

международная студенческая конференция

IT-МАРАФОН «АЗИАТСКИЙ ХАКАТОН»

19 сентября 2015 г.

Барнаул

Редакционная коллегия:

Шайдуров А.А. (главный редактор), Шмаков И.А.

IT-МАРАФОН «АЗИАТСКИЙ ХАКАТОН»: Сборник статей международной студенческой конференции. - Барнаул: Корвус, 2015. – 53 с.

Настоящий сборник подготовлен по результатам международной студенческой конференции студентов, магистрантов из различных вузов России, Казахстана, Кыргызстана, отражающим современное видение молодыми исследователями проблем разработки и использования свободного программного обеспечения в Российской Федерации и странах ближнего зарубежья.

УДК 004.6

СПО И ГОСУДАРСТВО: СОВРЕМЕННЫЕ ТЕНДЕНЦИИ И ПЕРСПЕКТИВЫ

О. В. Журенков

Каждый, кто сталкивается с проблемой внедрения свободного программного обеспечения (СПО) в своей организации, наверняка задаётся вопросом: «Что думает правительство об СПО?». Наши СМИ (по крайней мере подавляющее большинство) упорно молчит по этому вопросу, так же как и о достижениях России в различных отраслях промышленности и технологиях. Тем не менее, на государственном уровне предпринимаются решительные, чёткие, однозначные шаги по переходу на СПО.

Первым таким шагом стало принятие национального стандарта Российской Федерации ГОСТ Р ИСО/МЭК 26300-2010 «Информационная технология. Формат Open Document для офисных приложений (OpenDocument) v1.0», идентичного международному стандарту ИСО/МЭК 26300:2006 «Информационная технология. Формат Open Document для офисных приложений (OpenDocument) v1.0», с датой введения в действие 1 июня 2011 г. с правом досрочного применения [1]. Приказ об утверждении этого стандарта [2] носит рекомендательный характер.

Хотя напрямую в этом стандарте ничего не говорится об СПО, ИТ-специалисты понимают, что внедрение этого стандарта в организации приведёт к неминуемому переходу на СПО, по крайней мере для офисных приложений (самый распространённый сейчас в стране пакет MS Office не поддерживает этот стандарт). Любая компания, достигшая 3-го уровня зрелости СММ, должна опираться в своей деятельности на стандарты, а офисная деятельность, внутренний документооборот, как правило, присутствует в любом бизнесе. При этом у компании есть выбор: использовать готовый стандарт для офисных приложений или разработать свой стандарт. Понятно, что последний вариант влечёт дополнительные расходы. Таким образом, утверждение национального стандарта ГОСТ Р ИСО/МЭК 26300-2010 стимулирует использование свободного ПО.

Вторым, более решительным шагом государства на пути к СПО, стало создание (в рамках национальных технологических платформ) национальной программной платформы на базе свободного программного обеспечения, а затем внедрение этой платформы в государственные и другие предприятия [3]. Подобные программы соответствуют стратегии усиления национальной безопасности и действуют не только в РФ, но и в Евросоюзе.

Основная миссия технологической платформы «Национальная программная платформа» — создание в России мощной индустрии разработки программного обеспечения, завоевание страной статуса мирового центра инноваций в области программного обеспечения и вывод России на передовые позиции в мире в этой области.

В ходе выполнения данной миссии будет обеспечена технологическая независимость государства в области создания информационных систем и

созданы условия для обеспечения информационной безопасности критически важных систем. Это позволит в краткосрочной перспективе добиться следующих значимых результатов:

- ликвидация отставания в объёме и уровне использования ИТ в экономике, государственном управлении и общественной жизни;
- развитие системы образования, прикладных и фундаментальных исследований в области ИТ;
- развитие отечественных центров разработки информационных технологий мирового класса за счёт расширения интеграционных связей между фундаментальной и прикладной наукой, системой образования и промышленностью, в том числе международными;
- изменение структуры затрат (в т. ч. государственных структур) на ИТ и переориентацию финансовых потоков на отечественный рынок (импортозамещение);
- повышение конкурентоспособности отечественных ИТ-продуктов на отечественном и мировом рынках.

Развитие производства, системное проведение исследований, создание и вывод на рынок перспективных коммерческих технологий, продуктов и услуг, связанные с наукоёмкими прорывными информационными технологиями, — всё это является стратегическим движением к инновационной экономике, отвечающей современным мировым тенденциям.

Для решения поставленных задач был разработан План перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения на 2011–2015 годы [4]. План состоит из трёх частей:

1. Мероприятия в сфере образования и развития профессиональных навыков федеральных государственных гражданских служащих.
2. Мероприятия по техническому обеспечению перехода федеральных органов исполнительной власти на использование свободного программного обеспечения.
3. Мероприятия по организационному обеспечению перехода федеральных органов исполнительной власти на использование свободного программного обеспечения.

Основные результаты реализации этого плана:

1. Разработка квалификационных требований, образовательных программ и учебно-методических материалов, рекомендованных для применения в образовательных учреждениях высшего, среднего и дополнительного профессионального образования, осуществляющих профессиональную переподготовку и повышение квалификации федеральных государственных гражданских служащих.
2. Формирование пакета базового свободного программного обеспечения для решения типовых задач деятельности федеральных органов исполнительной власти с учётом потребностей федеральных органов исполнительной власти в видах программного обеспечения. Создание и обеспечение

функционирования единого репозитория СПО, используемого в федеральных органах исполнительной власти.

Основные исполнители: Минкомсвязь России, Минобрнауки России, Минздравсоцразвития России.

В рамках этого плана, соответствии с постановлением Правительства Российской Федерации от 30.01.2013 № 62 Минкомсвязью России создан национальный фонд алгоритмов и программ для электронных вычислительных машин (ФГИС ФАП). Одной из целей создания Фонда является обеспечение возможности многократного использования алгоритмов и программ для ЭВМ, созданных или приобретённых с привлечением средств федерального бюджета и бюджетов государственных внебюджетных фондов.

«Гослинукс» стала первой сертифицированной операционной системой на базе СПО, пригодной для использования во всех государственных органах, государственных внебюджетных фондах и органах местного самоуправления [5].

Разработка операционной системы типового дистрибутива АИС ФССП России, получившей наименование «Гослинукс» (транслитерация GosLinux согласована с институтом Linux Trademark), велась в ФССП (Федеральной службе судебных приставов) России в течение 2013 года в рамках госконтракта на доработку, внедрение и сопровождение подсистем автоматизированной информационной системы ФССП России на основе анализа деятельности судебных приставов-исполнителей, потребностей администраторов и применяемых в службе информационных технологий. Основным разработчиком дистрибутива выступила компания «РЕД СОФТ».

Операционная система разработана в двух исполнениях — для серверов и рабочих станций, устанавливается с преднастроенными средствами защиты информации.

За основу был выбран дистрибутив CentOS 6.4, который на тот момент широко применялся на серверах подсистем АИС ФССП России. Для повышения совместимости с аппаратной частью в дистрибутив добавлено актуальное ядро GNU/Linux. Основные доработки, выполненные подрядчиком, касались криптографической подсистемы и предконфигурации встроенных средств защиты информации.

В 2014 году ОС типового дистрибутива АИС ФССП России получила сертификат соответствия ФСТЭК России № 3119, подтверждающий, что операционная система «Гослинукс» имеет оценочный уровень доверия ОУДЗ и соответствуют требованиям руководящего документа Гостехкомиссии России по 4 уровню контроля отсутствия недеklarированных возможностей.

Список отечественных платформ GNU/Linux не ограничивается ГосЛинукс. На сегодняшний день это: AgiliaLinux, ALT Linux, Astra Linux, Calculate Linux, EduMandriva, InfraLinux, Linux XP, PuppyRus Linux, Rosa Linux, Runtu, Russian Fedora, ГосЛинукс, Мобильная система Вооружённых Сил, НауЛинукс. Не все дистрибутивы открыты для свободного доступа. Полностью закрытыми являются Linux XP, ГосЛинукс и Мобильная система Вооружённых Сил. Некоторые производители выпускают, как свободные релизы, так и

защищённые. Например, ООО «НТЦ ИТ РОСА» выпускает две версии ОС свободных и три сертифицированные версии — на продажу.

Что же изменилось за последние пять лет в масштабах страны? Если обратиться к статистике использования ОС на настольных системах (см. рис. 1), то увидим, что доля свободных платформ (на ядре Linux) почти не изменилась. Можно предположить, что компания GlobalStats не знает наши отечественные платформы и отнесла их к категории «Other». Однако, даже если рассматривать только данные по Linux (см. рис. 2), можно увидеть тенденции к росту доли этой ОС на настольных системах, причём рост этот нелинейный.

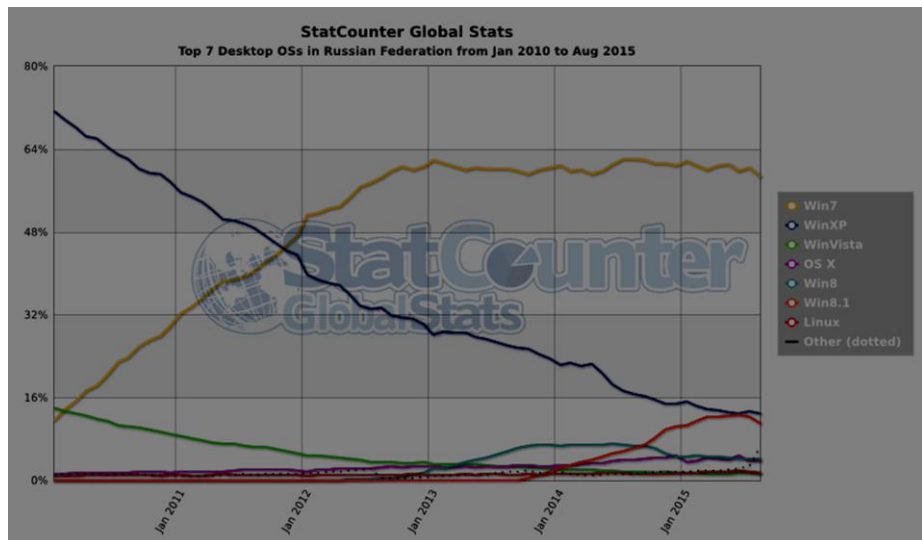


Рис. 1. Применение настольных ОС в России: динамика с 2010 г. по 2015 г.

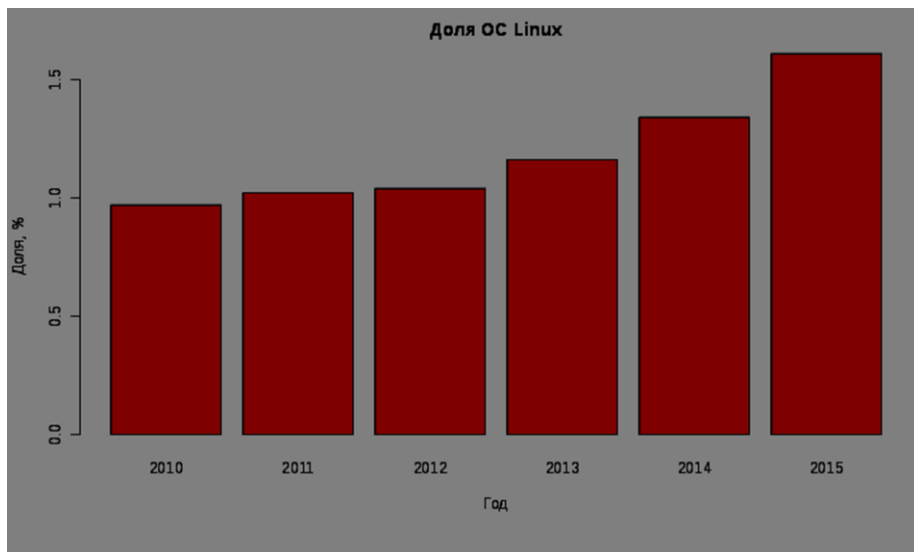


Рис. 2. Динамика роста доли ОС семейства Linux с 2010 г. по 2015 г.

Эксперты, работающие с госструктурами, с правительственными организациями, признают, что переход на СПО идёт с большим трудом. Зачастую, лица принимающие решение в отношении инфраструктуры саботируют правительственные решения и рекомендации. После запуска программы «Национальная программная платформа», во всех регионах России стали регулярно проходить мероприятия при поддержке правительственных и неправительственных организаций, имеющие, с одной стороны, просветительскую функцию, с другой стороны — функцию обратной связи, выполняющей мониторинг «ситуации на местах». В качестве примера такого мероприятия можно привести Форум «Электронная неделя на Алтае», в рамках которой проходит (уже) международная конференция «Информационные технологии в управлении: риски и возможности» [6].

Таким образом, в целом действия государства вселяют надежду, что в целях безопасности, независимости, поддержки отечественных производителей, свободное программное обеспечение будет иметь первостепенное значение в госструктурах. Это должно привести, с одной стороны, к необходимости подготовки ИТ специалистов и сотрудников органов государственного и муниципального управления для работы с СПО, с другой стороны, заставит коммерческие организации присмотреться к СПО и, возможно, что-то принять на вооружение.

Литература

1. Национальный стандарт Российской Федерации ГОСТ Р ИСО/МЭК 26300-2010 «Информационная технология. Формат Open Document для офисных

приложений (OpenDocument) v1.0» [Текст]. — Дата введения 1 июня 2011 г. (с правом досрочного применения).

2. ПРИКАЗ от 21 декабря 2010 г. N 800-ст ОБ УТВЕРЖДЕНИИ НАЦИОНАЛЬНОГО СТАНДАРТА [Электронный ресурс]. — М.: КонсультантПлюс, 2010. Режим доступа: <http://base.consultant.ru/cons/cgi/online.cgi?req=doc;base=EXP;n=498013>.
3. Технологическая платформа «Национальная программная платформа» [Электронный ресурс]. — М.: Технологическая платформа Национальная программная платформа, 2011. Режим доступа: <http://tp-npp.ru/>.
4. План перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения на 2011 — 2015 годы [Электронный ресурс]. — М.: Технологическая платформа Национальная программная платформа, 2011. Режим доступа: http://tp-npp.ru/index.php?option=com_content&view=article&id=102:-2011-2015-&catid=31:acts&Itemid=18.
5. ОС «Гослинукс» доступна для использования во всех ведомствах России [Электронный ресурс]. — М.: PC Week Live, 2015. Режим доступа: <http://www.pcweek.ru/foss/news-company/detail.php?ID=177035>.
6. Форум «Электронная неделя на Алтае» 2015, международная конференция «Информационные технологии в управлении: риски и возможности» [Электронный ресурс]. — Барнаул: Технологическая платформа Национальная программная платформа, 2015. Режим доступа: <http://itu2015.rf-seminar.ru/>.

ПРОГРАММНЫЙ КОМПЛЕКС WRF

А.К. Букина

Программный комплекс WRF (The Weather Research and Forecasting Model) — это система атмосферного моделирования, применяемая для метеорологических исследований и численного построения краткосрочных и среднесрочных прогнозов погоды [1-3]. Основные работы по созданию и поддержке текущих версий модели проводятся в Национальном центре атмосферных исследований США (the National Center for Atmospheric Research (NCAR)) [4], Национальном центре исследований окружающей среды США (The National Centers for Environmental Prediction (NCEP)) [5], и др.

Комплекс WRF поддерживает более 10 атмосферных моделей и может работать на различных вычислительных платформах. На сегодняшний день WRF используется в широком спектре метеорологических приложений на масштабах моделирования от десятков метров до нескольких тысяч километров.

Целью данной работы является представление основных сведений о программном комплексе WRF, а также проведение тестовых расчетов приземной температуры воздуха, сопоставление с данными разработчиков.

Программный комплекс WRF состоит из двух основных подсистем [1-3]: подсистемы препроцессинга WPS (WRF Preprocessing System) и подсистемы моделирования (см. рисунок 1).

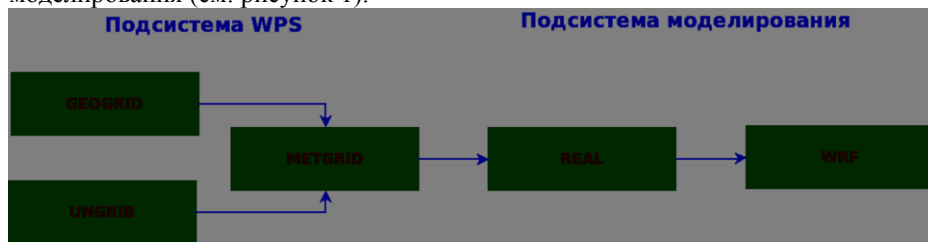


Рисунок 1. Блок-схема основных модулей программного комплекса WRF [1]

Подсистема WPS содержит три основных модуля, позволяющих сконфигурировать модель и подготовить входные данные, которые будут использоваться для задания начальных и граничных условий при моделировании.

1. Модуль GEOGRID предназначен для задания границ области моделирования, интерполяции и привязки глобальных географических данных к узлам сетки модели.

1. Модуль UNGRID служит для распаковки файлов, содержащих климатические и метеорологические данные, из специализированного формата GRIB в промежуточный (вспомогательный) формат, принятый в модели WRF. Какие именно климатические поля будут использоваться при моделировании описывается в конфигурационном файле Vtable. Все поддерживаемые моделью внешние архивы данных описаны в конфигурационных файлах, хранящихся в директории

WPS/ungrib/Variable_Tables/.

Модуль METGRID предназначен для интерполяции климатических данных, полученных после распаковки, на узлы горизонтальной сетки модели.

На завершающем этапе перед началом моделирования проводится интерполяция климатических данных, полученных на выходе подсистемы WPS, на вертикальные уровни, число которых задается в файле конфигурации модели. Делается это с помощью модуля REAL подсистемы моделирования. После этого производится запуск модели WRF.

Для проверки правильности сборки кода и корректности выдаваемых результатов разработчиками рекомендуется провести расчет с использованием тестовых данных, поставляемых вместе с пакетом [6]. Тестовые данные представляют собой архив метеорологических данных за 24-25 января 2000 года для восточного побережья США.

Моделирование проводится для области размером 61x74 ячейки с пространственным разрешением 30 км. Координаты области моделирования $\sim(28-44)^\circ$ с.ш. $\sim(95-70)^\circ$ з.д. (см. рисунок 2). В расчетах используется равноугольная коническая проекция Ламберта.

Результаты моделирования приземной температуры воздуха представлены на рисунке 3 и в таблице 1. Видно, что максимальное отклонение среднесуточных значений температуры, полученных в данной работе от оригинальных результатов разработчиков модели [6] не превышает по абсолютной величине 0.3°C . Выявленная разница значений может быть обусловлена некоторыми отличиями в конфигурации модели, используемой автором, от конфигурации разработчиков WRF.

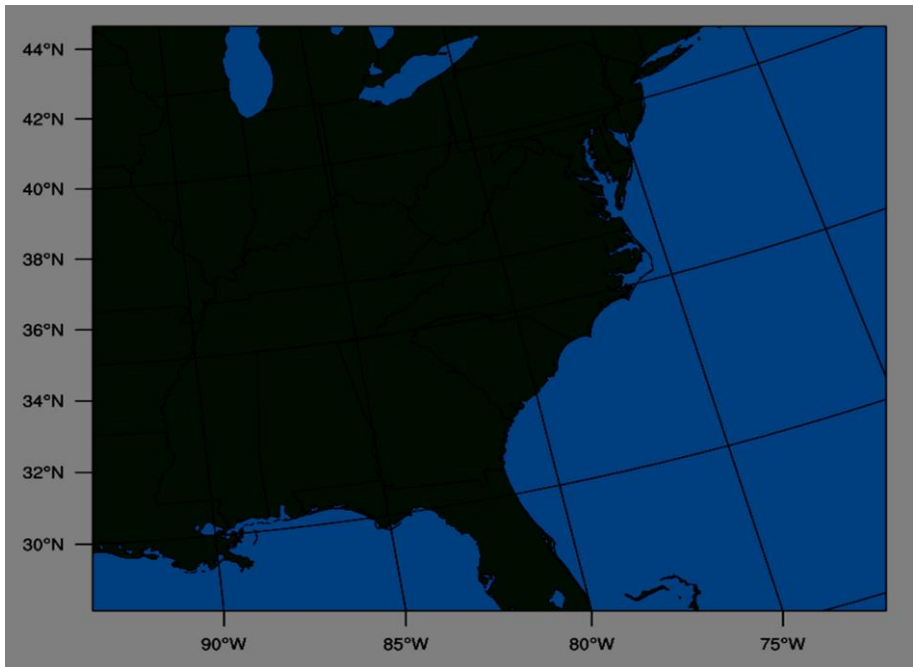


Рисунок 2. Координаты и положение области моделирования

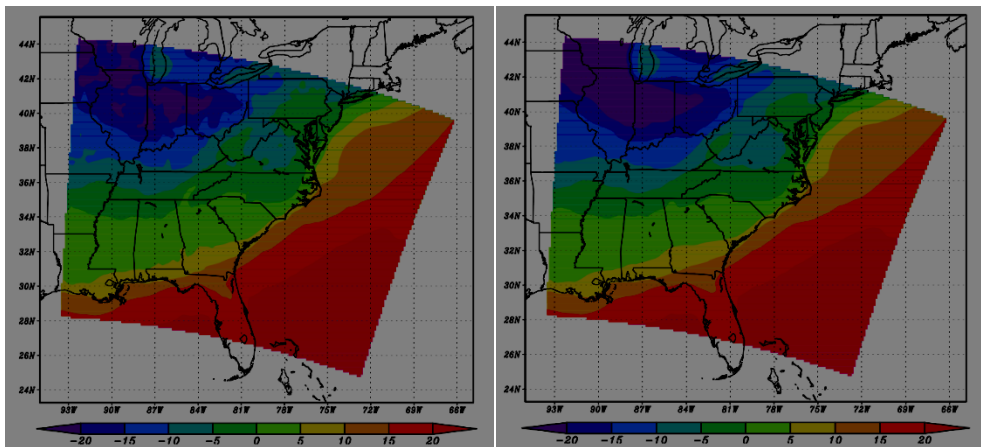


Рисунок 3. Сопоставление средней приземной температуры воздуха, полученной в данной работе (слева) с тестовым результатом разработчиков модели WRF [6] (справа)

Таблица 1.

Сопоставление среднесуточной приземной температуры воздуха (°C) для первых

42 ячеек области моделирования с оригинальными данными авторов пакета WRF [6]

Ячейка	Оригинальный результат	Тестовый расчет	Разность
1	15,98	15,97	0,01
2	4,10	4,23	-0,13
3	1,67	1,79	-0,12
4	-8,94	- 8,78	-0,15
5	-12,26	-12,30	0,04
6	-20,64	-20,78	0,13
7	16,39	16,40	0,00
8	5,36	5,15	0,20
9	1,10	1,14	-0,04
10	-5,74	-5,72	-0,02
11	-13,44	-13,31	-0,13
12	-24,60	-24,90	0,30
13	20,14	20,11	0,04
14	9,31	9,32	-0,01
15	2,02	1,46	0,27
16	-6,36	-6,38	0,01
17	-14,92	-14,85	-0,08
18	-20,68	-20,72	0,04
19	22,42	22,47	-0,05
20	16,42	16,54	-0,12
21	4,19	4,21	-0,02
22	-2,68	-2,79	0,11
23	-4,47	-4,23	-0,24
24	-17,41	-17,34	-0,06
25	15,61	15,38	0,23
26	16,83	16,67	0,16
27	11,05	11,15	-0,09
28	0,26	0,31	-0,05
29	-6,04	-5,93	-0,11
30	-4,40	-4,01	-0,28
31	21,39	21,34	0,05
32	22,92	22,93	0,00
33	19,05	19,08	-0,03
34	13,48	13,52	-0,04
35	6,42	6,60	-0,18
36	0,71	0,65	0,06
37	21,77	21,75	0,02
38	22,65	22,65	0,00
39	22,28	22,30	-0,03
40	18,43	18,42	0,01

41	11,98	11,92	0,05
42	12,76	12,87	-0,11

Главным результатом работы является вывод о правильности сборки и установки комплекса WRF и о возможности его использования при моделировании погоды для различных регионов, в том числе для территории Алтайского края.

Литература

1. Wang W., Bruyère C., Duda M. Et al. ARW Version 3 Modeling System User's Guide [Electronic resource]. URL: http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html
2. Official site the Weather Research and Forecasting (WRF) Model [Electronic resource]. URL: <http://www.wrf-model.org/index.php>
3. WRF-ARW tutorials [Electronic resource]. URL: <http://www2.mmm.ucar.edu/wrf/OnLineTutorial/index.htm>
4. The National Center for Atmospheric Research (NCAR) [Electronic resource]. URL: <http://ncar.ucar.edu/>
5. The National Centers for Environmental Prediction (NCEP) [Electronic resource]. URL: <http://www.ncep.noaa.gov/>
6. WRF - the tutorial case study data [Electronic resource]. URL: http://www2.mmm.ucar.edu/wrf/TUTORIAL_DATA/JAN00_AWIP.tar.gz

MONDRIAN И SAIKU КАК СРЕДСТВО АНАЛИЗА БОЛЬШИХ МАССИВОВ ДАННЫХ

А.А. Барбышев

Большие данные — это серия подходов, методов обработки данных огромных объемов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях постоянного прироста.

В качестве определяющих характеристик для больших данных отмечают «три V»:

- Объем (volume) — в смысле величины физического объема.
- Скорость (velocity) — в смыслах как скорости прироста, так и необходимости высокоскоростной обработки и получения результатов.
- Многообразие (variety) — в смысле возможности одновременной обработки различных типов структурированных и неструктурированных данных.

Основное отличие больших данных от высоконагруженных систем состоит в возможности строить гибкие запросы.

В качестве источников возникновения больших данных могут быть поступающие данные с измерительных приборов, потоки из социальных сетей, метеорологические данные и другое [1].

Программное обеспечение

На рынке ПО существует множество реализаций так называемых OLAP-серверов (аналитическая обработка в реальном времени, позволяющая анализировать большие массивы данных), распространяемых как на платной основе, так и бесплатно.

В числе наиболее распространенных можно выделить следующие:

- На коммерческой основе:
 - Microsoft Analysis Services [2]
 - Hyperion Essbase [3]
 - IBM Cognos TM1 [4]
- На основе open sources:
 - Mondrian [5]
 - Palo [6]

OLAP сервер	MOLAP	ROLAP	HOLAP
Microsoft Analysis Services	Да	Да	Да
Essbase	Да	Да	Да
TM1	Да	Нет	Нет
Mondrian	Нет	Да	Нет
Palo	Да	Нет	Нет

Таблица 1 Сравнение OLAP-серверов по модели хранения данных.

Mondrian

Основные возможности:

- Поддерживает язык запросов MDX (а также спецификации XML for Analysis и JOLAP (JSR-69))
- Для хранения данных может использовать любую СУБД с поддержкой SQL
- Может кешировать в памяти суммарные результаты

Mondrian поддерживает любые jdbc-источники данных. Заявлена поддержка SQL-серверов MySQL, Oracle Database, PostgreSQL, Microsoft SQL Server, хранилищ Greenplum и Infobright.

OLAP реализация

ПО Mondrian распространяется в виде исполняемых файлов или в виде пакетов с исходным кодом для операционных систем Linux и Windows.

Перед началом работы с OLAP-сервером Mondrian, необходимо выполнить несколько действий.

Для начала, нужно запустить одну из СУБД, с которым может работать Mondrian, например MySQL, Microsoft SQL Server или PostgreSQL.

ПО Mondrian состоит из специализированных java-файлов, для работы которых необходимо использовать Apache Tomcat.

Далее необходимо создать XML-схему, описывающую кубы. В схеме указываются: имя, размерности и меры куба; а также таблицы базы данных откуда информация должна загрузиться в куб.

Использование OLAP-клиента Saiku

Данный OLAP-клиент распространяется на основе open source, имеет приятный интерфейс, русскоязычную локализацию, а также позволяет быстро получить некоторую информацию из куба путем перетаскивания (Drag & Drop) нужных измерений и мер в поля для строк и столбцов. Работа клиента осуществляется в веб-браузере. Интерфейс написан на HTML, CSS и JavaScript, что позволяет легко изменять или полностью заменить интерфейс пользователя.

Анализ данных

Для практического изучения возможностей технологии OLAP воспользуемся реализацией, основанной на программном обеспечении Mondrian. В качестве СУБД выберем MySQL, в которую занесена следующая информация:

Таблица статистики, содержащая столбцы: исходные IP-адрес и порт, IP-адрес и порт назначения, дата и часы (0-24).

Таблица дат, она содержит в одном столбце все присутствующие в статистике дни, а в двух других столбцах — их ассоциацию с месяцами и годами. Такой подход необходим для того, чтобы создать измерение для дат с

иерархией Год-Месяц-День.

На основе этих данных в кубе формируются шесть измерений: дата, час, исходные IP-адрес и порт, IP-адрес и порт назначения.

Рассмотрим несколько примеров для анализа данных:

1. С использованием технологии перемещения Drag & Drop выведем отчет по количеству трафика за каждый месяц.
2. С помощью операции детализации (Drill Through), максимально детализируем выбранную ячейку.
3. Составим запрос с использованием редактора MDX кода. При этом узнаем количество трафика для определенного IP-адреса.

Пример 1

Нам необходимо заполнить поля «Колонки» и «Строки» из главной рабочей области клиента. Для этого из измерения даты перетянем иерархию «месяц», а из мер перетянем единственную меру «Traff» (Траффик). Когда поля «Колонки» и «Строки» будут заполнены хотя бы одним измерением или мерой, то клиент автоматически начнет построение отчета.

Year	Month	Traff
2012	December 2012	21 203 212 129 264
	November 2012	21 045 667 913 418
2013	April 2013	21 693 664 704 515
	August 2013	14 529 807 318 049
	February 2013	21 194 239 063 802
	January 2013	16 869 358 814 799
	July 2013	14 090 416 524 639
	June 2013	17 490 410 809 097
	March 2013	21 636 447 481 983
	May 2013	21 347 602 775 074

Рис 1 Отчет количества трафика за каждый месяц

Пример 2

Детализация по ячейке (англ. Drill Through) — это средство извлечения дополнительных к хранящимся в OLAP сервере данных.

Покажем на примере наших данных. В таблицу выведена информация о дне в месяце и количестве информации переданной и полученной в этот день. С помощью операции Drill Through можно, например, вывести весь трафик, потребленный каждым днем очень подробно, вплоть до каждого соединения.

src_addr	src_port	dest_addr	dest_port	bytes
0.0.0.0	0	192.168.99.174	0	48
0.0.0.0	0	192.168.99.187	0	48
0.0.0.0	0	192.168.99.222	0	48
0.0.0.0	0	192.168.99.240	0	48
0.0.0.0	0	192.168.99.249	0	48
0.0.0.0	0	255.255.255.255	0	280194
0.0.0.0	0	255.255.255.255	0	245832
0.0.0.0	0	255.255.255.255	0	269927
0.0.0.0	0	255.255.255.255	0	252257
0.0.0.0	0	255.255.255.255	0	255761
0.0.0.0	0	255.255.255.255	0	257830

Рис 2 Результат работы средства Drill Through

Пример 3

Для того, чтобы написать запрос на языке MDX, в меню нажмем на кнопку перехода в режим редактирования MDX. В этом случае рабочая область клиента изменится на редактор кода.

Например, выведем информацию о трафике, посланный с IP-адреса 66.249.72.19

Такой запрос на языке MDX будет выглядеть так:

```
SELECT
{[Measures].[Traff]} ON COLUMNS,
{Hierarchize({[Date.YMD].[Month].Members}, {[Date.YMD].[Day].Members})}
ON ROWS
FROM [Inet stat] where ([src IP].[66.249.72.19])
```

В первых двух строках указывается какие данные будут отображены в столбце и строке. Далее указывается имя куба и параметр, в котором выделим нужный IP-адрес (так как нам нужен конкретный адрес).

Month	Day	Traff
July 2013		12 443 696
	2013-07-01	12 443 096
	2013-07-02	600

Рис 3 Результат MDX запроса

Итог

Рассмотренное открытое программное обеспечение имеет большое количество различных функций, которые не уступают таковым у коммерческих

аналогов.

Литература

1. Большие данные. [Электронный ресурс] — Заглавие с экрана. Режим доступа: https://ru.wikipedia.org/wiki/Большие_данные
2. Business Intelligence | Database Management | Data Warehousing | Microsoft SQL Server [Электронный ресурс] — Заглавие с экрана. Режим доступа: <http://www.microsoft.com/sql/technologies/analysis/default.msp>
3. Oracle Hyperion Essbase. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <http://www.hyperion.ru/products/intgr/dwhs/>
4. IBM - Cognos TM1 - Russia [Электронный ресурс] — Заглавие с экрана. Режим доступа: <http://www-03.ibm.com/software/products/ru/ru/cognostm1/>
5. Mondrian. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <http://community.pentaho.com/projects/mondrian/>
6. palo.net - Open Source Business Intelligence [Электронный ресурс] - Заглавие с экрана. Режим доступа: <http://www.palo.net>

ПРИМЕНЕНИЕ R И RSTUDIO ДЛЯ АНАЛИЗА ПОЛИТИЧЕСКОГО ИМИДЖА ГОСУДАРСТВ

А. С. Большаков

Данная работа выполнялась в рамках многоцентрового исследования по аналитике больших данных, которое организовано по инициативе Академического партнерства ЕМС в России и СНГ.

В рамках этого исследования решалось несколько задач большим коллективом, а именно представителями Московского Государственного Университета (под руководством декана факультета глобальных исследований Ильина Ильи Вячеславовича), Петербургского государственного университета путей сообщения Императора Александра I (под руководством зам. зав. кафедрой «Информационные и вычислительные системы» Дашонка Виктора Леонидовича), Владивостокского государственного университета экономики и сервиса (под руководством заведующего кафедрой математики и моделирования Мазелиса Льва Соломоновича), Алтайской Академии Экономики и Права (группа студентов 4-го курса направления «прикладная информатика в экономике» и группа студентов 3-го курса направления «бизнес-информатика» под руководством доцента кафедры математики и информационных технологий Журенкова Олега Викторовича). В центре внимания многоцентрового исследования — использование неструктурированных текстовых данных масштаба Big Data (WWW) для поиска подходов в прогностической оценке внутренней политической и экономической ситуации в государствах.

В рамках этого исследования нашей группе была поставлена задача: разработать методику сбора и анализа данных, полученных из открытых источников World Wide Web. Наша группа занималась сбором и анализом данных. Исходными данными являлось название страны и ключевые слова (такие как терроризм, оккупация, демократия и прочее). Инструментом поиска был выбран сервис Яндекса: Поиск по блогам. Выбор данного инструмента обоснован возможностью задания временного отрезка в поиске. Поиск ключевых слов производился сразу на двух языках: английском и русском. Количество статей в популярных соцсетях, удовлетворяющих условию, заносилась в таблицу.

После сбора всех данных необходимо было визуализировать полученные результаты. Для этой задачи был выбран язык R и среда разработки Rstudio.

R — язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом. R широко используется как статистическое программное обеспечение для анализа данных.

R доступен под лицензией GNU GPL. Распространяется в виде исходных кодов, а также откомпилированных приложений под ряд операционных систем: FreeBSD, Solaris и другие дистрибутивы Unix и Linux, Microsoft Windows, Mac OS X.

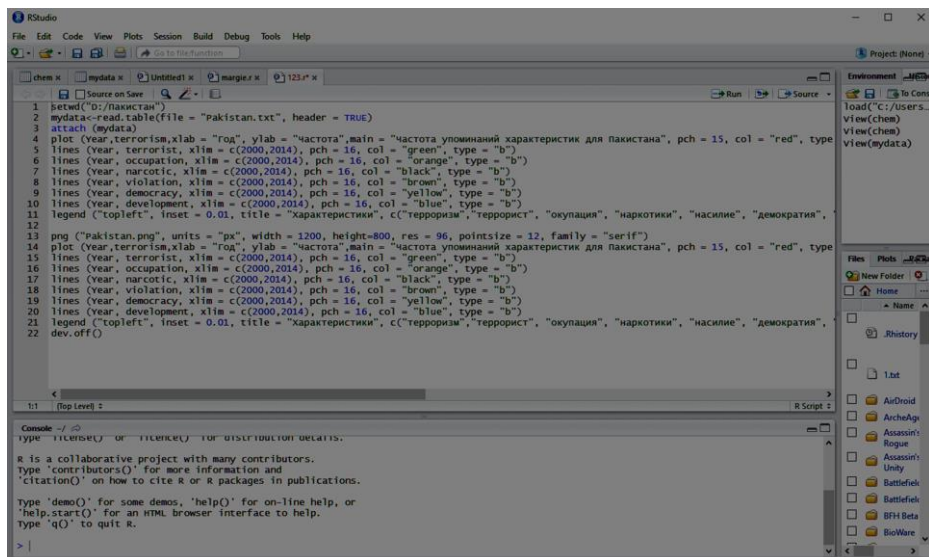


Рис. 1. Интерфейс RStudio

В R используется интерфейс командной строки, хотя доступны и несколько графических интерфейсов пользователя, например пакет RStudio (см. рис. 1).

RStudio — среда разработки приложений для R, тоже свободно распространяемое ПО, с открытым исходным кодом. RStudio доступна в двух версиях: RStudio Desktop, в которой программа выполняется на локальной машине как обычное приложение; и RStudio Server, в которой предоставляется доступ через браузер к RStudio установленной на удаленном Linux-сервере. Дистрибутивы RStudio Desktop доступны для Linux, OS X и Windows. RStudio написана на C++ с использованием фреймворка Qt.

Для визуализации полученных результатов был написан код, выполняющий построение графиков по полученным результатам.

Каждый участник нашей команды собирал данные в Интернете и выполнял построение графиков по определенной стране. Таким образом были собраны первичные данные и построены графики «Частота упоминаний характеристик для страны» (см. рис. 2).

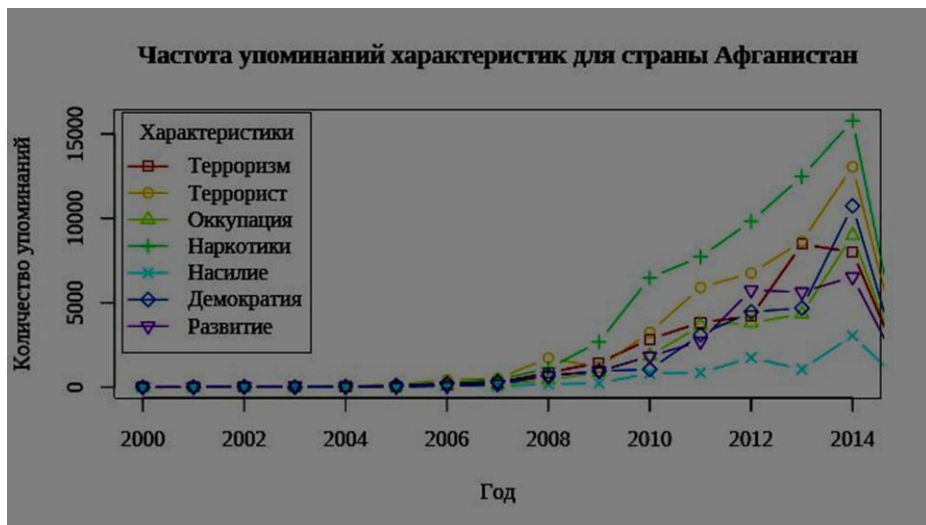


Рис. 2. Визуализация частоты упоминаний характеристик для стран

В индивидуальной работе использовалась система контроля версий Git (тоже с лицензией GNU GPL). Для сбора всех результатов был использован web-сервис GitHub (см. рис. 3). Этот сервис абсолютно бесплатен для проектов с открытым исходным кодом. Слоган сервиса — «Social Coding» — можно перевести как «Пишем код вместе».

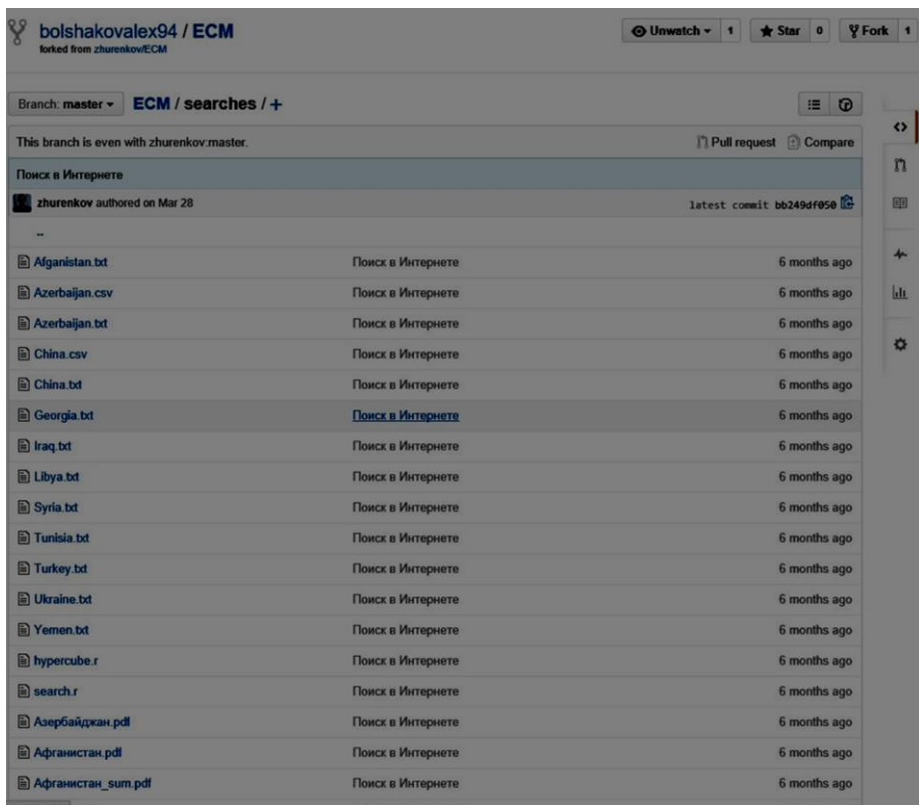


Рис. 3. Репозиторий в GitHub с результатами исследования

Для дальнейших исследований был построен гиперкуб с измерениями страна, характеристика, год. При консолидации данных в гиперкуб был добавлен агрегат — суммарное значениями фактов за все годы. Далее, из этого гиперкуба построены гистограммы по агрегатным значениям для измерений «ключевое слово» (по всем странам) и «страна» (по всем характеристикам). В результате получены наборы, вида рис. 4, 5.

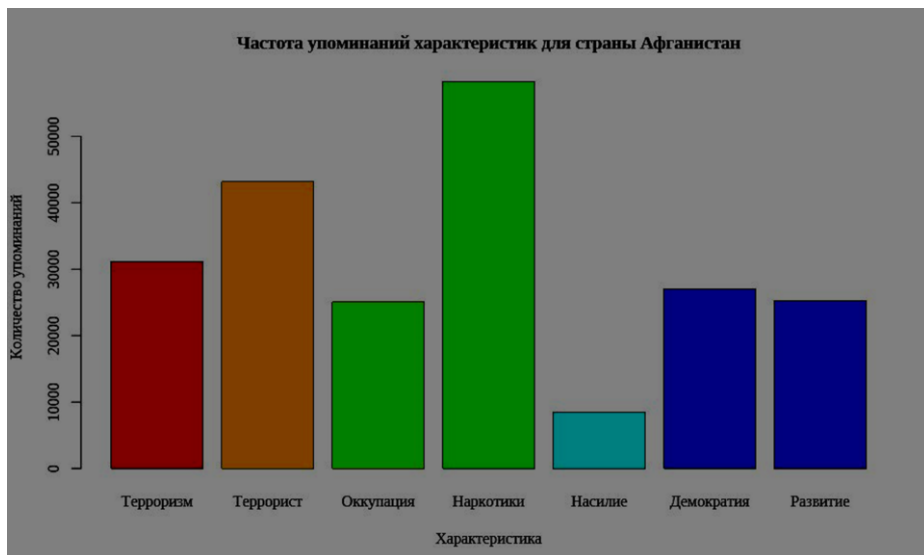


Рис. 4. Визуализация суммарной частота упоминаний характеристик для отдельной страны

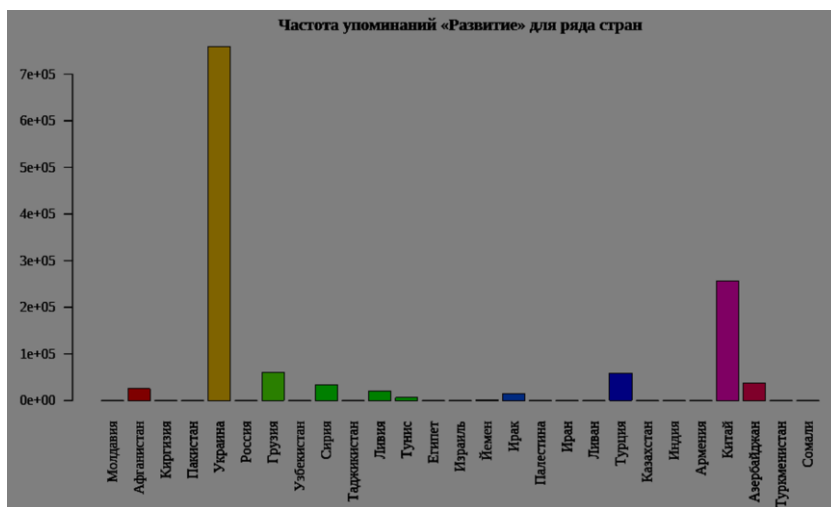


Рис. 5. Визуализация суммарной частота упоминаний отдельной характеристики для всех стран

В результате проделанной работы была разработана методика сбора и

анализа данных из открытых источников WWW. Результаты наших исследований, полученных по этой методике вошли в публикацию [1].

Начатое исследование будет продолжено. С первыми результатами можно ознакомиться также на опубликованных слайдах на SlideShare в русской и английской версии.

Литература

1. Новые возможности для понимания глобальных процессов с применением аналитики BIG DATA (результаты многоцентрового исследования) [Текст] / О. Ю. Колесниченко, Г. Н. Смородин, О. В. Журенков и др. // Национальная ассоциация ученых (НАУ). — 2015. — № 6 (11). — С. 165–171.

РАЗРАБОТКА СПЕЦИАЛИЗИРОВАННОГО WEB-ИНТЕРФЕЙСА СБОРА И АНАЛИЗА ДАННЫХ

С. С. Гарбуз

На сегодняшний день, актуально является задача разработки свободного программного обеспечения. Целью данного проекта является создание системы сбора и обработки данных, предназначенных для нейросетевого исследования. В силу обеспечения распределенного сбора данных, систему было решено реализовать в виде web-приложения. Это позволит обеспечить распределенную работу анализа данных. В качестве инструмента были выбраны языки программирования PHP, HTML и JavaScript, и СУБД MySQL [1].

Разрабатываемая система должна позволять создавать новые таблицы, заполнять созданные таблицы в ручном режиме, либо с помощью импорта файла. Самая важная функция данной системы - обработка уже созданных таблиц. Для этого должен быть реализован блок управления таблицами, который будет позволять находить пустые поля, повторяющиеся строки или столбцы, столбцы, имеющие одинаковые значения в полях, экстремальные выбросы значений полей, относительно среднего значения. Также должна иметься возможность изменения значений полей вручную.

WEB-интерфейс содержит 4 основных блока: блок создания таблиц, блок заполнения таблиц, блок управления таблицами, блок работы с учетными записями пользователей.

Блок создания таблиц представляет собой интерфейс для выбора параметров создаваемой таблицы, таких как: название таблицы, краткое описание таблицы, выбор количества полей, названия полей, выбор характера каждого поля, выбор типа каждого поля, выбор количества возможных значений каждого качественного поля, возможные значения каждого качественного поля.

При создании таблицы нужно учитывать, что названия таблицы и полей должны соответствовать правилам MySQL для выбора имен.

Характер поля указывает значение поля в дальнейшем нейросетевом моделировании: входное или выходное поле.

Тип поля выбирается в зависимости от того, какое значение предполагается хранить в данном поле. Предусматривается два типа полей: количественное и качественное. При выборе количественного типа пользователю предлагается ввести значение вручную, а при выборе качественного – предлагается выбор из списка возможных значений.

Блок заполнения таблиц представляет собой интерфейс для заполнения уже созданных таблиц. Пользователь сначала выбирает таблицу, в которую необходимо добавить записи, после чего предлагается выбрать способ заполнения. Возможны два варианта:

- Ручное заполнение, при котором в таблицу добавляется каждая строка отдельно. Пользователь вручную заполняет все значения.
- Импорт из файла.

Блок управления таблицами представляет собой интерфейс для управления созданными таблицами, содержащий следующий функционал: удаление строк, удаление столбцов, удаление таблиц, маркировка строк с пустыми полями, маркировка столбцов с пустыми полями, маркировка повторяющихся строк; маркировка повторяющихся столбцов, маркировка столбцов имеющих одинаковые значения в полях, маркировка столбцов имеющих сильные выбросы значений полей относительно среднего значения, позиционирование на маркированные поля таблиц, редактирование полей таблиц, экспорт таблиц.

Блок работы с учетными записями пользователей представляет собой интерфейс для создания учетных записей пользователей и управления ими.

Все пользователи системы разделены на ранги:

- Администратор имеет полный доступ к системе. Он может создавать и удалять пользователей, изменять их права, создавать и удалять таблицы, управлять доступностью таблиц для заполнения, заполнять созданные таблицы, проводить проверку данных и редактировать их, экспортировать данные в CSV-файл.
- Модератор имеет права только для работы с созданными таблицами: заполнение таблиц, проверка и редактирование данных.
- Пользователь может только заполнять уже созданные таблицы.
- Гостем является любой пользователь не авторизованный в системе. Ему лишь предлагается зарегистрироваться или войти в систему.

В проекте используются две базы данных: служебная и пользовательская. В служебной базе данных хранится информация о пользователях и созданных таблицах. Служебная база данных хранит в себе всю служебную информацию. Она состоит из 4 таблиц: «таблицы» (tables), «поля» (fields), «состояния» (states) и «пользователи» (users). В таблице «пользователи» содержатся идентификатор пользователя, его имя, ранг, логин и пароль для входа в систему. Таблица «таблицы» содержит в себе идентификатор, имя и метку о доступности таблицы для заполнения. Таблица «поля» содержит идентификатор поля, его имя, тип, характер и идентификатор таблицы, к которой относится данное поле. Таблица «состояния» содержит идентификатор состояния, значение состояния и идентификатор поля, к которому оно относится. Пользовательская база данных хранит в себе создаваемые таблицы.

При создании таблиц администратор выбирает название таблицы, определяет количество полей и свойства каждого поля. В свойствах указывается имя поля, его характер (вход или выход), тип (качественное или количественное). Если тип поля качественный, то перечисляются все возможные состояния этого поля. При сохранении вся информация о таблице заносится в служебную базу данных, а в пользовательской базе создается соответствующая таблица. По-умолчанию все таблицы создаются скрытыми от всех пользователей кроме администратора. Для того, чтобы открыть доступ остальным пользователям, необходимо в разделе управления таблицами поставить соответствующую отметку.

К разделу управления таблицами доступ имеет только администратор. В этом разделе он может управлять доступом к таблицам, удалять таблицы и экспортировать их в CSV-файл.

Пользователь может только заполнять уже созданные таблицы. Он выбирает таблицу, в которую нужно добавить записи, после чего указывает способ заполнения.

Если выбирается ручное заполнение таблицы, то открывается форма со всеми полями, пользователь вручную заполняет все поля. После чего необходимо нажать кнопку «Добавить запись».

Если выбрано заполнение из файла, то пользователю необходимо выбрать файл с данными в формате CSV. При этом он должен убедиться, что структура таблицы в файле полностью совпадает со структурой таблицы в базе данных. После чего необходимо нажать кнопку «Импорт».

Для управления таблицей пользователю сначала необходимо выбрать из списка таблицу. Дальнейшие действия предполагают, что таблица уже выбрана. При анализе данных пользователю выдается сводная информация о найденных ошибках в выбранной информационной таблице. Данная информация содержит ссылки для каждого ошибочного поля, по нажатию на которые курсор позиционируется на соответствующей ячейке таблицы. Это очень облегчает анализ больших таблиц.

Для редактирования полей таблицы необходимо сделать двойной клик по полю, которое необходимо изменить. После этого ячейка перейдет в режим редактирования. Для удаления строк или столбцов необходимо отметить один или несколько строк (столбцов), которые нужно удалить, и нажать кнопку «Удалить столбцы».

Таким образом, созданное свободное программное обеспечение позволит оперативно осуществлять нейросетевое моделирование при решении задач, не имеющих аналитического решения.

Литература

1. Бейли Л., Моррисон М. Изучаем PHP и MySQL : Эксмо, - 2010. - 707с.
2. Нейроинформационные технологии: учебное пособие / А.А. Шайдунов. – издательство Алтайского государственного университета, 2014. – 138с.

ФАКС В VOIP-СЕТЯХ.

Д.А. Дмитриев

С появлением и последующем развитием электронного документа оборота стала очевидно проявлять проблема обмена документами между пользователями, использующими современные метода обмена информацией и пользователями, которые используют традиционные способы передачи документов. К современным методам обмена информации можно отнести такие способы передачи информации как электронная почта различные системы электронного документа оборота. В большинстве случаев в роле традиционного метода передачи документов выступает факсимильная связь(факс). Факсимильная связь использует для передачи информации телефонные сети. Но с появлением компьютерных сетей стало возможным применение систем IP-телефонии, которые являются более дешёвым видом связи. В связи с этим начались проявляться проблемы передачи факсов из обычной телефонной сети в VoIP сети.

В большинстве случаев решением проблемы передачи факсов из обычной телефонной сети в VoIP сети является закупка дополнительного оборудования, и затраты на обучение персонала для настройки этого оборудования. Зачастую многие компании ограничены в бюджете при закупки дополнительного VoIP оборудования. В связи с этим возникает задача разработки и внедрения собственной программной системы, которая позволяла производить отправку и принятия факса с имеющимся в организации сетевым оборудованием без дополнительного финансового вложения. Разработанной программной системе факс передаётся через cisco colmenedger на сервер IP телефонии на котором происходит обработка пришедшего факса. На сервере также разработано специальное веб приложение, которое позволяет просматривать факс. Для создания системы было принято решения использовать открытый программный сервер телефонии Asterisk, свободная система управления базами данных MySQL и язык программирования PHP. Asterisk свободный программный сервер с открытым исходным кодом. Разработан компанией Digium на его основе можно создать классическую АТС, Asterisk поддерживает множество VoIP протоколов.

Существует возможность создания собственной функциональности с использованием собственного языка Asterisk для написания диалплана. Asterisk распространяется по условиям двойной лицензии, благодаря которой одновременно с основным кодом, распространяемым по открытой лицензии, возможно создание закрытых модулей, содержащих лицензируемый код. Работает на операционных системах GNU/Linux не требует никакого специального оборудования. Для того чтобы Asterisk имел возможность приема и отправки факса необходима дополнительная библиотека SpanDSP. В стандартной сборке данная библиотека отсутствует и ее необходимо устанавливать дополнительно. SpanDSP это специализированная библиотека, содержащая весь необходимый функционал для приема и передачи факса,

позволяет преподавать данные со скоростью 14400бит/с. Также необходимо установить приложения `app_gxfax` и `app_txfax` это два приложения позволяющие отправлять и принимать факс. После сборки всех необходимых компонентов в Asterisk появятся две новые команды `gxFax` и `txFax` принимающее одно значение — это имя файла. Для передачи факса необходим файл в формате Tiff. Преобразования PDF файла в формат Tiff было реализовано в WEB-интерфейсе.

Для удобства взаимодействия пользователей с сервером Asterisk был разработан WEB-интерфейс, позволяющий пользователям без длительного обучения производить отправку и прием факса. Интерфейс разделен на две области — это пользовательский интерфейс и интерфейс для администраторов системы. Пользовательский интерфейс отображает принятые и отправленные факсы и дает возможность загрузки документа на сервер для последующей отправки. Также пользователь может управлять имеющимися у него документами. Интерфейс позволяет пользователям сохранять документы для последующего просмотра передавать документ на повторную отправку как из области принятых так из области отправленных факсов. Для удобства пользователей все документам присваивает дата и время позволяющая пользователям ориентироваться, когда был принят или отправлен то или иной документ. Во избежание как-либо хакерских атак и отсеивания случайных пользователей для входа в интерфейс требует логин и пароль выдаваемый администраторами системы. У пользователя имеется возможность изменения своих контактных данных которые предоставляются ими при регистрации в системе.

Разработанная система дает возможность без дополнительных финансовых вложений реализовать передачу факса в VoIP сетях. Также можно отметить возможность быстрого развертывания сервиса. Такие системы должна быть развернута во всех малых и крупных организациях.

ПРОЕКТИРОВАНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОМОЩИ СТУДЕНТУ В ТРУДОУСТРОЙСТВЕ

А.А. Дружинина

Вопрос о трудоустройстве после обучения в ВУЗе является предметом размышления у всех студентов. В каких дисциплинах можно получить наиболее глубокие знания - одна из основных проблем, стоящих перед абитуриентами.

Любая должность имеет свои собственные требования, которые должны быть удовлетворены. Студент, в свою очередь, должен обладать определенными профессиональными навыками, умениями, соответствовать требованиям, которые выдвигает работодатель.

С учетом выше сказанного возникает необходимость создания свободного программного обеспечения, предоставляющего помощь студенту при выборе будущей профессии.

В настоящее время существуют разные системы содействия трудоустройству выпускников. Например «автоматизированная система трудоустройства выпускников» (АИСТ), которая позволяет:

- работодателям осуществлять поиск соискателей на замещение вакантных должностей среди студентов и выпускников учреждений профессионального образования всех субъектов Российской Федерации;
- студентам и выпускникам расширять параметры поиска работы во всех субъектах Российской Федерации соответственно полученным знаниям, устремлениям и карьерным амбициям;
- органам исполнительной власти, имеющим в своем ведении образовательные учреждения и осуществляющим управление в сфере образования оперативно принимать управленческие решения по различным направлениям деятельности сферы образования, в том числе связанным с приведением объемов и профилей подготовки квалифицированных кадров в соответствии с потребностями рынка труда субъектов Российской Федерации [1,2].

Однако данная система не может дать студенту исчерпывающую информацию о тех дисциплинах, которые ему необходимо глубоко изучать, для того, чтобы занимать желаемую должность.

Таким образом, актуальной является задача создания системы, предоставляющей помощь студенту при выборе будущей должности (профессии). Такая система позволит студенту исходя из должности, которую в будущем он хочет занимать, определить те дисциплины, на которые стоит обратить наибольшее внимание, для того чтобы соответствовать компетенциям, необходимым при исполнении обязанностей. Так же, данная информационная система может применяться в ВУЗах заведующими выпускающих кафедр для учета оптимизации учебного процесса, в связи с требованиями работодателей.

Принцип работы, такой системы, заключается в том, что на этапе наполнения информационной системы, на вход программы поступают данные в виде списка дисциплин. Причем каждой дисциплине соответствует

тематический план занятий и список компетенций для каждой темы. Каждая компетенция имеет описание. Так же на вход системы подаются должности, с соответствующими им требованиями, которые сохраняются в информационной базе. Назначение разрабатываемой системы заключается в формировании соответствий между:

1. изучаемыми дисциплинами и требованиями к должностям;
2. темами из тематического плана дисциплин и требованиями к должностям.

Соответствия формируются на основе выявления совпадений между требованиями к должностям, компетенциями дисциплин и тематическим планом дисциплин.

На начальном этапе работодатель загружает в информационную базу список должностей с их требованиями. Заведующий кафедрой загружает учебный план и рабочие программы дисциплин (РПД) того или иного направления (специальности).

В процессе работы системы, студент выбирает должность, которую он в дальнейшем планирует занимать, и система автоматически ведет поиск соответствующих дисциплин, указывая конкретные разделы учебной программы, интеллектуально сравнивая компетенции и требования с помощью тезауруса.

На выходе система выдает список рекомендованных дисциплин, необходимых для получения желаемой должности, ранжированный по значимости включенных в список дисциплин.

Таким образом, система должна выполнять следующие задачи:

- хранить данные в информационной базе: УП, РПД, должности и требования к ним, компетенции и их описание;
- выявлять соответствия посредством соотнесения слов по смыслу с помощью тезауруса;
- выводить темы занятий, полученные после проведения сравнения требований и компетенций.

Предложенную информационную систему было решено реализовать в виде интернет-приложения с использованием языка программирования PHP и СУБД MySQL [3, 4].

В процессе разработки и тестирования системы в качестве исходных данных были взяты учебный план (УП) подготовки бакалавров - направление подготовки «230100.62 Информатика и вычислительная техника» и магистров – направление подготовки «230100.68 Информатика и вычислительная техника». В плане указаны все дисциплины, которые должны изучить студенты за все время обучения, и компетенции, соответствующие каждой дисциплине.

Данные из таблиц с помощью разработанного алгоритма, реализованного на языке PHP автоматически загружаются в определенные таблицы БД. Список дисциплин из УП помещается в таблицу «Дисциплины», список компетенций и их описание помещается в таблицу «Компетенции», соответствующие компетенции дисциплинам заносятся в таблицу «Компетенции».

Список должностей из формы «ТребованияКДолжностям» заносится в таблицу «Должности», а требования к ним в таблицу «Требования».

Созданная система имеет несколько интерфейсов, определенных для каждого типа пользователей, а именно: Администратор, Студент, Работодатель, Заведующий.

Администратор-пользователь, обладающий всеми правами, который может производить изменения в самой конфигурации. Студент – пользователь, который не производит никаких изменений, а лишь получает необходимую ему информацию. Работодатель – пользователь, который заносит в систему данные о должностях. Заведующий – пользователь, который загружает в базу данных учебные планы.

Студент входит в систему, выбирает должность, которую в будущем он хочет занимать, система автоматически выводит список лекций определенных дисциплин, необходимых для ее получения.

Таким образом, разработанная информационная система на данный момент времени позволяет автоматически загружать необходимые данные в базу данных и производить автоматическое сравнение требований и компетенций. В результате формируется список необходимых дисциплин, на изучение которых студент должен обратить особое внимание.

На следующем этапе развития системы, планируется реализовать возможность формирования списка дисциплин, ранжированного по проценту соответствия тематического плана требованиям работодателей. На данном этапе в систему будут загружаться рабочие программы, из которых будут в автоматическом режиме выявляться наименование дисциплины, направление подготовки и список тем лекционных (практических) занятий. Выявленная информация будет сохраняться в регистрах сведений, что обеспечит хранение истории. Выявление соответствий между тематическим планом и требованиями работодателей будет осуществляться при помощи тезауруса. На основе результатов такого сравнения будет рассчитываться коэффициент «важности» дисциплины, на основе которого будет осуществляться ранжирование итогового списка дисциплин, необходимых для претендования на желаемую должность.

В результате спроектирована информационная справочная система трудоустройства выпускников, которая позволяет автоматически загружать необходимые данные в базу данных, производить автоматическое сравнение требований и компетенций, позволяет получить список необходимых дисциплин, на изучение которых студент должен обратить особое внимание, позволяет очистить записи информационной базы по определенному направлению.

Литература

1. Информация о системе АИСТ. [Электронный ресурс]. Режим доступа: <http://aist.magtu.ru> — Загл. с экрана.
2. Введение в АИСТ. [Электронный ресурс]. Режим доступа: <http://aist.elsu.ru> — Загл. с экрана.

3. Шайдуров А.А., Зацепин П.М. Информационно-поисковые системы для intranet-сетей // Известия Алтайского государственного университета. 2000. № 1. С. 061-062.

4. Цхай А.А., Рыков Д.А., Сибиряков А.В., Шайдуров А.А. Информационная система мониторинга деятельности сельхозпроизводителей для повышения эффективности регионального управления // Вычислительные технологии. 2011. Т. 16. № 6. С. 85-92.

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КАК ПРОСТРАНСТВО ДЛЯ ВОПЛОЩЕНИЯ ЛЮБОЙ ИДЕИ

Р.В. Краев

Свободное и бесплатное программное обеспечение (ПО) — это разные вещи, так как свободное ПО можно модифицировать, изменять на коренном уровне и добавлять новые модули. Бесплатное ПО — это лицензионная программа с закрытым кодом, что запрещает её модифицировать и производить любые иные действия которые не разрешены лицензией.

Обычный пользователь может не видит разницы между коммерческим и свободным ПО, кроме красивой обложки. Однако на более высоком уровне, где компании тратят большой бюджет для покупки лицензий, в то время как можно нанять специалистов для настройки свободного ПО и забыть про лицензии, это очень серьезная вещь по отношению к бюджету и пользователю.

Как показывает статистика, примерно 90% [1] пользователей имеют на своем домашнем ПК операционную систему Windows. И это не случайность, так как Windows зарекомендовала себя как самая удобная операционная система (ОС). Пользоваться данной ОС для всех кажется простой нормой. Например, можно полностью заменить настройки рабочего стола. Так же в Windows присутствует большое количество служб, которые зачастую не используются, и могут принести потенциальную угрозу фирме. Службы которые не используются также нагружают систему. Их можно отключить, но удалить нельзя. Такие бесполезные службы могут представлять серьезную уязвимость в системе, так как об них забывают пользователи, но помнят злоумышленники.

Однако, если установить Linux, как основную операционную систему — то в результате её настраивания можно добавлять функционал или урезать его. Существует Linux и для обучения, где высокая степень защиты ОС в виде запуска приложений в отдельных виртуальных машинах.[2] Linux представлен в виде ОС для тестирования и проверки безопасности приложений и сетей,[3] в виде максимально уязвимой системы,[4] и т.д. Все эти Linux-системы очень хорошо подходят для обучения специалистов. При этом, указанные виды Linux представляют только малую часть Linux-систем. Linux как свободная ОС очень далеко продвинулась по возможностям модифицирования и настройки, до той степени которая нужна конечному пользователю.

Свободное ПО используется большим количеством пользователей: от обычного офисного работника, который работает с мультимедийными файлами до достаточно серьёзного специалиста всех сфер. Можно создать на основе любой свободной программы свою версию модификации с добавлением нужного функционала или его удаления за ненадобностью.

Свободное ПО позволяет не только возможность модифицировать и предлагать свои версии программ. Зная принцип работы ядра программы, продвинутое сообщество программистов, может помочь в поисках уязвимости и улучшении функционала продукта, для обеспечения нужд рядовых пользователей. Таким образом, коллективное создание свободного ПО — простая

возможность реализовать любую безумную идею. Некоторые программы созданные в сообществе и являющиеся свободными, имеют достаточно серьезную поддержку сообщества, которое указывает на недостатки и помогает данные программы сделать лучше [5,6].

Коммерческое ПО никогда не сможет достичь тех целей которое может свободное ПО. Хотя коммерческое ПО и легко поставить и настроить, но оно ограничено и не позволит вам выйти за рамки, которые были поставлены производителем.

Что касается аппаратной части, зачастую свободное ПО требует меньше ресурсов. Если в свободном ПО есть ненужная функция, то вы можете её убрать. То есть не просто в настройках отключить, а удалить из функционала ПО или ОС, тем самым уменьшив размер самой программы и возможно увеличив быстродействие, из-за отсутствия этой функции для которой требовались дополнительные ресурсы.

К примеру, можно взять тот же Windows которые не способен без значительных ресурсов запустится на микроконтроллере или же портативном ПК, например Raspberry Pi [7]. Конечно, сейчас аппаратная часть этого карманного ПК ушла далеко и может поддерживать Windows, но изначально на ней мог стоять только Linux, так как его можно было модифицировать и оставлять только необходимый для платформы функционал.

Литература

1. Статистическая доля рынка интернет технологий. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <http://www.netmarketshare.com>
2. Официальный сайт TAILS. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://tails.boum.org/index.en.html>
3. Официальный сайт KALI. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://www.kali.org>
4. Самая уязвимая система Damn Vulnerable Linux. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <http://www.damnulnerablelinux.org>.
5. Цхай А.А., Рыков Д.А., Сибиряков А.В., Шайдуров А.А. Информационная система мониторинга деятельности сельхозпроизводителей для повышения эффективности регионального управления // Вычислительные технологии. 2011. Т. 16. № 6. С. 85-92.
6. Цхай А.А., Рыков Д.А., Сибиряков А.В., Шайдуров А.А. Информационно-моделирующая система мониторинга деятельности сельхозпроизводителей региона // Известия Алтайского государственного университета. 2011. № 1-2. С. 126-130.
7. Что такое Raspberry Pi. [Электронный ресурс] — Заглавие с экрана. Режим доступа: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi>.

ПРИМЕНЕНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МНОГОМЕРНОГО АНАЛИЗА ДАННЫХ

О.В. Махныткина, С.К. Хакимов, С.Ш. Чилаев

В настоящее время многомерный анализ данных активно применяется практически во всех областях исследований [1, 2, 3, 4]. Многомерный анализ данных представляет собой совокупность методов и алгоритмов, которые позволяют получать информацию о массиве данных, расположенных в некоторой области многомерного пространства.

Реализация алгоритмов многомерного анализа данных наряду с коммерческими статистическими пакетами, также представлена в свободном программном обеспечении для статистического анализа данных. При этом необходимо отметить, что некоторые свободно распространяемые статистические пакеты не только не уступают, но и превосходят по функциональности коммерческие. Рассмотрим некоторые примеры свободного ПО для многомерного анализа данных.

PSPP является свободным программным обеспечением для статистического анализа данных. PSPP позволяет проводить расчет основных показателей описательной статистики и применять основные методы многомерного анализа данных. Так, в PSPP представлен корреляционный анализ (позволяет оценить степень тесноты связи между переменными); регрессионный анализ (определяет аналитическую зависимость между показателями и факторами); кластерный анализ (служит для нахождения групп сходных объектов); факторный анализ (предназначен для выделения из множества всех факторов те, которые действительно оказывают влияние на изменение зависимой переменной).

С использованием PSPP проводятся научные исследования в различных областях. Например, в работе [5] с использованием PSPP был выполнен статистический анализ динамики морфометрических параметров макулярной зоны по данным оптической когерентной томографии у пациентов с возрастной макулярной дегенерацией после факэмульсификации катаракты с имплантацией интраокулярной линзы. В [6] анализируется эффективность восстановления целостности грудины с использованием имплантата из никелида титана при послеоперационных стерномедиастинитах. В работе [7] показаны возможности факторного анализа, который в совокупности с теоретическим исследованием позволяет определить эффективность национальной пенитенциарной системы.

Примером пакетов с открытым кодом является пакет R, объединяющий в себе одноименный язык и среду статистической обработки данных. В настоящее время R предоставляет большое количество методов, в том числе линейное и нелинейное программирование, классификацию и кластеризацию, факторный и компонентный анализ и пр. Преимущество R состоит в возможности расширений за счёт дополнений, при этом создание новых дополнений не требует знания никаких языков программирования кроме самого

Р. Считается, что R является лидером среди свободно распространяемых систем статистического анализа и используется широким кругом специалистов. Например, работе [8] на основе компонентного анализа с использованием R был построен сводный показатель финансовых рисков для оценки способности финансовой системы качественно и бесперебойно выполнять свои функции. В [9] рассматривается использование дисперсионного анализа, регрессионного анализа, многомерных методов классификации на примерах биологического характера.

Среди других примеров свободного ПО для многомерного анализа данных можно выделить Gretl, SIMFIT, Weka, KNIME и проч. Таким образом, в настоящее время свободное ПО предоставляет широкие возможности для проведения многомерного анализа данных.

Литература

1. Ниमेंья И.Н. Многомерный статистический анализ развития кризиса в реальном секторе экономики//Вестник ИНЖЭКОНа. Серия: Экономика. 2012. № 2. С. 151-159.
2. Каширский Д.В. Гроблемы применения количественных методов в современной российской психологии//Вестник алтайской науки. 2013. № 2-1. С. 170-174.
3. Беляев И.А., Кучерявский С.В., Родионова О.Е., Померанцев А.Л. Многомерный анализ изображений в медицине и биологии//Энциклопедия инженера-химика. 2011. № 2. С. 13-23.
4. Мальцева А.В. Современные информационные технологии в обеспечении прикладных исследовательских задач в социально-культурной сфере//Известия Алтайского государственного университета. 2009. № 1. С. 143.
5. Кузнецов А.А., Тур Е.В., Кузнецова Н.В., Быстров А.М. динамика морфометрических параметров макулярной зоны по данным оптической когерентной томографии у пациентов с возрастной макулярной дегенерацией после факэмульсификации катаракты с имплантацией интраокулярной линзы//Вестник Оренбургского государственного университета. 2014. № 12. С. 190-194.
6. Шведова М.В., Вусик А.Н., Дамбаев Г.Ц., Козлов А.Е., Порохова Е.Д. Восстановление целостности грудины у пациентов с послеоперационным стерномедиастинитом//Сибирский медицинский журнал (г. Томск). 2014. Т. 29. № 4. С. 45-51.
7. Тепляшин П.В. Оценка эффективности пенитенциарной системы: метод факторного анализа//Юридическая наука и правоохранительная практика. 2015. № 2 (32). С. 80-90.
8. Гамбаров Г.М. Мхитарян В.С. Моделирование обобщенного показателя рисков финансовой системы страны//Математико-статистический анализ социально-экономических процессов: Межвузовский сборник научных

трудов. Выпуск 10/ Московский государственный университет экономики, статистики и информатики.- М., 2013.-С 21-26

9. Шитиков В.К., Розенберг Г.С. Рандомизация и бутстреп: статистический анализ в биологии и экологии с использованием R. - Тольятти: Кассандра, 2013. - 314 с.

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ ПЛАГИАТА НА ОСНОВЕ НЕРЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ MONGODB И СИСТЕМЫ ДЛЯ РАСПРЕДЕЛЕННОГО ВЫЧИСЛЕНИЯ APACHE HADOOP

С.В. Мурзинцев, А.А. Цхай

В связи с бурным развитием информационных технологий и повсеместным их внедрением во все сферы человеческой деятельности появляется большое количество информации, которая аккумулируется на различных сайтах в сети Интернет. Это ведет к достаточно интенсивному обмену идеями, мыслительного плагиата в различных сферах: образования, науки, промышленности. Проблема, связанная с заимствованием идей и результатов научной деятельности, активно обсуждается в образовании и научном сообществе. Для поиска механизма защиты в различных учреждениях создаются отделы, которые занимаются проверкой научных работ на наличие заимствований как «ручным» способом, так и с использованием специализированных программ. Выявление плагиата в документах достаточно сложная проблема, так как плагиат может варьироваться от прямого копирования текстов до плагиата идей. В данной статье мы бы хотели акцентировать внимание на бесплатном и свободном программном обеспечении, которое может использоваться для разработки систем занимающихся выявлением заимствованием.

На рынке представлены различные сервисы по обнаружению плагиата, например: интернет-сервис «Антиплагиат», «eTXT Антиплагиат», SafeAssign, CopyScape. Все перечисленные системы имеют закрытый исходный код и отсутствие какого-либо упоминания о технологиях, которые используются при анализе документов.

Для разработки нашей системы мы используем следующее программное обеспечение:

1. **MongoDB 2.4.9** - документо-ориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц.

2. **Hadoop Apache** - проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и фреймворков для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Используется для реализации поисковых и контекстных механизмов многих высоконагруженных веб-сайтов.

3. **Linux Ubuntu server 12.04 LTS** - операционная система, основанная на Debian GNU/Linux.

4. **Eclipse** – это программная платформа с открытым исходным кодом, написанная на языке Java. Основная цель ее создания - повышение производительности процесса разработки программного обеспечения.

5. **Robomongo 0.8.5** - это оболочка, кросс-платформенный инструмент с открытым исходным кодом для управления MongoDB.

6. **VirtualBox** - программный продукт виртуализации для

операционных систем Microsoft Windows, Linux, FreeBSD, Mac OS X, Solaris/OpenSolaris, ReactOS, DOS и других.

На основе описанного программного обеспечения нами была разработана система обнаружения плагиата, общая структура которой представлена ниже. Работа системы разделена на две части: клиентскую и серверную. Клиентская часть включает в себя формирование «отпечатков» документа, используя алгоритм Winnowing [1] и отправку сформированных «отпечатков» на сервер для проверки. Алгоритм работы пользователя, для проверки документа на наличие заимствований предполагает, что:

1. Пользователь скачивает программу с сайта компании, предоставляющие соответствующие услуги.
2. Пользователь с помощью скачанной программы формирует «отпечатки» проверяемого документа используя алгоритм Winnowing.
3. Пользователь отправляет «отпечатки» своего документа на сервер предприятия, указав параметры при которых он осуществлял формирование «отпечатков».
4. Пользователь получает отчет о проверяемом документе.

Серверная часть занимается получением, обработкой, сравнением «отпечатков» документов с базой «отпечатков» организации и формированием отчётов для отправки пользователям.

Из представленной архитектуры становится очевидно, что содержимое документа пользователя для организации занимающейся проверкой документов на заимствования, является полностью закрытым. Организация работает только с «отпечатками» (хешами), которые предоставил пользователь. Восстановить документ пользователя по предоставленным «отпечаткам» не представляется возможным, так как, во первых, алгоритм Winnowing отбрасывает часть хешей, тем самым ужимая содержимое документа, а, во вторых, содержимое данных для проверки представлено в виде хешей («отпечатков»). Таким образом становится ясно, что исходный документ пользователя гарантированно не подлежит восстановлению.

Очевидно, что предъявленная архитектура существенно увеличивает требования к мощности вычислительных ресурсов, которые необходимы для сравнения «отпечатков», поступающих для анализа на плагиат. Так же данная система требует больших объемов дискового пространства для хранения «отпечатков» всех документов с которыми сравниваются присылаемые «отпечатки».

Несмотря на все описанные недостатки, данный подход к защите содержимого документа пользователя становится всё более реализуем [2]. Проблема, связанная с хранением большого количества данных, может быть решена за счёт использования нереляционной базы данных MongoDB и технологии распределенного хранения данных (Sharding в MongoDB). А технология MapReduce может способствовать решению проблемы, связанной с поиском среди большого количества документов [3].

Нами был разработан прототип информационной системы для проверки

документов на заимствования, в котором используются описанные технологии. Структура информационной системы представлена на рисунке 1.

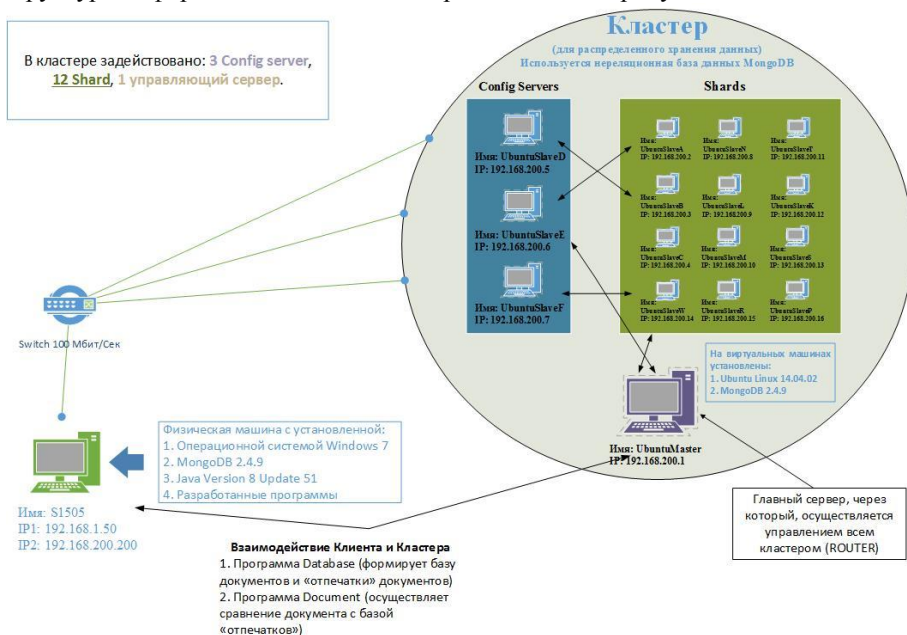


Рисунок 1. Прототип информационной системы для защищенной проверки документов пользователей на заимствования

Разработанная система способствовала лучшему пониманию механизмов, связанных с работой поиска в нереляционной базе MongoDB при использовании Sharding и технологии MapReduce. В настоящее время осуществляется совершенствование возможностей информационной системы через написание новых модулей, а также специальных алгоритмов для работы с другими технологиями, например, такими как Apache Hadoop [4].

В заключении стоит отметить что благодаря используемому бесплатному и свободному программному обеспечению был разработан прототип программного комплекса и проведены соответствующие эксперименты, которые способствовали выработки новых стратегий по улучшению описанного механизма проверки документов на наличие плагиата. Так же в перспективе планируется интеграция нереляционной базы данных MongoDB и системы распределенного вычисления Hadoop Apache. MongoDB будет выступать в роли системы для хранения данных, а Apache Hadoop в роли средства, использующего технологию MapReduce для распределенного поиска.

Литература

1. Schleimer S., Wilkerson D., Aiken A. Winnowing: Local Algorithms for Document Fingerprinting // ACM SIGMOD International Conference on Management of Data, June 2003. Proceedings. P. 76-85.

2. Цхай А.А., Бутаков С.В., Мурзинцев С.В., Ким Л.С. Обнаружение плагиата с использованием нереляционных баз данных // Вестник алтайской науки 2015. Выпуск: 1. С. 280-285.

3. Документация по работе с базой данных MongoDB. [Электронный ресурс]URL: <http://docs.mongodb.org/manual/reference/command/mapReduce/> (дата обращения: 12.09.2015)

4. Apache Hadoop. [Электронный ресурс] <http://hadoop.apache.org> (дата обращения: 12.09.2015)

РАЗРАБОТКА СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СКРИНИНГА ЗАБОЛЕВАНИЙ

К. П. Стародубцев

Одной из наиболее важных проблем в современной медицине является ранняя диагностика заболеваний. Диагностика больных затруднена постоянным изменением данных, в силу которого приходится обрабатывать большие массивы информации. Поэтому, одним из путей повышения эффективности диагностики, является разработка проблемно-ориентированных систем обработки информации и применение к ним математических методов анализа информации.

Использование методов искусственного интеллекта и статистических методов составляет практическую основу для построения системы поддержки принятия диагностических решений [1]. Рекомендации системы могут использоваться как врачами для принятия решений о состоянии здоровья пациента, так и самими пациентами для определения предрасположенности к тому или иному заболеванию. Цель проекта заключается в исследовании взаимосвязей между исходными данными и разработке проблемно-ориентированной системы обработки информации и принятия диагностических решений.

Для создания полноценной интеллектуальной диагностической системы необходимо осуществление трех основных этапов: сбор данных, математический анализ данных, разработка программного обеспечения.

Сбор данных о пациентах подразумевает формирование информации о пациентах, которым поставлен исследуемый диагноз и о пациентах, которым не поставлен исследуемый диагноз. При этом анализируемые данные должны иметь достаточное количество записей, чтобы была возможность их обработки математическими методами.

Математический анализ данных подразумевает предварительную обработку данных, а также исследование данных при помощи методов статистического анализа (факторный, кластерный, дискриминантный). Кроме методов статистического анализа возможно включить использование методов искусственного интеллекта, в частности искусственных нейронных сетей.

Нейросетевые методы позволяют более точно осуществлять диагностику пациентов в условиях большой лабильности клинических проявлений. Соответственно необходимо проведение исследований в целях создания системы обработки информации и интеллектуальной поддержки принятия решений на основе применения методов системного анализа. Это позволит на ранних стадиях развития заболеваний своевременно и точно оценивать вид поражения и проводить лечебные мероприятия [2].

Разрабатываемая система в первую очередь должна быть направлена на профилактику заболеваний и соответственно, на излечение больных на первичном уровне.

Для обеспечения профилактических и реабилитационных мероприятий разрабатывается комплекс программ, обладающих WEB-интерфейсом и соответственно доступных любому пользователю ПК, имеющему выход в INTERNET. На основе вводимой информации, содержащей историю болезни пациента, система определяет принадлежность человека к той или иной группе риска, и формирует рекомендации по поддержанию здоровья.

Создаваемая экспертная система предоставит врачам центральных районных больниц информацию о предполагаемом диагнозе и о сопутствующих диагнозах, осложнениях, аллергиях и прочем, а также информацию об оптимальном подборе лекарственных средств на основе данных. Также возможности системы позволят пациентам самим определить предрасположенность к тем или иным заболеваниям и получить рекомендации для предотвращения болезни еще до ее возникновения [3].

Данная диагностическая система позволит врачам на первичном уровне с большой точностью диагностировать заболевания и назначать перечень лечебных, профилактических и реабилитационных мероприятий даже для диагнозов, имеющих слабо выраженную симптоматику.

Таким образом, разрабатываемая система позволит на первичном уровне обеспечить снижение уровня заболеваний населения без привлечения специализированного и высокотехнологичного медицинского оборудования краевого уровня.

Литература

1. Шайдуров А.А., Зацепин П.М., Шатохин А.С., Пиянзин А.И., Акинина З.Ф. Нейросетевой диагностический комплекс с элементами автоматической модификации // Нейроинформатика и ее приложения: Материалы XII Всероссийского семинара. Красноярск, 2004. с. 171-172.
2. Интеллектуальные информационные системы: учеб. пособие/А.А. Смагин, С.В. Липатова, А.С. Мельниченко. -Ульяновск: УлГУ, 2010,-136 с.
3. Бубликов А.С., Шайдуров А.А. Анализ применения искусственных нейронных сетей для построения медицинских экспертных систем // Ломоносовские чтения на алтае: фундаментальные проблемы науки и образования, Барнаул, 11-14 ноября 2014 г. Барнаул, 2014. С. 591-594.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО ОБЕЗЛИЧИВАНИЮ ПЕРСОНАЛЬНЫХ ДАННЫХ БОЛЬШИХ ОБЪЕМОВ

В. С. Трофимов

Компьютерный мир невозможно представить без программного обеспечения. Оно обеспечивает взаимодействие между пользователем и операционной системой [1].

Программное обеспечение по способу распространения и использования подразделяется на проприетарное, открытое и свободное [2].

Основной особенностью свободно распространяемого программного обеспечения является свобода пользователя на его неограниченную установку, запуск, изучение, изменение и распространение копий и результатов изменения.

Именно из этих соображений свободно распространяемое программное обеспечение в настоящий момент активно внедряется в государственные учреждения. Но зачастую это нарушает безопасность конфиденциальной информации организации, так как в большинстве случаев не предусмотрена даже малейшая защита.

Обеспечение безопасности информации в вычислительных системах реализуется в совокупности с требованиями конфиденциальности, целостности и доступности. Конфиденциальность обеспечивается выполнением требований существующего законодательства: защита интеллектуальной собственности, защита профессиональной тайны, защита персональных данных и т.д. [3].

К способам, позволяющим обеспечить конфиденциальность персональных данных при сокращении требуемых материальных ресурсов, относится обезличивание. Известен целый набор методов обезличивания [4]:

- метод введения идентификаторов;
- метод изменения состава или семантики;
- метод декомпозиции;
- метод перемешивания.

Каждый из них обеспечивает определенные преимущества и приносит свои недостатки [5]. Для указанных выше методов представлена диаграмма соответствия между методами и свойствами обезличенных данных (рис.1).

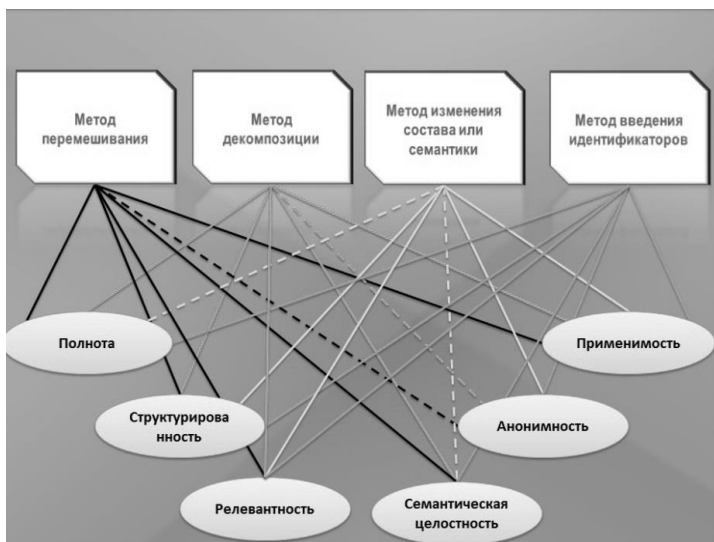


Рис. 1 - Диаграмма соответствия между методами и свойствами обезличенных данных.

Представляло интерес разработать программное обеспечение по алгоритму обезличивания, способные работать с большими массивами данных. При выборе метода обезличивания в качестве определяющих факторов выделены:

- скорость работы алгоритма;
- потребление ресурсов системы;
- защищенность обезличенных персональных данных;
- возможность восстановить исходные данные.

Применен следующий подход – при разработке алгоритма обезличивания персональных данных совместно использованы методы перемешивания и введения идентификаторов.

Алгоритм обезличивания реализован следующим образом:

- производится перемешивание элементов исходного массива данных;
- значения одного из атрибутов из массива данных заменяются на идентификаторы;
- исходные значения замененного идентификаторами атрибута записываются в массив, хранимый отдельно.

Примененная комбинация методов обезличивания позволяет решать следующие задачи:

- защищать данные от вскрытия;
- дополнить преимущества одного метода преимуществами другого (рис.1).

В разработанном программном обеспечении реализованы следующие функции: работа с файлами в формате *.CSV; работа с большими объемами данных;

обезличивание персональных данных путем перемешивания и/или введения идентификаторов.

Панель программы ЭВМ по обезличиванию персональных данных представлена на рис. 2.

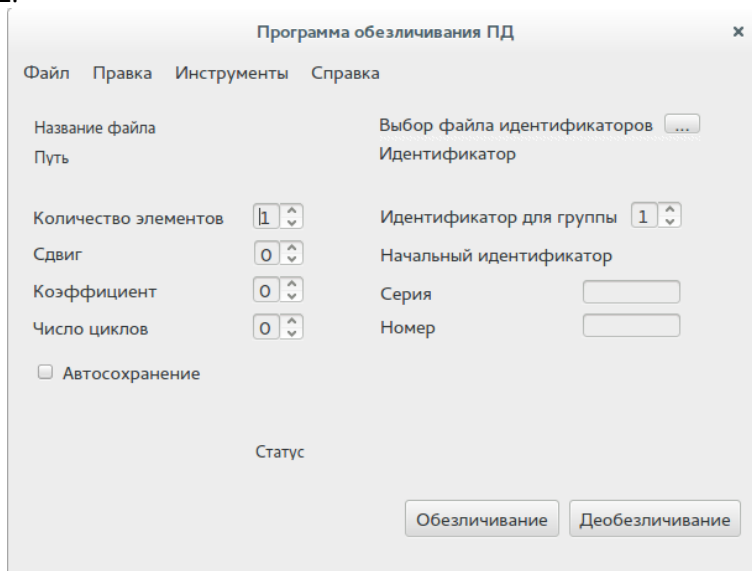


Рис. 2 - Внешний вид программы.

Особенности разработанного программного обеспечения: кроссплатформенность; низкое потребление ресурсов при больших объемах данных; широкий набор настроек; поддержка «горячих» клавиш; интуитивно понятный интерфейс.

Алгоритм апробировался для различных входных данных и показал эффективную работу. Программа работает с достаточно большими массивами данных. На практике были обработаны массивы, содержащие порядка 6 000 000 записей, затраченное время составило около 35 секунд.

Таким образом, на основании анализа существующих методов обезличивания персональных данных реализовано программное обеспечение, способное обезличивать большие массивы данных.

Литература

1. ГОСТ 19781-90 Термины и определения.
2. ГОСТ 19.101-77 Виды программ и программных документов.
3. Минакова Н.Н. Информационная система анализа структуры радужной оболочки глаза / Н.Н. Минакова, И.В. Петров // Ползуновский вестник. – 2012. – № 3/2. – С. 230–234.

Методические рекомендации по применению приказа Роскомнадзора от 5 сентября 2013 г. № 996 «Об утверждении требований и методов по обезличиванию персональных данных».

4. Защита персональных данных в организациях здравоохранения /А.Г. Сабанов, В.Д. Зыков, Р.В. Мещеряков и др. /Под ред. А.Г. Сабанова. – М.: Горячая линия. – Телеком, 2012 – 206 с.

БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ОТКРЫТЫМ И ЗАКРЫТЫМ ИСХОДНЫМ КОДОМ

А.Е. Фролов

Программное обеспечение (ПО) представляет собой совокупность команд, объединенные в готовые программы согласно реализации определенных алгоритмов для обработки какой-либо информации, необходимой для эксплуатации этого ПО. Зачастую, современное ПО создается для обработки данных, необходимых для эксплуатации другого ПО (компонентов программы), или даже целых групп ПО (программного комплекса). При большом количестве таких комплексов, дополняющих друг друга, они могут объединяться в один большой комплекс, автоматизирующий управление обработкой данных и предоставляющий пользователю определенный набор услуг для управления этим комплексом. Такие комплексы называются операционными системами [1-3].

На сегодняшний день количество ПО, создаваемого для решения каких-либо задач, создается огромное количество, причем они различаются как по целям, которые преследуют разработчики, так и по объемам этого ПО. Объем ПО и скорость изменение его структуры зависит от популярности этого продукта на рынке. От популярности ПО зависит финансирование проекта, который направлен на поддержку спроса этого продукта на рынке, а в современных условиях многообразия предлагаемых продуктов, на рынке определена жесткая конкуренция между производителями ПО. Такая конкуренция требует производить достаточно оперативное написание кода программ, жертвуя при этом временем на поиск ошибок. Появление таких ошибок в коде ПО ведет к возникновению уязвимостей, которыми может воспользоваться злоумышленник для получения несанкционированного доступа (НСД) [4] к ресурсам системы, на котором используется это ПО. Последствия НСД могут быть весьма незначительными если речь идет об общедоступных данных (например регламенте работы или графике отпусков), или нанести непоправимый вред организации, если это будут конфиденциальные данные (например секретные разработки).

Разрабатываемое ПО, в зависимости от его назначения, принято делить на 3 большие категории:

1. системное;
2. прикладное;
3. инструментальное.

Отнесение ПО к тому или иному пункту классификации позволяет судить о степени значимости этого ПО, его объеме и распространенности, а так же наличию уязвимостей в этом ПО.

Основным фактором наличия уязвимостей ПО является количеством строк исходного кода. Такой фактор называется плотностью кода (KLOC) [5] и влияет на количество уязвимостей, содержащихся в нем. Обычно этот фактор определяется объемом ПО и расчетом количества ошибок на 1000 строк кода и в

большинстве случаев исходят из соображений 1 ошибки на 1000 строк [5] (обычное прикладное ПО, рассчитанное на решение мелких задач, содержит всего несколько тысяч строк кода, в то время как современные операционные системы содержат в себе десятки миллионов строк кода). Часть из этих ошибок приходится на проектирование алгоритма ПО, часть допускается во время кодирования, причем с увеличением размеров проектов возрастает и плотность ошибок на 1000 строк кода. Стивом Макконаком была выявлена зависимость плотности ошибок от размера проекта [5] (таблица 1).

Таблица 1 — Зависимость уязвимости в исходном коде от его объема.

Размер проекта (в тысячах строк кода), K	Плотность ошибок (на 1000 строк кода), KLOC
Менее 2	0-25
2-16	0-40
16-64	0,5-50
64-512	2-70
Более 512	4-100

Из таблицы 1 видно какое количество ошибок может содержать в себе ПО. Этот показатель может варьироваться в зависимости от выделенных ресурсов, направленных на поиск и устранение этих ошибок.

Основными методами, направленными на поиск и исправление этих ошибок являются следующие техники:

- статический анализ кода;
- динамический анализ кода (fuzz-testing);
- инспекция кода (code review);
- реорганизация кода (рефакторинг);
- обратная разработка (реверс-инжиниринг).

Однако процесс поиска ошибок в коде ПО иногда бывает затруднен по причине ограничения его анализа некоторыми типами лицензий, которые регламентируют правила использования и распространения этого ПО без нарушения авторских прав [6]. Существует множество типов лицензий (freeware, beerware, linkware, shareware, payware, Cdware, open source и др.), которые регламентируют правила использования ПО [100]. Каждый из этих типов лицензий может содержать несколько однотипных лицензий (GNU GPL, LGPL, BSD, OEM, MIT, EULA, PUR и др.), описывающее право использования этого ПО. Все эти лицензии делят все ПО на 2 большие категории:

- свободное (ПО с открытым исходным кодом);
- проприетарное (ПО с закрытым исходным кодом) [7].

Свободное ПО (СПО) распространяется вместе с исходными кодами, позволяя каждому желающему проанализировать этот код, и в случае

обнаружения ошибки его исправить. Проприетарное ПО (ППО), в отличие от свободного ПО, распространяется в виде уже собранных исполняемых бинарных файлов, что существенно затрудняет поиск ошибок в коде ПО и его исправления.

Исходя из соображений безопасности, кампанией Coverity совместно со Стэнфордским университетом были проанализированы программные продукты установленные в государственных учреждениях [8]. Результаты исследования приведены в таблице 2.

Таблица 2 - Сравнение безопасности СПО и ППО.

Показатель	СПО	ППО
Кол-во проанализированных проектов	37	41
Средний размер проекта	832 000	7 500 000
Кол-во ошибок на KLOC	0,45	0,64

Эти данные показывают, что СПО содержит меньше уязвимостей, чем тоже самое ППО, что делает его более безопасным в плане эксплуатации.

Еще одним фактором безопасности, при использования СПО, является «открытость» исходного кода, позволяющая проанализировать весь его функционал, в отличии от ППО, где исходный код является закрытым, а алгоритм представляет собой так называемый «черный ящик». Этот фактор «черного ящика» ограничивает свободный доступ к исходному коду такого ПО, тем самым снижая вероятность обнаружения ошибок и недокументированных возможностей. Из-за фактора снижается общий уровень безопасности По. Для поддержания необходимого уровня безопасности такого ПО, приходится приобретать дополнительные средства защиты, что ведет к увеличению расходов на безопасность конфиденциальной информации.

Таким образом, можно сделать вывод, что использование СПО является более безопасным и выгодным, нежели использование ППО. Это достигается за счет открытости исходного кода обществу, что позволяет улучшить качество используемого ПО. Кроме того, «открытость» кода делает любое СПО абсолютно прозрачным, что исключает обнаружения в нем недокументированных возможностей (вредоносных закладок), от которых, в любом случае, приходится защищаться при использовании ППО. Все эти факторы делают использование СПО и его развитие весьма перспективным направлением.

Литература

1. Обеспечение систем обработки информации программное [Текст]: ГОСТ 19781-90. - Введ. 1992-01 -01. - М.: Изд-во стандартов, 1990. - 22 с.: ил.
2. Виды программ и программных документов [Текст]: ГОСТ 19.101-77. -

- Введ. 1980-01 -01. - М.: Изд-во стандартов, 1987 (переиздание). - 4 с.: ил.
3. Системы обработки информации. Термины и определения [Текст]: ГОСТ 15971-90. - Введ. 1992-01 -01. - М.: Изд-во стандартов, 1991. - 22 с.: ил.
 4. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования [Текст]: ГОСТ Р 50739-95. - Введ. 1996-01 -01. - М.: Стандартиформ, 2006 (переиздание). - 8 с.: ил.
 5. Макконнелл С. Совершенный код. Мастер#класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр. : ил.
 6. Виды лицензий на программы [Электронный ресурс]: Чайникам-нет! (ред. От 02.12.2012) - . - Режим доступа: http://www.chaynikam.net/view_articles.php?id=22, свободный. - Загл. с экрана.
 7. Лицензия для вашего open-source проекта [Электронный ресурс]: Habrahabr.ru (ред. От 13.11.2014) - . - Режим доступа: <http://habrahabr.ru/post/243091/>, свободный. - Загл. с экрана.
 8. Coverity: На 1000 строк исходного кода открытых программ насчитывается 1 дефект [Электронный ресурс]: SecurityLab (ред. От 27.02.2012) - . - Режим доступа: <http://www.securitylab.ru/news/420674.php>, свободный. - Загл. с экрана.

Оглавление

О.В. Журенков СПО И ГОСУДАРСТВО: СОВРЕМЕННЫЕ ТЕНДЕНЦИИ И ПЕРСПЕКТИВЫ	1
А.К. Букина ПРОГРАММНЫЙ КОМПЛЕКС WRF	9
А.А. Барбышев MONDRIAN И SAIKU КАК СРЕДСТВО АНАЛИЗА БОЛЬШИХ МАССИВОВ ДАННЫХ	14
А.С. Большаков ПРИМЕНЕНИЕ R И RSTUDIO ДЛЯ АНАЛИЗА ПОЛИТИЧЕСКОГО ИМИДЖА ГОСУДАРСТВ	19
С.С. Гарбуз РАЗРАБОТКА СПЕЦИАЛИЗИРОВАННОГО WEB-ИНТЕРФЕЙСА СБОРА И АНАЛИЗА ДАННЫХ	25
Д.А. Дмитриев ФАКС В VOIP-СЕТЯХ.	28
А.А. Дружинина ПРОЕКТИРОВАНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОМОЩИ СТУДЕНТУ В ТРУДОУСТРОЙСТВЕ	30
Р.В. Краев СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КАК ПРОСТРАНСТВО ДЛЯ ВОПЛОЩЕНИЯ ЛЮБОЙ ИДЕИ	34
О.В. Махныткина, С.К. Хакимов, С.Ш. Чилаев ПРИМЕНЕНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МНОГОМЕРНОГО АНАЛИЗА ДАННЫХ	36
С.В. Мурзинцев, А.А. Цхай РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ ПЛАГИАТА НА ОСНОВЕ НЕРЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ MONGODB И СИСТЕМЫ ДЛЯ РАСПРЕДЕЛЕННОГО ВЫЧИСЛЕНИЯ APACHE HADOOP	39
К.П. Стародубцев РАЗРАБОТКА СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СКРИНИНГА ЗАБОЛЕВАНИЙ	43
В.С. Трофимов РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО ОБЕЗЛИЧИВАНИЮ ПЕРСОНАЛЬНЫХ ДАННЫХ БОЛЬШИХ ОБЪЕМОВ	45
А.Е. Фролов БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ОТКРЫТЫМ И ЗАКРЫТЫМ ИСХОДНЫМ КОДОМ	49