектно-конструкторской и эксплуатационно-управленческой деятельности; преподавание цикла математических дисциплин (в том числе информатики).

Выпускник, освоивший программу бакалавриата, в соответствии с научно-исследовательским и организационно-управленческими видами профессиональной деятельности, должен быть готов решать такие профессиональные задачи как

– применение методов математического и алгоритмического моделирования при анализе прикладных проблем;

– использование базовых математических задач и математических методов в научных исследованиях;
– участие в работе научно-исследовательских семинаров, конференций, симпозиумов, представление собственных научных достижений, подготовка научных статей, научно-технических отчетов;
– контекстная обработка общенаучной и научно-технической информации, приведение ее к проблемно-задачной форме, анализ и синтез информации;
– решение прикладных задач в области защищенных информационных и телекоммуникационных технологий и систем.

Согласно ФГОС ВО выпускник, освоивший программу бакалавриата по направлению 02.03.01 Математика и компьютерные науки, должен обладать следующими компетенциями:

универсальные: способностью осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач (УК-1);

способностью определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений (УК-2);

способностью осуществлять социальное взаимодействие и реализовывать свою роль в команде (УК-3);

способностью осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и на иностранном(ых) языке(ах)(УК-4).

общепрофессиональными способностью консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа, алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности (ОПК-1);

способностью проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности (ОПК-2);

способностью самостоятельно представлять научные результаты, составлять научные документы и отчеты (ОПК-3);

способностью находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем (ОПК-4).

Выпускник, освоивший программу бакалавриата по направлению 02.03.01 Математика и компьютерные науки, должен обладать профессиональными компетенциями, соответствующими видам профессиональной деятельности, на которые ориентирована программа бакалавриата:

способностью к определению общих форм и закономерностей отдельной предметной области (ПК-1);

способностью математически корректно ставить естественнонаучные задачи, знание постановок классических задач математики (ПК-2);

способностью строго доказать утверждение, сформулировать результат, увидеть следствия полученного результата (ПК-3);

способностью публично представлять собственные и известные научные результаты (ПК-4) [7].

Таким образом, анализ основополагающих документов (ОП ВО, ФГОС ВО, профессиональный стандарт) позволяет утверждать, что отличительной чертой общественного развития XXI века является повышение уровня математических знаний, математический склад мышления становится необходимым для специалистов разных профилей и направлений научной и практической деятельности, особенно для выпускников технических вузов.

По мнению академика Л.Д. Кудрявцева, общая цель содержания всех математических курсов должна заключаться в приобретении выпускниками вузов определенной математической подготовки, в умении использовать изученные математические методы, в развитии математической интуиции, в воспитании математической культуры. Выпускникам вузов важно знать основы математического аппарата для решения теоретических и практических задач, иметь достаточно высокий уровень развития логического мышления, уметь переводить практическую задачу с профессионального языка на математический язык. Математическое образование будет наиболее эффективно способствовать формированию у будущих инженеров определенной системы профессионально значимых качеств, если его объем и содержание будут адекватными будущей производственной деятельности, а само оно будет образовывать систему в единстве с содержанием общетехнических и специальных дисциплин [3].

Математическая подготовка будущего бакалавра «Математика и компьютерные науки» интегрирует содержание следующих дисциплин: базовая часть (Численные методы, Математический анализ, Фундаментальная и компьютерная алгебра, Аналитическая геометрия, Дискретная математика, математическая логика и их приложения в информатике и компьютерных науках, Дифференциальные уравнения, Комплексный анализ, Функциональный анализ, Дифференциальная геометрия и топология, Теория вероятностей, математическая статистика и случайные процессы, Языки и технологии программирования, Технологии баз данных, Компьютерная геометрия и графика, Операционные системы); вариативная часть (Архитектура вычислительных систем и компьютерных сетей, Пакеты прикладных программ в математике, Криптографические методы защиты информации, Системы искусственного интеллекта, Теория игр и исследование операций, Теория конечных графов, Теория кодирования, сжатия и восстановления информации, Уравнения с частными производными, Методы оптимизации, Современные средства разработки программного обеспечения, Алгебраические системы, Методы алгебраической геометрии в криптогра-

фии, Теория алгоритмов, Теоретико-числовые методы в криптографии, Объектно-ориентированные языки и системы); дисциплины по выбору (Дискретный анализ, Криптографические свойства булевых функций, Актуальные проблемы фундаментальной и компьютерной алгебры, Криптографические протоколы, Основы криптоанализа, Теория псевдослучайных генераторов). Анализ структуры преподаваемых дисциплин бакалаврам данного направления указывает на необходимость формирования математической компетентности в рамках профессиональной, для достаточного уровня сформированности математических знаний в решении профессиональных задач программиста (работа с отраслевым оборудованием обработки информационного контента; разработка программного обеспечения отраслевой направленности на основе готовых спецификаций и стандартов; разработка проектной и технической документации; обеспечение содержания проектных операций; определение ресурсов, сроков, стоимости, качества и рисков проектных операций; работа по измерению и контролю качества продуктов и т. д.)

В рамках нашего исследования в структуре профессиональной компетентности будущего бакалавра направления подготовки 02.03.01 Математика и компьютерные науки, рассматривается математическая компетентность. По мнению В.Г. Плаховой, математическая компетенция студентов технических вузов – это способность обучаемых, позволяющая им применять систему усвоенных математических знаний, умений и навыков в исследовании математических моделей профессиональных задач, включающая умения логически мыслить, оценивать, отбирать и использовать информацию, самостоятельно принимать решения [5]. Математическая компетентность, по мнению Н.А. Казачек, представляется как интегральное свойство личности, выражающееся в наличии глубоких и прочных знаний по математике, в умении применять имеющиеся знания в новой ситуации, способности достигать значимых результатов и качества в деятельности [2]. По мнению Э.Н. Гусинского, математическая компетенция — это «способность структурировать данные (ситуацию), вычленять математические отношения, создавать математическую модель ситуации, анализировать и преобразовывать ее, интерпретировать полученные результаты» [1].

В своей работе, Л.Б. Усова рассматривает математическую компетентность будущего бакалавра безопасности жизнедеятельности как интегративное качество личности, характеризующееся наличием глубоких математических знаний, умением применять их в решении практико-ориентированных заданий профильной направленности, готовностью их использования в проектировании методов и систем обеспечения техносферной безопасности, обоснованным выбором устройств и способов защиты человека и природной среды[6].

В контексте изложенного выше, можно определить математическую компетентность будущего бакалавра направления подготовки 02.03.01 «Математика и компьютерные науки» профиля «Алгоритмы и приложения компьютерной математики» как сформированное в процессе обучения качество личности, об-

ладающее наличием глубоких математических знаний и умений применять методы математического и алгоритмического моделирования при анализе прикладных задач профильной направленности; готовностью их использовать в решении прикладных задач в области защищенных информационных и телекоммуникационных технологий и систем; способностью структурировать общенаучную и научно-техническую информацию, приведением ее к проблемно-задачной форме, анализу и синтезу информации.

Специфическая особенность формирования математической компетентности бакалавра направления подготовки 02.03.01 «Математика и компьютерные науки» такова, что качественное математическое образование способно обеспечивать высокий уровень творческой деятельности, являющейся многоплановой, сложной, масштабной, требующей наличия алгоритмического мышления, междисциплинарных системных и обобщенных знаний и их обновления, целеустремленности, высокой работоспособности, отдачи огромных физических и духовных сил, неординарного подхода к достижению конечного результата. Математические дисциплины имеют огромные ресурсы для развития их творческого потенциала, становления как специалиста, поскольку развитие способностей, умений математического и алгоритмического моделирования происходит на основе изучения математических дисциплин. Тем самым математическая подготовка становится системообразующим началом в подготовке специалистов данного направления подготовки. Таким образом, математическая подготовка будущих специалистов является одной из важнейших характеристик профессионального становления, а математическая компетентность является базовой составляющей его профессиональной компетентности.

Список литературы

1. *Гусинский, Э. Н. Введение в философию образования: учеб. пособие для вузов / Э. Н. Гусинский, Ю. И. Турчанинова. – Москва: Логос, 2003. - 248 с. - ISBN 5-94019-079-1.*
2. *Казачек, Н. А. Математическая компетентность будущего учителя математики / Н. А. Казачек // Известия РГПУ им. А. И. Герцена. - 2010. - № 121. - С. 106-110.*
3. *Кудрявцев, Л. Д. Мысли о современной математике и методике ее преподавания : учеб. пособие / Л. Д. Кудрявцев. - Москва: Физматлит, 2008. - 434 с.*
4. *Национальная доктрина образования в Российской Федерации // Высшее образование сегодня. – 2001. –№2. – С. 2 – 4.*
5. *Плахова, В. Г. Формирование математической компетенции у студентов технических вузов: автореф. дис. ... канд. пед. наук: 13.00.02 / Плахова Валентина Геннадьевна. - Саранск, 2009 - 20 с.*
6. *Усова, Л.Б. Актуализация математических знаний будущего бакалавра безопасности жизнедеятельности: монография*

/В.Г.Гладких, Л.Б. Усова. – Оренбург: ООО ИПК «Университет», 2014. – 191 с.

7. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 02.03.01 Математика и компьютерные науки: утв. приказом М-ва образования и науки РФ от 23 августа 2017 г. № 807. Москва.

АВТОМАТИЗАЦИЯ СОЗДАНИЯ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ В ЛАБОРАТОРНОЙ ПРАКТИКЕ

Ушаков Ю.А. канд. техн. наук, доцент, Ушакова М.В.

Оренбургский государственный университет

В настоящее время довольно существенной частью учебного процесса в информационных технологиях является работа с виртуальными машинами, на которых установлены разнообразные операционные системы. На некоторых предметах, например «операционные системы» или «информационные системы», приходится иметь несколько запущенных виртуальных машин, соединенных в сеть, имеющих запущенные сервисы DHCP, выход в интернет. Однако на обычных компьютерах запустить такие инфраструктуры довольно проблематично из-за требуемых ресурсов и ограничений сети университета.

Самый необходимый инструмент для изучения операционных систем и сетей в современной учебной практике – виртуальные машины – в произвольных топологиях и количествах доступны при реализации частных облаков. Однако для реализации частного облака необходимы высококвалифицированные администраторы, которые настраивают хостовые операционные системы и сети. Самыми популярными системами для реализации частного облака типа IaaS являются OpenStack, OpenNebula, CloudStack. Другие продукты, такие как Vmware vCloud, Microsoft System Center, ProxMox не предоставляют в бесплатном варианте требуемый функционал.

Установка и настройка любой opensource платформы состоит из множества шагов, которые не так просто повторить на произвольном оборудовании. Кроме установки самой системы, необходимо настраивать беспарольный удаленный вход, сетевые мосты, права доступа, различные службы, доступ к системе хранения и прочее. Это останавливает преподавателей и системных администраторов от разворачивания учебных облаков и использования открытых систем в пользу пропиетарных технологий.

Процесс обучения с использованием частного облака представлен следующим образом. Преподаватель подготавливает образы операционных систем, создает учетную запись для группы, создает нужное количество виртуальных машин, соединённых изолированными сетями. Затем студенты пользуются заранее созданными машинами (или создают свои). После того, как необходимость использования закончилась, машины выключаются до следующего раза или удаляются. Для автоматизации этих процессов часто используют программные интерфейсы (API), а для изоляции сети удобнее всего использовать OpenvSwitch.

Для разворачивания такой облачной системы можно использовать бездисковую загрузку Linux серверов с заранее преднастроенного образа облачно-

го узла. Для этого используется Linux bootstrap по классической схеме загрузки PXE и присоединения корня файловой системы к NFS [1].

Однако при использовании NFS в качестве корня операционной системы для многих серверов, приходится использовать NFS каталог в режиме «только для чтения» в следующем варианте:

```
/var/lib/nfsroot *(ro,async,no_root_squash,no_subtree_check,no_all_squash)
```

При этом большинство функций может быть нарушено из-за запрета на запись. Существует два пути решения проблемы множественного доступа к корня: использование tmpfs/squashfs - создание отражение корня операционной системы в оперативной памяти, вся запись ведется в отдельный раздел, и использование технологии overlayfs из контейнерных систем для работы с образом. Второй вариант автоматизирован через пакет overlayroot [2] и позволяет объединять несколько образов в один, записывать изменения в отдельный слой, позволяет сохранять изменения на отдельный раздел или сохранять изменения в оперативной памяти только на время работы сервера.

Учебные задания как правило подразумевают временное использование типовых конфигурации, установку и настройку программного обеспечения. Лучше всего использовать в этом случае вариант tmpfs, в котором все изменения стираются при перезагрузке. Для этого необходимо выполнить установку overlayroot в режиме tmpfs внутри файловой системы собираемого образа сервера:

```
sudo apt-add-repository ppa:cloud-initramfs-tools/ppa
sudo apt-get install -y overlayroot
echo 'overlayroot="tmpfs"' >> /etc/overlayroot.conf
```

В данном режиме корень сервера поставляется из сервера хранения по NFS в режиме «только для чтения», а все изменения записываются в отдельный слой. Однако все требуемые каталоги для работы облачной системы поставляются по NFS с самого сервера в режиме «shared storage», то есть общего хранилища. Управление в системе OpenNebula происходит по выполнению скриптов из общего хранилища по SSH, в системе Openstack – по REST запросам управляющего сервиса.

Поскольку загрузка серверов происходит по сети по протоколу PXE, требуется обеспечить начальную загрузку ядра для монтирования по NFS корня системы. Для этого в стандартной pxelinux загрузке [3] после установки и настройке DHCP и TFTP сервера необходимо прописать опции монтирования NFS и прочие параметры. Для этого нужно скопировать текущее ядро и initrd из каталога /boot в корень TFTP сервера, добавить ссылку на них и прописать все остальные параметры в строке «append»:

DEFAULT linux
LABEL linux
initrd initrd.img-4.4.0-109-generic
kernel vmlinuz-4.4.0-109-generic
append root=/dev/nfs nfsroot=192.168.0.2:/var/lib/nfsroot ip=dhcp rw ipv6.disable=1 net.ifnames=0 biosdevname=0 aufs=tmpfs acpi=force reboot=pciroot

здесь:
net.ifnames=0 biosdevname=0 - параметры для отмены именования интерфейсов в новой нотации Ubuntu 16;
aufs=tmpfs – режим работы OverlayFS;
ip=dhcp – указание взять адрес по DHCP;
root=/dev/nfs – использовать NFS как корень файловой системы;
nfsroot=192.168.0.2:/var/lib/nfsroot – путь до корня запускаемой файловой системы;
acpi=force reboot=pciroot – параметры работы ACPI.

Общая схема работы показана на рисунке 1.

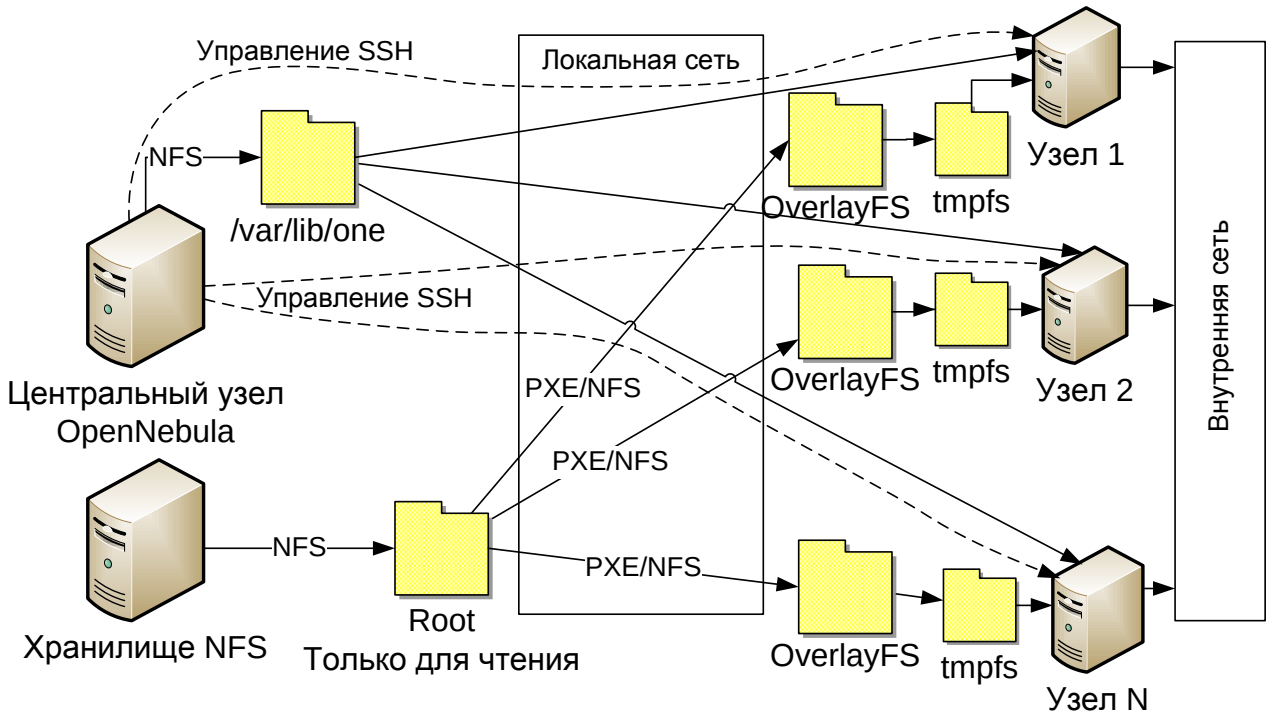

Рисунок 1 – Схема работы учебного кластера с автозагрузкой PXE

Для примера был создан образ узла OpenNebula с установленными режимом KVM, виртуальный коммутатор OpenvSwitch, подключенный ко второму интерфейсу, корень общего хранилища перенесен на сервер хранения NFS на

базе Openmediavault. Основное условие работоспособности серверов – совпадение версии ядра в папке /tftpboot и ядра внутри образа сервера.

Данный способ позволил создать один образ менее чем за час, после чего загрузить с него 20 серверов и успешно запустить первую виртуальную машину на вновь созданном облаке. Этот путь создания инфраструктуры гораздо экономичнее и быстрее, чем предлагают создатели Ubuntu (MaaS), Ansible, pupput, Kickstart, поскольку не происходит инсталяция сервера, а он загружается с заранее заданной конфигурацией. Обновление образа сервера автоматически передается на все запущенные сервера (но нужно сделать переключение снапшотов на системе хранения, чтобы все файлы одновременно обновились). Такой подход также экономит время адинистратора, а, при скачивании готового образа сервера позволяет запускать произвольные кластеры и облачные решения на любом количестве оборудования, главное что бы оно было одной архитектуры (x86_64, aarch64б sparc64 и подобное).

Статья подготовлена при поддержке РФФИ проект № 18-07-01446 и 17-07-01584.

Список литературы

1. Gavin Thomas. Boot Linux from an NFS server. https://www.gadgetdaily.xyz/boot-linux-from-an-nfs-server/ 2015

2. Justin Kulesza. Protecting the Root Filesystem on Ubuntu with Overlayroot. https://spin.atomicobject.com/2015/03/10/protecting-ubuntu-root-filesystem.- 2015.

3. Installing Debian using network booting. https://wiki.debian.org/PXEBootInstall.- 2013.

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МЕТОДОВ ВЫЧИСЛЕНИЙ В ПРЕПОДАВАНИИ МАТЕМАТИЧЕСКИХ ДИСЦИПЛИН

Ушакова М.В., Ушаков Ю.А., канд. техн. наук, доцент

Оренбургский государственный университет

В настоящее время в математическом образовании наблюдается тенденция отставания использования современных методов расчетов и анализа от стремительно развивающихся современных технологий. Многие специалисты по окончании вузов, имея достаточно крепкие знания в области фундаментальной и прикладной математики, не имеют возможности работать по специальности в связи с недостаточным освоением прогрессивных технологий и эвристических и интеллектуальных алгоритмов. Внедрение возможности качественного преподавания студентам таких перспективных направлений как, например, нейронные сети, позволит вывести будущих специалистов-математиков на более высокий уровень конкурентоспособности на рынке труда.

Использование нейронных сетей становится все более актуальным для решения большого количества разнообразных задач. Нейронные сети применяются для анализа данных, оптимизации, классификации, прогнозирования и многих других задач. Основной областью применения нейронных сетей остается обработка изображений, звука, видео, последовательных действий. Тем не менее, не стоит забывать, что нейронные сети также являются универсальными аппроксиматорами функций, поэтому они могут применяться к более «классическим» математическим задачам в качестве инструмента численного анализа.

Так, например, нейронные сети вполне успешно решают задачи, связанные с решением дифференциальных уравнений, как обыкновенных, так и в частных производных. Численное решение обыкновенных и дифференциальных уравнений с частными производными имеет большое значение для многих областей инженерии. Хотя традиционные методы, как правило, адекватны и эффективны во многих технических приложениях, их ограничение состоит в том, что полученные решения являются дискретными или имеют ограниченную дифференцируемость. Чтобы избежать этой проблемы при численном решении дифференциальных уравнений, можно реализовать другой метод, который опирается на нейронные сети[1].

Необходимость применения нейронных сетей для приближенного решения классической задачи Коши $y'(x)=F(x,y)$, $y(x0)=y0$ на промежутке $(a;b)$ на первый взгляд не ясна, так как существует множество стандартных методов её решения. Большая часть из них - это численные методы, приводящие к поточечной аппроксимации. Получение аналитического выражения из решения, заданного в конечном наборе точек, представляет собой отдельную задачу. В отличии же от численных методов, нейронная сеть позволяет получить решение

сразу в виде функции. Кроме того, нейронные сети более устойчивы по отношению к ошибкам в данных[2].

В работе [3] рассматривается применение нейросетевых методов для решения жестких дифференциальных уравнений первого порядка. Для решения жестких задач обычно используются неявные и полуявные вычислительные методы, результатом применения которых является точечная аппроксимация решения. С помощью нейросетевых методов строится их приближенное параметризированное решение в виде некоторой функции, в то время как классические методы имеют дискретный характер решения. Использование параметризованных сетей, которые позволяют получить приближенное решение на всем интервале изменения параметра, открывает широкие возможности. Описанный в данной работе подход к решению жесткого дифференциального уравнения можно применить и к уравнениям высшего порядка, а также перейти к краевым задачам, решению уравнений, заданных в частных производных и т.д.

Авторы работы [4] предлагают использование радиально-базисных нейронных сетей для решения уравнений в частных производных, играющих важную роль в различных областях науки и техники. В отличие от традиционных численных алгоритмов решения дифференциальных уравнений, нейросетевые модели устойчивы по отношению к неточностям в задании коэффициентов уравнений, начальных и граничных условий, погрешностям вычислений [5] Кроме того, существует возможность распараллеливания задачи и возможность использования набора сетей.

В качестве примеров практического применения нейронных сетей для решения задач «классической» математики можно привести разработанные нейросетевые подходы решения уравнений теплопроводности, решения задачи о катализаторе, моделирования параметров воздушной среды в тоннелях, восстановления начального профиля ударной волны, моделирования процессов деформации и разрушения образцов на основе экспериментальных данных, управления беспилотными летательными аппаратами и т.д.

Рассмотрим, например, возможность применения нейронных сетей для управления квадракоптером.

Основные работы, направленные на разработку и изучение алгоритмов построения управлений и сравнение их эффективности, включают в себя метод построения управления, основанный на теории Ляпунова [6]; алгоритм управления, в основе которого лежит пропорционально-интегрально-дифференциальный ПИД) регулятор [7]; энергетические методы, применимые для пассивных систем с недостатком управляющих воздействий [8]; методы, основанные на визуальном управлении от видеокамеры (видеокамер); методы, основанные на управлении с помощью нейросетевого регулятора, используемого в задачах стабилизации, при поиске оптимальных параметров регулятора [9].

Команда американских ученых из Университета Миссури предлагает алгоритм управления группой квадракоптеров, содержащий два двухслойных нейросетевых контроллера [10]. Назначение первого контроллера – синтез

управляющих воздействий ведущего коптера. Второй контроллер стабилизирует групповой полет на основе данных с беспроводных бортовых датчиков. В работе представлен метод оптимизации каналов связи между коптерами, основанный на теории графов.

В работе ученых из Политехнического университета Мадрида описан гибридный контроллер, состоящий из двух нейронный рекуррентных сетей с обратными связями. Авторы показывают, что при рассмотрении отдельных этапов полета достигается оптимальное управление коптером [11].

Авторы работы [12] описывают нейросетевой контроллер для управления высотой полета. Здесь используется совместная работа нейросетевого и ПИД регуляторов. Полученный алгоритм имеет быструю адаптацию к внешним воздействиям.

В работе [13] описано моделирование динамики квадрокоптера, изучение базовых траекторий, построение нейроалгоритмов управления для базовых траекторий и изучение влияния погрешностей системы на действие нейросетевого контроллера для задач взлета-посадки-парения. Нейросетевой подход к управлению летательными аппаратами предполагает создание алгоритма, способного адаптироваться к непрогнозируемым внешним возмущениям и погрешностям измерительных устройств.

Авторы описывают отдельный нейросетевой контроллер и принципы его работы на некоторых базовых траекториях. Схема синтеза управления с использованием нейросетевого контроллера состоит из блока, моделирующего датчики, нейронной сети и интегратора. Задачей блока нейронной сети является вычисление управляющих воздействий в зависимости от текущих параметров системы и требуемой точности. В основе этого блока лежит трехслойная нейронная сеть прямого распространения. Скрытый слой состоит из 10 нейронов. Весовая матрица и смещения вычисляются путем супервизорного обучения [14]. Для формирования обучающей выборки составляется семейство функций, которыми будет приближаться целевое решение. Далее с помощью численного моделирования решается обратная задача динамики по вычислению соответствующего семейства функций управления. Полученное семейство формирует набор выходов нейронной сети. В качестве входов используются параметры целевой траектории, а также информация, поступающая с блока датчиков. Блок интегратора интегрирует систему уравнений Лагранжа с учетом входящих параметров, в том числе управляющих воздействий. На выходе получается новое состояние системы, которое на следующем шаге цикла моделирования опять подается в нейронную сеть.

С помощью численного моделирования авторами рассматривались взлет-парение-посадка и полет по прямой, а также нейроуправление коптером, случаи работы регулятора для управления углом поворота, для взлета на заданную высоту с учетом погрешности датчика высоты и без нее. В результате проведенного эксперимента по моделированию показана возможность реализации нейроконтроллера управления квадрокоптером на типовых траекториях.

Успешные применения нейронных сетей к задачам математической физики внушают уверенность в том, что таким же образом можно решать значительно более широкий круг задач.

Но все же существуют проблемы, тормозящие развитие этого направления. Все описанные выше примеры применения нейронных сетей тесно переплетаются с задачами «классической» математики. Отсюда вытекает необходимость как хорошо владеть математическим аппаратом, так и четко представлять особенности нейросетевых технологий, в том числе знать используемые фреймворки и языки программирования. При разработке нейросетевх моделей необходимо решать ряд специфических вопросов, таких как определение количества слоев в нейронной сети, определение количества нейронов в каждом слое, способ установления связи между слоями и т.д. Кроме того, разработка нейросетевых моделей имеет высокие требования к вычислительным ресурсам.

Таким образом, при наличии высококвалифицированных специалистов и необходимых вычислительных ресурсов становится возможным внедрение в учебный процесс такой перспективной области науки, как нейронные сети.

Статья подготовлена при поддержке РФФИ: проект № 18-07-01446 и 17-07-01584.

Список литературы

1. *Chiaramonte, M. M. Solving differential equations using neural networks [Электронный ресурс] / M.M. Chiaramonte, M. Kiener // Режим доступа : https://docgo.net/solving-differential-equations-using-neural-networks (дата обращения : 15.01.2018).*

2. *Васильев, А. Н. Построение приближённых нейросетевых моделей по разнородным данным [Электронный ресурс] / А.Н. Васильев, Д.А. Тархов // Математическое моделирование. – 2007. – т.19, №12, с.43–51. - Режим доступа : http://www.mathnet.ru/links/3da1b9f999fb5194a0767125004f018a/mm1225.pdf (дата обращения: 15.01.2018).*

3. *Тархов, Д.А. Об использовании методов нейронных сетей для одного жесткого уравнения первого порядка // Д.А. Тархов, Т.В. Лазовская Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XIV Междунар. научно-техн. конф. – Пенза: ПДЗ, 2014. – С. 171-175.*

4. *Коваленко, А.Н. О применении нейронных сетей для решения дифференциальных уравнений в частных производных / А.Н. Коваленко, А.А. Черноморец, М.А. Петина // Научные ведомости белгородского государственного университета. Серия: Экономика. Информатика. – 2017. - №9 (258), выпуск 42. –с.103-110.*

5. *Васильев, А.Н. Нейросетевой подход к задачам математической физики / А.Н. Васильев, Д.А. Тархов, Т.А. Шемякина. // Санкт-Петербург : Нестор-История, 2015. - 259 с.*

6. Dzul, P.A. Real-time stabilization and tracking of a four-rotor mini-rotorcraft / P.A. Dzul, Lozano R. // IEEE Transaction on Control System Technology, 12(4). – p.510 – 516. - July 2004.

7. Bresciani T. Modelling, Identification and Control of a Quadrotor Helicopter // Department of Automatic Control, Lund University. - 2008.

8. Фантони, И.Нелинейное управление механическими системами с дефицитом управляющих воздействий / И. Фантони, Р. Лозано. - Москва-Ижевск: ООО "Компьютерная динамика", 2012.

9. Евгенов А.А. Нейросетевой регулятор системы управления квадрокоптером // Научное обозрение. Технические науки. – 2014. – № 1. – С. 148-149. – Режим доступа: https://science-engineering.ru/ru/article/view?id=208 (дата обращения: 17.01.2018).

10. Dierks, T. Neural Network Control and Wireless Sensor Networkbased Localization of Quadrotor UAV Formations / Dierks T., Jagannathan S. // Aerial Vehicles. 2009. - P. 601-620.

11. Munoz, R.S.M. Modelling and Identification of Flight Dynamics in Mini-Helicopters Using Neural Networks // Munoz R.S.M., Rossi C., Cruz A.B. // Aerial Vehicles. 2009. - P. 287-312.

12. Lavi B. An Adaptive Neuro PID for Controlling the Altitude of quadcopter Robot // International Conference on Methods and Models in Automation and Robotics. Poland. Volume: 18th. - 2014. - P.662-665.

13. Савицкий А.В., Павловский В.Е. Модель квадрокоптера и нейросетевой алгоритм управления // Препринты ИПМ им. М.В.Келдыша. 2017. № 77. - 20 с. – Режим доступа: http://keldysh.ru/papers/2017/prep2017_77.pdf (дата обращения : 17.01.2018).

14. Голубев Ю.Ф. Нейронные сети в мехатронике // Фундаментальная и прикладная математика. 2005, т.11, № 8, С. 81-103.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ В КОНТЕЙНЕРНЫХ СРЕДАХ

Чернова Е.В., Полежаев П.Н.

Оренбургский государственный университет

На сегодняшний день нейронные сети обрели общую популярность. Появилось множество приложений в самых разных областях, использующих нейронные сети для решения сложных задач. Но все же существуют проблемы, тормозящие развитие этого направления. Главным образом – это высокие требования к вычислительным ресурсам, используемым для вычисления нейронных сетей. Также немало важным фактором считается необходимость хорошо знать саму технологию, в том числе фреймворки и языки. Разработка методов установки необходимых программных средств, их запуска и автоматической настройки помогли бы в решении этих задач. Подобные подходы позволили бы специалисту с базовыми сведениями о работе в системе Linux начать работать с нейросетевыми фреймворками наиболее эффективным способом.

Для реализации подобных идей необходимо провести эксперимент с целью выяснения наиболее эффективных конфигураций и набора ресурсов для запуска моделей нейронной сети. Главным критерием является время, которое потенциальный пользователь потратит на получение результата работы с нейронной сетью. Эксперимент можно разделить на две стадии:

1. Обучение нейронной сети на наборе данных с использованием только ресурсов центральных процессоров виртуальной машины или контейнеров, запущенных на виртуальной машине. Исследование проведется при увеличении количества используемых вычислительных ядер от 1 до 24.

2. Обучение нейронной сети на наборе данных с использованием ресурсов центральных и графических процессоров. Запуск происходит на виртуальных машинах или запущенных на виртуальной машине контейнерах с поддержкой ускорения на графических процессорах.

Для работы с разными типами нейронных сетей была выбрана библиотека TensorFlow, как надежный и развивающийся проект, поддерживающий параллельные вычисления с помощью графических процессоров, а также Keras в качестве высокоуровневого API для TensorFlow [1]. В обоих сценариях вычисления будут запускаться как на процессоре виртуальной машины, так и внутри контейнера Docker. Во втором сценарии внутри контейнера запустится Keras с использованием nvidia-docker, который позволяет использовать мощность графических процессоров для параллельных вычислений.

В данной статье будет описана установка необходимых библиотек и фреймворков на операционную систему Linux Ubuntu. Python 2.7 уже должен быть установлен.

Для первой стадии эксперимента установим версию с поддержкой CPU (центральных процессоров). Установка TensorFlow может осуществляется с помощью следующих инструментов: virtualenv, pip, Docker, Anaconda. Будем использовать pip, следующая строка установит его в систему [2]:

sudo apt-get install python-pip python-dev

Теперь используя pip, установим Tensorflow без поддержки GPU:

pip install tensorflow

Для проверки правильности установки запустим скрипт Python:

```
import tensorflow as tf
hello = tf.constant('Hello world!')
sess = tf.Session()
print(sess.run(hello))
```

Если Tensorflow установлен правильно, появится сообщение «Hello world!», иначе – сообщение об ошибке.

Keras требует наличие нескольких пакетов, их также можно установить с помощью pip [3]:

```
pip install numpy scipy
pip install scikit-learn
pip install pillow
pip install h5py
```

После этого можно установить Keras:

pip install keras

Необходимо убедиться, что конфигурационный файл keras.json настроен верно. Особенно следует обратить внимание на параметр «backend», он отвечает за то, каким фреймворком пользуется Keras:

```
{
    "image_dim_ordering": "tf",
    "epsilon": 1e-07,
    "floatx": "float32",
    "backend": "tensorflow"
}
```

Для проверки установки достаточно написать «import keras». Если команда выполнилась без ошибок, значит Keras установлен правильно.

Чтобы снимать показания во время эксперимента будем использовать инструмент мониторинга dstat, позволяющий анализировать производительность системы. Для его установки достаточно ввести команду:

sudo apt-get install dstat

Теперь для проведения эксперимента нужен скрипт Python, который бы содержал в себе описание модели нейронной сети и команду обучения. Пример такого скрипта можно найти в официальном репозитарии Keras. Следующая строка запускает скрипт и dstat, который сохранит результат в файл в формате cvs.

*dstat -cmdr --output *файл для записи результата.cvs* | python *скрипт*,*

где ключ *c* выводит процессорную статистику, *m* – статистику памяти, *d* – диска, *r* - запросов ввода-вывода.

Вторая часть эксперимента проводится с помощью контейнеров Docker. Чтобы его установить необходимо добавить отдельный репозитарий [4]. Для этого установим следующие пакеты:

```
sudo apt-get install \
   apt-transport-https \
   ca-certificates \
   curl \
   software-properties-common
```

Добавим официальные GPG ключи Docker:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

Следующая команда добавит стабильный репозитарий для Docker:

```
sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

Теперь можно установить Docker, версию Community Edition:

sudo apt-get install docker-ce

Чтобы убедиться, что установка прошла успешно, запустим образ hello-world. Контейнер должен работать без ошибок и печатать сообщение «Hello, world!».

sudo docker run hello-world

Для начала работы контейнера Docker с TensorFlow, поддерживающего только CPU и запускающего оболочку bash, нужно ввести команду:

docker run -it gcr.io/tensorflow/tensorflow bash

Затем следует установить Keras, так же как было описано выше, но только внутри этого контейнера. Контейнер готов для проведения эксперимента.

Во второй части эксперимента обучение нейронной сети нужно проводить с использованием графических процессоров. Потребуется другое программное обеспечение. Для графических процессоров компании NVIDIA существуют готовые решения: CUDA – это платформа для параллельного вычисления на графических процессорах, cuDNN – мощная библиотека для машинного обучения. Дистрибутивы распространяются бесплатно на официальном сайте.

На странице CUDA Toolkit необходимо выбрать операционную систему (Linux), архитектуру (X86_64), дистрибутив (Ubuntu), версию (16.04), тип дистрибутива (runfile) и скачать его [5]. Для выбранных нами параметров установка библиотеки начнется после введения следующей строки:

sudo sh cuda_9.1.85_387.26_linux.run

Для того чтобы скачать дистрибутив cuDNN необходимо войти под своей учетной записью на официальный сайт (или зарегистрироваться) [6]. После этого нужно подтвердить свое согласие с лицензионным соглашением и выбрать версию дистрибутива. Мы выбрали библиотеку для Linux версию 7.0.5 для CUDA 9.1. Следующая строка распакует скачанный архив:

tar -xzvf cudnn-9.1-linux-x64-v7.tgz

Файлы нужно скопировать в директорию CUDA Toolkit:

sudo cp cuda/include/cudnn.h /usr/local/cuda/include
sudo cp cuda/lib64/libcudnn /usr/local/cuda/lib64*
sudo chmod a+r /usr/local/cuda/include/cudnn.h
*/usr/local/cuda/lib64/libcudnn**

Установка библиотек для графических процессоров завершена. Последующая установка вспомогательных библиотек и фрэймфорков ничем не отлича-

ется от описанных выше, кроме версии TensorFlow [7]. Если установлена версия без поддержки графических процессоров, то ее нужно предварительно удалить. Команду установки TensorFlow следует заменить на следующую:

pip install tensorflow-gpu

Вместо обычного будем ставить контейнер Docker с поддержкой графических процессоров – nvidia-docker [8]. Сначала нужно добавить репозитарии:

curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -

curl -s -L https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list

sudo apt-get update

Затем устанавливается nvidia-docker2 и перегружается конфигурация демона Docker:

sudo apt-get install -y nvidia-docker2
sudo pkill -SIGHUP dockerd

Чтобы проверить, что установка прошла успешно, запустим утилиту для мониторинга графических процессоров nvidia-smi:

docker run --runtime=nvidia --rm nvidia/cuda nvidia-smi

Теперь можно запустить контейнер TensorFlow с поддержкой графических процессоров:

nvidia-docker run -it gcr.io/tensorflow/tensorflow:latest-gpu bash

Установка Keras осуществляется аналогично тому, как было написано выше. На этом подготовка к эксперименту завершена.

Эксперимент проводился с помощью услуг публичных провайдеров – 1Cloud, Azure, Google Cloud. У них можно заказывать виртуальные сервера нужной конфигурации с поддержкой GPU и без нее. В качестве набора данных использовались набор изображений 28x28 с рукописными цифрами MNIST и набор цветных изображений 32x32 с 10 классами CIFAR-10. Однако для первой части эксперимента использовался только MNIST (рис.1) , а для второй – оба набора (рис. 2, 3), чтобы лучше проследить зависимость времени от конфигурации системы. Это объясняется тем, что нейронной сети проще обрабатывать MNIST, чем CIFAR-10, а мощности графических процессоров больше, чем центральных.

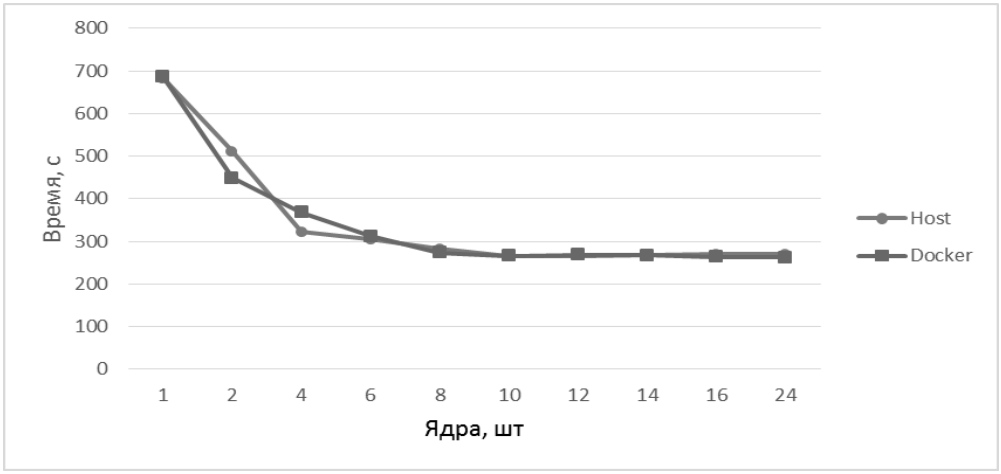

Рисунок 1 – Время обучение модели нейронной сети на наборе MNIST

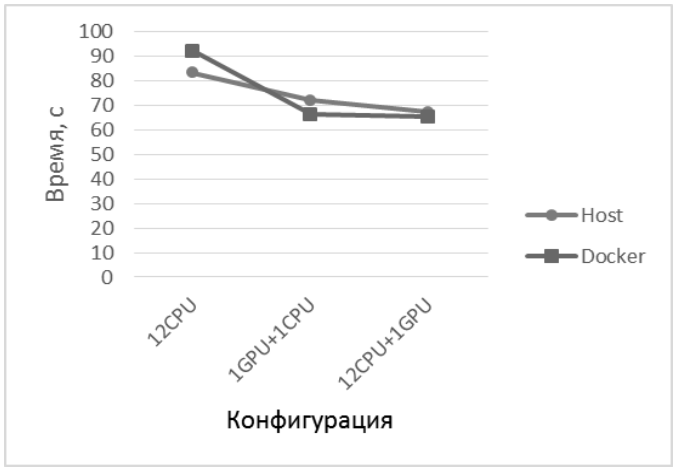

Рисунок 2 – Время обучение модели нейронной сети на наборе MNIST

Рисунок 3 – Время обучение модели нейронной сети на наборе CIFAR-10

Результаты показали, что использование контейнеризации Docker при достаточном количестве вычислительных ядер позволяет не ухудшить показатели производительности и стоимости, при этом существенно упрощает первоначальный запуск модели. Однако Docker с поддержкой графических процессоров не готов к автоматизации на сегодняшний день, так как требуется слишком много индивидуальных настроек вручную. Тем не менее, при написании статьи был подготовлен скрипт, устанавливающий нужные драйвера и вспомогательные пакеты.

Исследование выполнено при финансовой поддержке Правительства Оренбургской области и РФФИ (проекты №17-47-560046, №16-29-09639 и №18-07-01446), Президента Российской Федерации в рамках стипендии для молодых ученых и аспирантов (СП-2179.2015.5).

Список литературы

1. Чернова Е.В., Полежаев П.Н. Анализ существующих технологий нейронных сетей // IV Международная научно-техническая конференция «Новые информационные технологии и системы» (НИТиС-2017), 2017. – С. 232-236.

2. Installing TensorFlow [Электронный ресурс] //Режим доступа: https://www.tensorflow.org/install/ – Загл. с экрана. – (Дата обращения: 22.12.2017)

3. Get Started [Электронный ресурс] //Режим доступа: https://docs.docker.com/get-started/ – Загл. с экрана. – (Дата обращения: 24.12.2017)

4. Keras: The Python Deep Learning library [Электронный ресурс] //Режим доступа: https://keras.io/ – Загл. с экрана. – (Дата обращения: 24.12.2017)

5. CUDA Zone [Электронный ресурс] //Режим доступа: https://developer.nvidia.com/cuda-zone – Загл. с экрана. – (Дата обращения: 25.12.2017)

6. NVIDIA cuDNN [Электронный ресурс] //Режим доступа: https://developer.nvidia.com/cudnn – Загл. с экрана. – (Дата обращения: 25.12.2017)

7. Чернова Е.В., Полежаев П.Н. Архитектура распределенной библиотеки Distributed TensorFlow // Компьютерная интеграция производства и ИПИ-технологии: материалы VIII Всероссийской научно-практической конференции. – Оренбург, 2017. – С. 354-357.

8. NVIDIA Docker: GPU Server Application Deployment Made Easy [Электронный ресурс] //Режим доступа: https://devblogs.nvidia.com/parallelforall/nvidia-docker-gpu-server-application-deployment-made-easy/ – Загл. с экрана. – (Дата обращения: 28.12.2017)

АРХИТЕКТУРА ПРОТОТИПА СИСТЕМЫ МНОГОАДРЕСНОЙ ПЕРЕДАЧИ ШИРОКОПОЛОСНОГО МУЛЬТИМЕДИЙНОГО ТРАФИКА

Шухман А.Е., канд. пед. наук, доцент,
Полежаев П.Н., Ушаков Ю.А., канд. техн. наук, доцент,
Легашев Л.В.
Оренбургский государственный университет

В настоящее время весьма актуальной является разработка различных сетевых решений на базе SDN (Software Defined Networking, программно-конфигурируемые сети) и NFV (Network Function Virtualization, виртуализация сетевых функций). В рамках данной статьи описывается решение, представляющее собой прототип системы многоадресной передачи широкополосного мультимедийного трафика для провайдеров IPTV на базе SDN.

Основное назначение данного прототипа – повышение эффективности работы сетей провайдеров IPTV [1].

На рисунке 1 приведена архитектура прототипа системы.

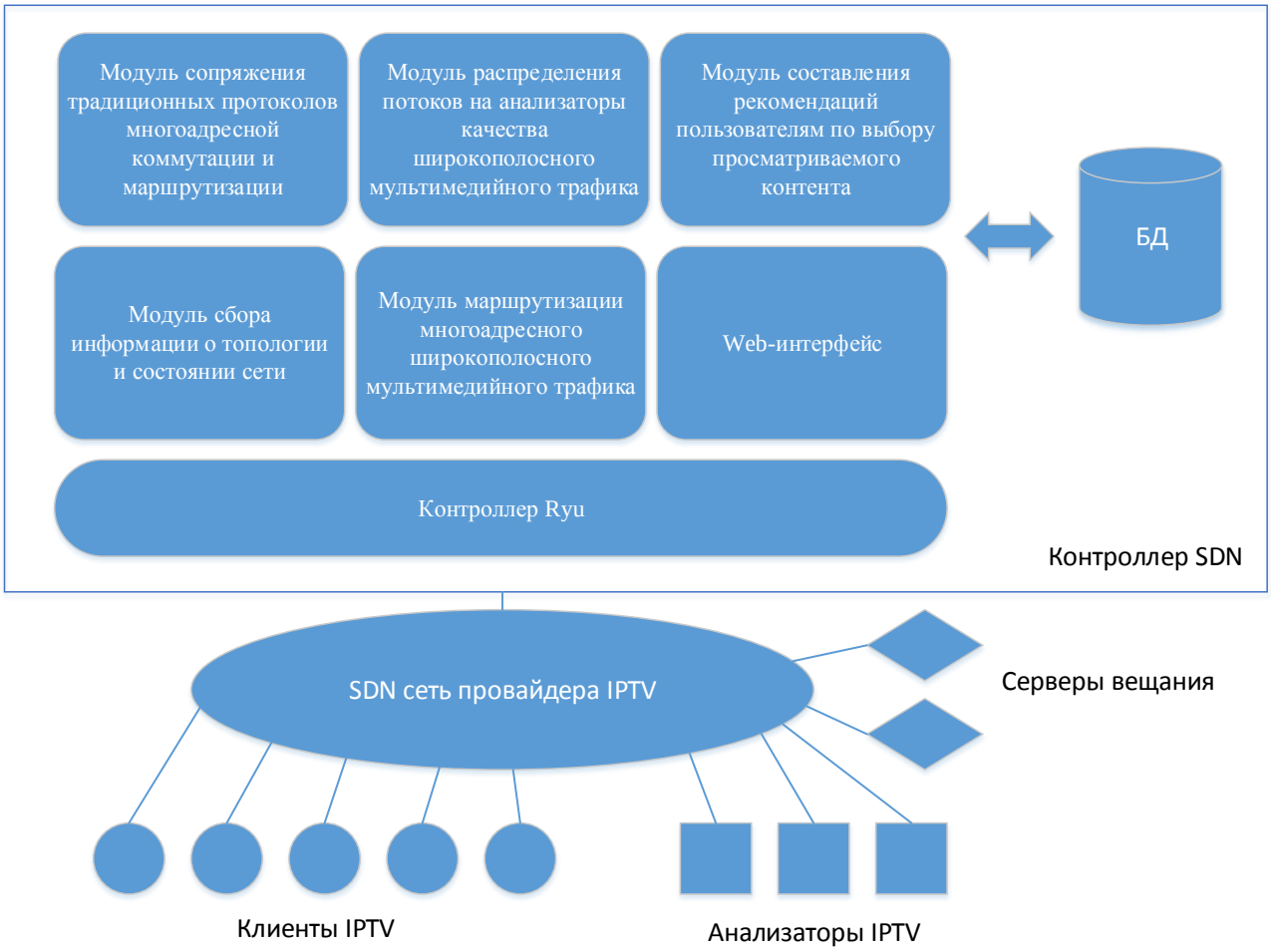

Рисунок 1 – Архитектура прототипа системы

Одной из основных компонент прототипа системы многоадресной передачи широкополосного мультимедийного трафика является контроллер SDN, который управляет программно-конфигурируемой сетью. В качестве готового контроллера была выбрана система Ryu. Для нее были разработаны отдельные модули, реализующие ранее созданные алгоритмические решения [2-5]:

а) Модуль сопряжения традиционных протоколов многоадресной коммутации и маршрутизации с алгоритмами маршрутизации многоадресного широкополосного мультимедийного трафика на базе программно-конфигурируемых сетей. Конечное оборудование пользователей IPTV, как правило, использует протокол IGMP, поэтому для корректной работы с многоадресной рассылкой необходимо эмулировать пакеты IGMP. Для этого используется класс ryu.lib.packet.igmp контроллера Ryu. Также необходима эмуляция протокола PIM для взаимодействия с серверами вещания. Для сопряжения с PIM разработанный алгоритм использует режим Sparse, потому что в этом режиме известен вещатель (RP, точка рандеву) и до него можно проложить маршрут. Общая схема работы алгоритма для PIM совпадает с IGMP, однако используются определенные PIM способы поиска ближайшего маршрутизатора (пакет PIM Hello), присоединения к нему (PIM Join), проверки получателей (PIM Register).

б) Модуль сбора информации о топологии и состоянии программно-конфигурируемой сети. Основывается на данных от SDN контроллера, обычных коммутаторов (LLDP, STP, IPv6) и компьютеров (SMB, Kerberos, UPnP, ARP). Собираемая информация позволяет получить адекватное представление сети, используемое прочими модулями системы.

в) Модуль распределения потоков на анализаторы качества широкополосного мультимедийного трафика на базе программно-конфигурируемых сетей. Использует модуль маршрутизации с целью включения в группу вещания анализируемого канала одного или нескольких анализаторов в зависимости от их типа. При распределении учитываются ограничения каждого анализатора на количество анализируемых потоков. В случае если количество транслируемых потоков превышает общее количество анализируемых потоков для всех анализаторов, с заданной периодичностью происходит переключение анализируемого потока на другой, так чтобы при этом каждый поток обязательно попал в число анализируемых. Анализаторы поддерживают определение следующих метрик: Continuity Counters (CC), IAT (Inter-packet Arrival Time), Media Loss Rate (MLR), Delay Factor (DF) и др.

г) Модуль маршрутизации многоадресного широкополосного мультимедийного трафика на базе программно-конфигурируемых сетей. Может запус-

каться в разных режимах с указанием выбираемого алгоритма – муравьиного алгоритма для деревьев Штейнера на ориентированных мультиграфах или эвристического алгоритма на базе объединения путей, найденных алгоритмом Дейкстры.

д) Модуль составления рекомендаций пользователям по выбору просматриваемого контента. Для составления рекомендаций пользователям системы по выбору просматриваемого контента проведена классификация контента по различным жанрам. Жанры отнесены к критериям с заданной степенью принадлежности. Наборы жанров, критериев и оценок степени влияния предоставляются вместе с системой и сформированы на основе экспертных оценок. Каждый оператор имеет возможность изменять состав предоставляемых каталогов в зависимости от своих целей, а также формировать свои новые каталоги. Данные о времени потребления абонентом контента по каждому жанру определяются на основе базы данных оператора. На основе этих данных вычисляется вектор критериев – профиль каждого абонента. После этого задача нахождения рекомендуемых жанров сводится к задаче сравнения векторов – необходимо найти жанры, имеющие влияние на те критерии, к которым, согласно составленным профилям, относится абонент. Для решения задачи используется метод коллаборативной фильтрации, основанный на анализе данных по большому числу абонентов [6].

е) Web-интерфейс системы. Реализует простой визуальный интерфейс для прототипа системы. Сайт реализован на базе библиотеки Django на языке программирования Python.

ж) БД – база данных. Служит хранилищем данных для прочих компонент системы. Реализована на базе СУБД PostgreSQL.

Также на рисунке 1 изображены: SDN сеть провайдера IPTV (многоуровневая территориально распределенная сеть, включающая ядро, уровень распределения и доступа), клиенты IPTV (приставки конечных пользователей), анализаторы IPTV и серверы вещания.

На рисунке 2 представлена диаграмма вариантов использования UML, иллюстрирующая основных пользователей прототипа системы (действующие лица) и доступные им функции (варианты использования).

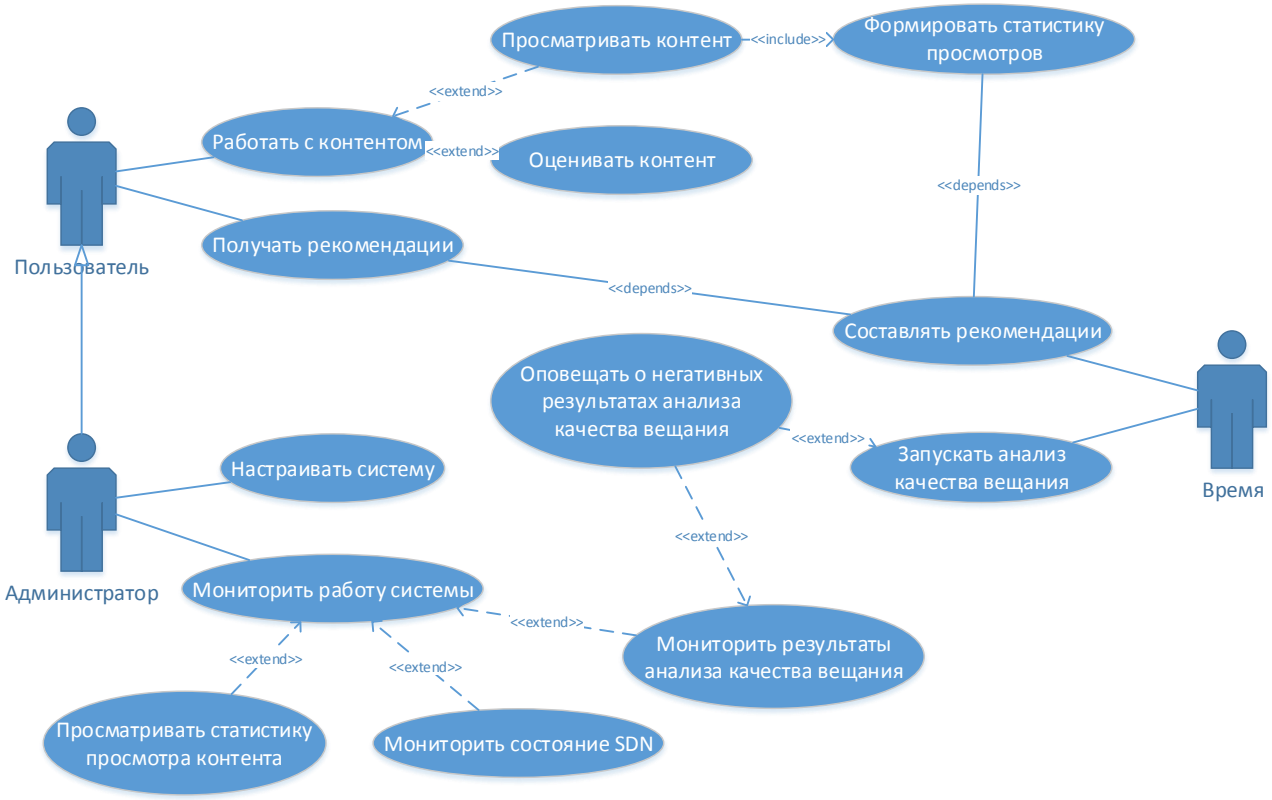

Рисунок 2 – Диаграмма вариантов использования прототипа системы

Основные действующие лица:

а) Пользователь – обычный пользователь-потребитель контента IPTV.

б) Администратор – специальный пользователь, отвечающий за настройку прототипа системы.

в) Время – выполняет операции через регулярные интервалы времени.

Пользователь при взаимодействии с прототипом системы может:

а) Работать с контентом, в т.ч. его просматривать (параллельно собирается и формируется статистика просмотров) или оценивать.

б) Получать составленные рекомендации по контенту для последующего просмотра.

Администратор имеет возможность:

а) Настраивать систему – задавать параметры работы для всех компонентов системы.

б) Мониторить работу системы, в т.ч. просматривать статистику просмотра контента, мониторить состояние SDN или мониторить результаты анализа качества вещания, получать оперативные оповещения о негативных результатах анализа.

Время может периодически:

а) Составлять рекомендации и отдавать их пользователям.

б) Запускать анализ качества вещания.

Данный прототип реализован на базе оборудования Оренбургского государственного университета и прошел экспериментальную апробацию.

Работа выполнена при поддержке РФФИ (проект №15-07-06071).

Список литературы

1 Ушаков Ю.А., Полежаев П.Н., Шухман А.Е., Бахарева Н.Ф. Реализация симулятора инфраструктуры для многоадресной широкополосной передачи мультимедийного трафика на базе программно-конфигурируемых сетей // Университетский комплекс как региональный центр образования, науки и культуры [Электронный ресурс]: материалы Всероссийской научно-методической конференции; Оренбург. гос. ун-т. - Электрон. дан. - Оренбург: ОГУ, 2017. - С. 3234-3241.

2 Ushakov Yu., Polezhaev P., Legashev L., Bolodurina I., Shukhman A., Bakhareva N. Increasing the Efficiency of IPTV by Using Software-Defined Networks // Proceedings of 16th International Conference, NEW2AN 2016 and 9th Conference ruSMART 2016, St. Petersburg, Russia. Lecture Notes in Computer Science. - vol. 9870. - PP. 550-560.

3 Москалева Т.С., Полежаев П.Н. Обзор существующих технологий для IPTV // Университетский комплекс как региональный центр образования, науки и культуры [Электронный ресурс]: материалы Всероссийской научно-методической конференции; Оренбург. гос. ун-т. - Электрон. дан. - Оренбург: ОГУ, 2016. - 1 электрон. опт. диск (CDROM): зв., цв.; 12 см. - Систем. требования: IBM PC 686 (Pentium II, K6-2); MS Windows9.x/NT 5.x; процессор 233 МГц; оперативная память 128 Мб; доп. программные средства: веб-браузер; Adobe Acrobat Reader XI - Загл. с этикетки диска. . - С. 2512-2519.

4 Бахарева Н.Ф., Полежаев П.Н., Ушаков Ю.А., Шухман А.Е., Легашев Л.В. Имитационная модель инфраструктуры многоадресной передачи широкополосного мультимедийного трафика в программно-конфигурируемых сетях // Интеллект. Инновации. Инвестиции. - 2016. - №8. - С. 75-78.

5 Полежаев П.Н., Ушаков Ю.А., Шухман А.Е., Бахарева Н.Ф. Применение технологии программно-конфигурируемых сетей для многоадресной передачи широкополосного мультимедийного трафика в системах IPTV // Интеллект. Инновации. Инвестиции. - 2015. - №3. - С. 84-90.

6 Ricci F. Recommender Systems Handbook / F. Ricci, L. Rokach, B. Shapira, P. B. Kantor // Springer Science+Business Media, LLC. – 2011. – 1003 c.

РЕАЛИЗАЦИЯ ПРИНЦИПА ИСТОРИЗМА ПРИ ИЗУЧЕНИИ АЛГОРИТМОВ И ВЫЧИСЛИТЕЛЬНЫХ МЕТОДОВ

Шухман Е.В., канд. физ.-мат. наук

Оренбургский государственный педагогический университет

Реализация образовательных программ в системе общего среднего и высшего образования опирается на ряд принципов, одним из которых является принцип историзма. В современном понимании историзм является один из ведущих гносеологических принципов, требующий изучать предметы или явления в процессе их становления и развития, в органической связи с порождающими их условиями. Реализация данного принципа дает возможность не только формировать знания, умения и навыки, но и приобщать учащихся к культурным ценностям.

Использование исторического материала в отечественном математическом образовании имеет давние традиции. Исторические сведения присутствуют в работах Ф.Магницкого, Л.Эйлера, Н.И.Лобачевского, М.В.Остроградского, П.Л.Чебышева. В XX веке вопросы использования исторического материала при обучении математики рассматривались в работах Г.П. Матвиевской, С.С. Демидова, Г.И.Глейзера, Б.В.Гнеденко, К.А.Рыбникова, Т.С. Поляковой и др.

Однако вопросы использования исторического материала в преподавании информатики очень мало изучены в методической литературе. Связано это, в первую очередь, с тем, что информатика является очень молодой наукой: ее структура сформировалась только в 50-60-е гг. XX века, после создания первых компьютеров. Фундаментальная сущность информатики, как науки об общих законах и свойствах информации и информационных процессов (поиска, передачи, хранения, обработки и использования информации), стала очевидной только в 70-е гг. прошлого века, когда и появился термин «информатика».

В то же время к истории информатики можно отнести многовековую историю развития средств и методов вычислений, хранения и передачи информации, начиная от истоков человеческой цивилизации в Древнем Египте и Вавилоне. Отметим, что в пособиях по истории информатики наиболее подробно рассмотрена лишь история вычислительной техники [1] и методов хранения и передачи информации [2]. Вопросы появления и развития алгоритмов и вычислительных методов до последнего времени не были представлены в учебной литературе.

Нами разработано учебное пособие [3], включающее как исторический материал по истории алгоритмов и вычислительных методов, так и их подробное описание, а также задания к лабораторным работам с учетом возможности применения рассматриваемых методов и алгоритмов для решения

актуальных задач обработки данных. Особенностью пособия стало включение в него новейших научных результатов в области истории алгоритмов и вычислительных методов, полученных за последние 10 лет.

Разработанные лабораторные работы могут использоваться не только в курсе «История информатики», но и в курсах «Структуры и алгоритмы компьютерной обработки данных» и «Анализ алгоритмов».

Так при изучении алгоритмов преобразования чисел из одной системы счисления в другую целесообразно опираться на исторический материал, представляющий основные исторические системы счисления: древнеегипетскую, вавилонскую, римскую и славянскую.

При изучении двоичной системы счисления есть смысл рассмотреть алгоритмы преобразования целых чисел, предложенные в трудах Томаса Хэрриота и Готфрида Вильгельма Лейбница. Для преобразования дробных чисел можно сравнить методы Лейбница и Леонарда Эйлера [4,5]. Также преобразования двоичных чисел в шестнадцатеричную систему и обратно могут быть рассмотрены на основе рукописных материалов Лейбница [3].

Большие возможности по использованию исторического материала возникают при изучении алгоритмов на графах. Основоположником теории графов является Леонард Эйлер. Он впервые поставил и решил задачи о кенигсбергских мостах и о поиске маршрута коня на шахматной доске, проходящего через все поля один раз [6]. Важно, что алгоритмы Эйлера могут использоваться и для решения современных задач, сводящихся к поиску эйлеровых и гамильтоновых циклов

Очень интересна история задач дискретной оптимизации. Так, задача о назначениях в геометрической форме была поставлена еще Гаспаром Монжем. В 40-е годы XX в. задача о назначениях сводилась к задаче линейного программирования транспортного типа и решалась только путем полного перебора вариантов. В 1955 г. Харолд Кун предложил «венгерский метод» – эффективный алгоритм решения задачи о назначениях, основанный на работах Кенига и Эгервари. Интересно, что в 2006 году Кун обнаружил венгерский алгоритм в малоизвестной работе XIX в. Карла Густава Якоби. В 60-е гг. XX в. задача была сведена к поиску полного паросочетания минимального веса, которая в свою очередь решается методами поиска максимального потока минимальной стоимости. Постановка задачи о максимальном потоке была связана с изучением сети железных дорог Советского Союза.

В пособии значительный исторический материал приведен в разделе, посвященном методам вычисления приближенных значений известных математических констант: числа π, числа e, константы Эйлера-Маскерони γ. Рассмотрены ранее не упоминавшиеся в учебниках методы Эйлера для вычисления констант [7,8], в том числе дробно-рациональные приближения, известные в современной математике как аппроксимации Паде [9]. Задания к лабораторной работе включают задачи на разработку программ,

осуществляющих вычисления констант различными методами, изложенными в теоретической части.

Также исторический материал может быть использован при изучении криптографических алгоритмов и методов криптоанализа. Симметричные шифры имеют богатую историю, связанную с именами Юлия Цезаря, Франсуа Виета, Блеза Виженера. Основная деятельность Христиана Гольдбаха на службе в Министерстве иностранных дел Российской империи также заключалась в дешифровальной работе. В историю криптоанализа вписаны яркие страницы, связанные с расшифровкой шифра немецкой шифровальной машины «Энигма» группой английских криптоаналитиков под руководством Алана Тьюринга. История асимметричных систем шифрования интересна тем, что основные идеи и алгоритмы были разработаны Клиффордом Коксом для английской разведки в 1973 г., что оставалось засекреченным до 1996 г.

Лабораторный практикум прошел апробацию в Оренбургском государственном педагогическом университете в 2017-18 уч. году при изучении алгоритмов на 3 курсе направления бакалавриата «Педагогическое образование», профиль «Информатика». Следует отметить, что использование исторического материала при изучении алгоритмов и вычислительных методов значительно повышает мотивацию учащихся, способствует развитию интереса к информатике. Деятельность ученых может послужить примером для самостоятельного творчества учащихся.

Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 15-33-01300).

Список литературы

1. *Душутин Н.К. Из истории вычислительной техники: учеб. пособие/ Н. К. Душутин, С. Н. Ушакова, Ю. В. Ясюкевич– Иркутск: Изд-во ИГУ, 2011. – 275 с.*

2. *Левин В.И. История информационных технологий: учебное пособие– М: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. –336 с.*

3. *Шухман Е.В. История информатики. Лабораторный практикум: учебное пособие – Оренбург: ОГПУ, 2017. – 100 с.*

4. *Шухман Е. В. О недесятичных представлениях дробных чисел в работах математиков XVII-XVIII вв //История науки и техники. – 2013. – №. 1. – С. 3-16.*

5. *Шухман Е.В. Недесятичные дроби в работах Г.В. Лейбница и Л. Эйлера // Вестник Оренбургского государственного университета. – 2013. – № 12 (161). – С. 196-202.*

6. *Шухман Е. В. Алгоритм Эйлера поиска гамильтоновых циклов и путей //История науки и техники. – 2014. – №. 12. – С. 3-11.*

7. *Шухман Е.В. Приближенное вычисление числа пи с помощью ряда для arctg x в опубликованных и неопубликованных работах Леонарда Эйлера. // История науки и техники. – 2008 – №4. – С. 2-17.*

8. Шухман Е.В. Приближенное вычисление некоторых математических констант в опубликованных и неопубликованных работах Л. Эйлера. // Вестник Оренбургского государственного университета. – 2010. – №9. – С. 74-80.

9. Шухман Е. В. Приближенное вычисление константы e в опубликованных и неопубликованных работах Леонарда Эйлера //История науки и техники. – 2012. – №. 3. – С. 19-28.