№9 2013
сентябрь

электроника инфо

Научно-практический журнал для специалистов
Республика Беларусь, г. Минск

Тема номера:
«Компьютерные сети. Облачные технологии»

ПОДПИСКА В БЕЛАРУСИ: «Белпочта» – 00822
ПОДПИСКА В РОССИИ: «Роспечать» – 00822
«АРЗИ – Почта России» – 91654

ISSN 1999-7515

ЖУРНАЛ ИЗДАЕТСЯ ПРИ ИНФОРМАЦИОННОЙ ПОДДЕРЖКЕ ФАКУЛЬТЕТА РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ БЕЛГОСУНИВЕРСИТЕТА. ЖУРНАЛ ВКЛЮЧЕН В СПИСОК НАУЧНО-ТЕХНИЧЕСКИХ ИЗДАНИЙ ВАК РЕСПУБЛИКИ БЕЛАРУСЬ

электроника инфо

International magine of amateur and professional electronics

№9 (99) сентябрь 2013

Зарегистрирован Министерством информации Республики Беларусь

Регистрационный №71 от 05 марта 2009 года

Главный редактор:
Раковец Леонид Иванович

Заместитель главного редактора:
Асмоловская Ирина Михайловна
i.asmalouskaya@electronica.by

Редактор технический:
Бортник Ольга Викторовна

Редакционная коллегия:

Председатель:
Чернявский Александр Федорович
академик НАН Беларуси, д.т.н.

Секретарь:
Садов Василий Сергеевич, к.т.н.
e-mail: sadov@bsu.by

Члены редакционной коллегии:
Беляев Борис Илларионович, д.ф.-м.н.
Борздов Владимир Михайлович, д.ф.-м.н.
Голенков Владимир Васильевич, д.т.н.
Гончаров Виктор Константинович, д.ф.-м.н.
Есман Александр Константинович, д.ф.-м.н.
Ильин Виктор Николаевич, д.т.н.
Кугейко Михаил Михайлович, д.ф.-м.н.
Кучинский Петр Васильевич, д.ф.-м.н.
Мулярчик Степан Григорьевич, д.т.н.
Петровский Александр Александрович, д.т.н.
Попечиц Владимир Иванович, д.ф.-м.н.
Рудницкий Антон Сергеевич, д.ф.-м.н.

Отдел рекламы и распространения:
Антоневич Светлана Геннадьевна
тел./факс. +375 (17) 204-40-00
e-mail: s.antonevich@fek.by

Учредитель:
ТЧУП «Белэлектронконтракт»
220015, Республика Беларусь,
г. Минск, пр. Пушкина, 29 Б,
тел./факс: +375 (17) 210-21-89,
+ 375 (17) 204-40-00

Подписной индекс в РБ:
00822 (индивидуальная),
008222 (ведомственная)

Цена свободная

Подготовка, печать:
150 экз. отпечатано
тип. ООО «Поликрафт»
г. Минск, ул. Кнорина, 50/4-401А
Лицензия №02330/0494199 от 03.04.2009 г.
Подписано в печать 09.09.2013 г.
Заказ №

СПИСОК РЕКЛАМОДАТЕЛЕЙ

ОТЕЦ ВСЕМИРНОЙ ПАУТИНЫ

Имя Пол Бэран, скорее всего, неизвестно большинству современных пользователей Интернета. А между прочим, если бы не он, то вряд ли бы в настоящее время кто-нибудь смог бы собирать информацию, общаться и заниматься коммерцией с помощью Глобальной Паутины.

Именно этот скромный ученый явился создателем первой сети, на основе которой после и появился Интернет. Всему миру он был известен как Пол Бэран, и мало кто помнит, что исходно его имя звучало по-другому – Павел Баранов.

Интересно, что о личной жизни и творческих научных достижениях создателя Интернета известно весьма и весьма немного. Это неудивительно – ведь большую часть своей жизни он проработал в закрытых лабораториях, подчиняющихся Пентагону, а это ведомство обычно не разглашает подобные данные. Именно поэтому этот гениальный исследователь не публиковал результаты своих работ в рецензируемых журналах и очень редко давал интервью. Тем не менее, кое-что о его прошлом все-таки известно.

Павел Баранов родился в 1926 году в городе Гродно, который сейчас находится на территории Республики Беларусь, но в те времена эта территория принадлежала Польше. Так что будущий создатель Интернета был белорусом по национальности и поляком по гражданству. Его родители, Мойше (Моррис) Баран (1884–1979) и Хана-Фейга (Анна) Серейская (1888–1972), происходили из местечка Сидра Соколковского уезда Гродненской губернии. В 1928 году он с семьей переехал в США. Сначала семья поселилась в Бостоне, где его отец устроился разнорабочим на обувную фабрику; затем перебралась в Филадельфию, где отец открыл овощную лавку.

В 1949 году окончил Технологический институт Дрекселя, правда, лишь со степенью бакалавра по специальности «инженер электросетей». Работал в компании Eckert-MauchlyComputer, затем в HughesAircraft в Лос-Анджелесе, параллельно заканчивая свою докторскую. После чего поступил на службу в корпорацию RAND, которая работала в интересах национальной безопасности, специализируясь на производстве вооружений, а с начала 60-х – еще и на вычислительной технике. Работая там, Бэран в течение всей своей жизни занимался исследованиями в области военных коммуникаций.

В 60-х годах прошлого столетия, в самый разгар холодной войны, Бэран предложил разработку, которая потом легла в основу современного Интернета. Она заключалась в том, чтобы наладить связь между отдельными вычислительными машинами оборонных научно-исследовательских центров путем передачи данных в виде небольших пакетов информации, которые сам ученый назвал «блоками сообщений». Конкретный механизм этой передачи был следующим – данные, предназначенные к передаче с компьютера на компьютер, разбиваются на порции (те самые пакеты), сопровождаемые дополнительной идентификационной информацией. Каждый пакет должен был путешествовать по компьютерной сети независимо от других, то есть он перебрасывался из узла в узел сети по любому доступному каналу и, в конце концов, попадал в пункт назначения.

Интересно, что идея Бэрана подразумевала то, что даже если часть каналов передачи вдруг будет разрушена (например, в результате ядерной бомбардировки), но сохранится связность сети, то данные все равно дойдут до получателя. Компьютеру последнего останется лишь собрать все пакеты воедино и восстановить то, что теперь принято называть «исходным информационным файлом». В качестве физического канала связи Бэран предложил использовать телефонный кабель.

Возможно, эта поистине революционная идея Бэрана так и осталась бы в истории науки как разработка, опередившая время, но не получившая реального воплощения (даже в наше время многие специалисты признают, что реализация разработки полвека назад была задачей, инженерная сложность которой была сопоставима со сложностью космического проекта), однако, обстоятельства в тот момент сложились весьма удачно. После запуска Советским Союзом искусственного спутника Земли в 1957 году Министерство обороны США посчитало, что на случай войны Америке нужна надежная система передачи информации. Агентство передовых оборонных исследовательских проектов США (DARPA), ознакомившись с проектом Бэрана, предложило создать для этого компьютерную сеть. Техническая реализация данного проекта была поручена Калифорнийскому университету в Лос-Анджелесе, Стэндфордскому исследовательскому центру, Университету штата Юта и Университету штата Калифорния в Санта-Барбаре.

Само собой разумеется, что раз в проекте были задействованы лучшие научно-технические центры страны, то его реализация была достаточно быстрой. Судите сами – заявка от DARPA поступила в начале 1969 года, а уже 29 октября состоялось первое испытание нового способа коммуникации. Первая сеть состояла из двух терминалов, один из которых находился в Калифорнийском университете, а второй – на расстоянии 600 км от него, в Стэндфордском университете.

Тестовое задание заключалось в том, что первый оператор вводил слово «LOG» (сокращение от «LOGON», являвшегося командой входа в систему), а второй должен был подтвердить, что он видит его у себя на экране. Первый эксперимент оказался неудачным, поскольку отобразились лишь буквы «L» и «O», после чего первая в мире компьютерная сеть «упала», то есть перестала функционировать. Через час эксперимент был повторен, и на этот раз все прошло успешно. Интересно, что наличие нужного слова на экране оператор подтвердил своему коллеге, позвонив тому по телефону. Так что 29 октября 1969 года смело можно считать днем рождения Всемирной Глобальной Паутины.

Правда, она еще достаточное время не являлась ни всемирной и ни глобальной, поскольку созданная на основе разработки Бэрана сеть (ее назвали ARPANET) поначалу объединяла лишь компьютеры военных лабораторий и нескольких научных центров США. Однако она становилась все популярнее, особенно после разработки в 1971 году первой программы для пересылки электронной почты. К ARPANET подключались все

новые организации, получавшие на это разрешение от Пентагона (поскольку данная сеть, в отличие от современного Интернета, была закрытой), и в 1973 году она наконец-то стала международной. Тогда к ARPANET были подключены через трансатлантический телефонный кабель первые иностранные организации из Великобритании и Норвегии.

Несмотря на то, что в 70-х годах прошлого века данная сеть, в основном, использовалась для пересылки электронной почты, уже тогда появились первые списки почтовой рассылки, новостные группы и доски объявлений. Однако в то время ARPANET еще не мог легко взаимодействовать с другими сетями, построенными на других технических стандартах, которые уже существовали. Но к концу десятилетия начали бурно развиваться протоколы передачи данных, которые в 1982–1983 годах были наконец-то стандартизированы.

Активную роль в данном процессе стандартизации сетевых протоколов сыграл исследователь Джон Постел, который 1 января 1983 года перевел сеть ARPANET с протокола NCP на всем нам известный TCP/IP. Интересно, что в том же году впервые прозвучало слово «Интернет», которое тогда стал синонимом сети ARPANET. А годом позже была разработана система доменных имен, которая используется и в наши дни.

Но тогда же, в 1984 году, у ARPANET появился серьезный соперник. К тому времени Национальный научный фонд США (NSF) основал обширную межуниверситетскую сеть NSFNet (сокращение от National Science Foundation Network), которая была составлена из более мелких сетей (включая известные тогда сети Usenet и Bitnet) и имела гораздо большую пропускную способность (56 кбит/сек против арпанетовских 32 кбит/сек). Неудивительно, что к этой сети за год подключились около десяти тысяч компьютеров! Поэтому почетное право именоваться Интернетом начало плавно переходить от ARPANET к NSFNet. В дальнейшем, именно он трансформировался в хорошо известный нам современный Интернет (это произошло в 1991 году, когда данная сеть стала общедоступной), в состав которого вошли множество других локальных сетей, в том числе и отечественная научная сеть, выросшая в недрах Института атомной энергии им. И.В. Курчатова и ИПК Минавтопрома (это произошло чуть раньше, в 1990 году, и с этого времени начинается история Рунета).

Что касается первопроходца по имени ARPANET, то он, будучи не в силах конкурировать с более развитым соперником, закончил свое существование в 1990 году. А вскоре ушел на пенсию и его создатель, Пол Бэран, который последние годы своей жизни практически безвыездно провел в своем доме в Пало Альто (Калифорния). В 2008 году этому замечательному исследователю президент США Джордж Буш-младший торжественно вручил Национальную медаль за вклад в развитие технологий. Эта награда была единственной, которую Бэран получил при жизни. В то же время ученый осознавал, что самой лучшей оценкой его заслуг является тот факт, что сейчас Интернетом пользуются миллионы людей во всем мире.

Возможно, Пол Бэран и не был изобретателем Интернета, как такового, но его работа над ARPANET была по истине уникальна. Всемирная сеть в том виде, в каком мы знаем ее сегодня, обязана своим появлением именно этому человеку.

pravda.ru

НОВОСТИ

«БЕЛОРУССКИЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ» СТАНУТ АЛЬТЕРНАТИВОЙ «БЕЛТЕЛЕКОМУ»

СООО «Белорусские облачные технологии» (оператор beCloud) создаст в Беларуси вторую единую систему передачи данных, чтобы появилась конкуренция «Белтелекому», в том числе в ценообразовании.

Об этом сообщил председатель совета ассоциации «Инфопарк» Владимир Анищенко участникам круглого стола «Лучшие практики электронного правительства».

Владимир Анищенко отметил, что процессы информатизации и широкого внедрения информационно-коммуникационных технологий должны протекать в рамках государственно-частного партнерства. Государство не всегда располагает свободными ресурсами, чтобы в достаточной мере финансировать процессы информатизации. Но для реализации подобного рода проектов можно найти инвесторов.

Среди основных для Беларуси проектов в области связи и информатизации – внедрение Единой республиканской сети передачи данных, запуск корневого удостоверяющего центра Государственной системы управления открытыми ключами проверки электронной цифровой подписи и скорейшее внедрение единой системы идентификации физических и юридических лиц.

Опытом внедрения электронного правительства и перехода на электронные услуги поделились грузинские эксперты. В Грузии распространены национальные ID-карты, совмещающие в себя сразу несколько решений – инструмент идентификации, средство пропуска и механизм получения доступа к электронным услугам.

В Грузии стала активно развиваться электронная инфраструктура. В частности, в стране появились службы JustDrive (по аналогии с сервисом одной из сетей быстрого питания), позволяющие автомобилисту, не покидая салона автомобиля, получить при предъявлении ID-карты юридически значимый документ. Кроме того, в стране был запущен государственный портал, предоставляющий гражданину доступ к информации о его имуществе (недвижимости, автомобиле), пересечении им границы и т.д. Также в Грузии реализован механизм электронной регистрации юридического лица. Местный бизнес перемены воспринял с большим позитивом.

СООО «Белорусские облачные технологии» (оператор beCloud) создано в Беларуси для реализации перспективных проектов в сфере информационно-коммуникационных технологий. Его учредителем с белорусской стороны выступает РУП «Национальный центр обмена трафиком» (51 % уставного фонда), с российской – бизнесмен Константин Николаев, выступающий совладельцем крупных транспортных компаний.

ej.by

ИСТОРИЯ РАЗВИТИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

Компьютерные сети, называемые сетями передачи данных, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации компьютерных и телекоммуникационных технологий. Многотерминальные системы, работающие в режиме разделения времени, стали первым шагом на пути создания локальных вычислительных сетей. Первыми появились глобальные сети, то есть сети, объединяющие территориально рассредоточенные компьютеры, возможно, находящиеся в различных городах и странах.

В 1969 году Министерство обороны США инициировало работы по объединению суперкомпьютеров в единую сеть оборонных и научно-исследовательских центров. Эта сеть, получившая название ARPANET, послужила отправной точкой для создания Интернет. Сеть ARPANET объединила компьютеры разных типов, работающих под управлением различных операционных систем с дополнительными модулями, реализующие коммутационные протоколы, общие для всех компьютеров в сети.

Локальные сети – объединение компьютеров, сосредоточенных на небольшой территории, обычно в радиусе не больше двух километров. В общем, сетевая технология – это согласованный набор программных и аппаратных средств (драйверов, сетевых адаптеров и т.д.) и механизмов передачи данных по линиям связи, достаточных для построения вычислительной сети.

История создания сети Интернет

В 1957 году в США, по указанию президента Дуайта Эйзенфаура, в составе Отдела Обороны (DoD, Department of Defence) формируются два правительственных органа: Национальная аэрокосмическая администрация NASA (National Aeronautics and Space Administration), которая в представлении не нуждается, а также Агентство по Передовым Оборонным Исследованиям (DAPRA или Defence Advanced Research Projects Agency). Сделано это было с целью продвижения военных технологий США на лидирующие позиции в мире.

Прогресс человечества и военные технологии всегда идут вместе, поэтому весь начальный этап развития Сети будет связан с военным ведомством США. В начале 60-х годов основные работы DAPRA были посвящены разработке метода соединений компьютеров друг с другом. Агентство выделяет денежные средства для привлечения к перспективным разработкам университетов и корпораций (Массачусетский Технологический Институт – MIT – некоммерческая организация, занимающаяся стратегическими исследованиями и разработками RAND Corporation).

В 1962 году Дж. Ликлайдер (J.C.R. Liclider) публикует работу «Galactic Network», в которой предсказывает возможность существования в будущем глобальной компьютерной связи между людьми, имеющими мгновенный доступ к программам и базам данных из любой точки земного шара. Как это не удивительно, его предвидение в полной мере отражало современное устройство всемирной Сети. Тогда же, в августе 1962 года, вышла статья Дж. Ликлайдера и В. Сларка «Интерактивная связь человека с компьютером».

Возглавив первую исследовательскую программу, начатую DAPRA 4 октября 1962 года, Ликлайдер сумел увлечь своей концепцией группу ученых, среди которых был и его преемник-исследователь из MIT Лоуренс Робертс (Lowrence G. Roberts), а также Иван Сазерленд (Ivan Sutherland) и Боб Тейлор (Bob Taylor).

В июле 1961 года Леонард Клейнрок (Leonard Kleinrock) разработал и впервые опубликовал статью «Информационный поток в крупных коммутационных Сетях», где представил новую теорию передачи данных. Это была первая публикация по теории коммутаций пакетов. В 1964 году новая концепция вышла уже в книге. Тогда же Л. Клейнрок убедил Л. Робертса в возможности коммуникаций с использованием пакетов и в преимуществах своей теорий перед древнейшим принципом коммутаций каналов. Как известно, при пакетной коммутации необходимые для передачи данные разбиваются на фрагменты, к каждому из которых присоединяется заголовок (адрес), содержащий полную информацию о доставке пакета по назначению. В результате один канал связи может использоваться для одновременной передачи данных множества пользователей, тогда как при коммутации каналов, широко используемой в традиционной телефонной связи, канал связи выделяется исключительно к услугам двух пользователей, расположенных на его концах.

Для проверки новой концепции пакетной коммутации Л. Робертс и Т. Мерилл еще в 1965 году соединили компьютер TX-2 в штате Массачусетс (MIT, Лабораторий Линкольна) с компьютером Q-32 в System Development Corporation (Санта-Моника, Калифорния) с помощью низкоскоростных телефонных коммутируемых линий (пока еще без коммутаций пакетов).

Таким образом, в 1965 году в США была создана первая в истории маленькая, но вовсе даже не локальная компьютерная сеть. Результатом эксперимента стало понимание того, что компьютеры могут успешно работать вместе, выполняя программы и осуществляя выборку данных. Стало также ясным и то, что телефонная сеть с коммутацией каналов абсолютно непригодна для построения компьютерной сети. Разумеется, Л. Клейнрок еще раз убедился в необходимости пакетной коммутации, и это было в тот момент самым главным.

В конце 1966 года DARPA пригласило Л. Робертса для реализации проекта компьютерной сети ARPANET. Целями проекта были объединения исследовательских учреждений, проведение экспериментов в области компьютерной коммуникации, а также изучение способов поддержки надежной связи в условиях ядерного нападения.

Итак, Л. Робертс начал работать над разработкой концепции децентрализованного (распределенного) управления военными и гражданскими объектами в период ведения войн. Довольно быстро появился план ARPANET. В 1967 году на симпозиуме по Принципам

Взаимодействия (Operating Principles), организованной Ассоциацией машинных вычислений (ACM, Association for Computing Machinery), которая была основана еще 1947 году и является первым научным и образовательным компьютерным сообществом, был представлен проект сети с коммутацией пакетов. И тогда же, в 1967 году первое издание проекта ARPANET опубликовано Л. Робертсом.

В 1964 году группа сотрудников RAND Corporation написала статью по сетям с пакетной коммутацией для надежных голосовых коммуникаций в военных системах. Работы, которые проводились в середине 60-х годов в MIT, RAND и NPL, были во многом параллельными, и эти организации не имели информации о деятельности друг друга. Разговор Л. Робертса с сотрудниками NPL увенчался заимствованием слова «пакет» и решением увеличить предлагаемую скорость передачи по каналам проектируемой сети ARPANET с 2,4 Кб/с до 50 Кб/с.

В конце 1969 года в одну компьютерную сеть были включены четыре исследовательских центра:

University of California Los Angeles (UCLA);

Stanford Research Institute (SRI);

University of California at Santa Barbara (UCSB);

University of Utah.

В октябре 1969 года было послано первое электронное сообщение между узлами UCLA (Калифорнийский Университет, Лос-Анджелес) и SRI (Исследовательский Институт Стэнфорда). Говорят, что в самом начале работы эта сеть сразу же «зависла», но процесс пошел.

Вот так четыре удаленных компьютера были объединены в первоначальную конфигурацию ARPANET. Одновременно Р. Кан разработал общую архитектуру сети ARPANET, Л. Робертс разработал топологию и экономические вопросы, Л. Клейнрок представил все средства измерений и анализа сети.

djamaev-mtt.hut2.ru

НАДЕЖНОСТЬ – ПРИОРИТЕТ

APS Energia SA – это один из наиболее крупных на польском рынке поставщиков индивидуальных решений в сфере систем гарантированного питания для промышленности. Основным видом деятельности APS Energia является разработка и производство широкого спектра энерго-электронного оборудования для применения в энергетике, промышленности, телекоммуникации, медицине и других секторах экономики. Надежность и стабильность снабжения электроэнергией является основной задачей компании.

В дополнение к спектру производимого оборудования, APS Energia SA обеспечивает комплексное проектирование, монтаж, пуско-наладочные работы, гарантийное и послегарантийное обслуживание систем электропитания.

APS Energia предлагает: системы аварийного электроснабжения для приемников постоянного и переменного тока, выпрямители, буферные источники питания, инверторы, преобразователи частоты, активные фильтры, системы питания, системы мониторинга энергоблоков и топливные элементы.

Основными потребителями продукции являются клиенты, для которых надежность работы и высокие технические параметры являются приоритетом. Предлагаемые системы аварийного электроснабжения обеспечивают непрерывность производственных процессов и возможность работы в случае отказа основного питания. Таким образом, APS Energia помогает клиентам исключить риск потерь в случае аварии в системе питания. Кроме того, каковы бы ни были нужные для Вашего уникального оборудовании параметры нестандартной питающей сети – APS Energia сумеет обеспечить их за разумную цену.

«Устройства APS Energia – это лучшие технические параметры, современные технологии и инновационные решения. Благодаря этому мы опережаем конкурентов, добиваясь лучших результатов там, где серийная продукция не имеет никаких шансов, чтобы удовлетворить индивидуальные потребности наших потребителей. APS Energia предлагает системы аварийного электроснабжения, технические параметры и конфигурации которых мы приводим в соответствие с потребностями и ожиданиями клиента. Высокое качество продукции и услуг APS Energia подтверждается полученными сертификатами. Устройства APS Energia отвечают требованиям польских и европейских стандартов по безопасности, электромагнитной совместимости, а также директивам ЕС (знак CE)» – говорит Президент компании др. инж. Петр Шевчик.

В течение 18 лет компания APS Energia осуществляет и развивает свою деятельность в Польше и за рубежом. Нашими заграничными партнерами являются: рынки Восточной Европы (Россия, Казахстан, Азербайджан, Литва, Белоруссия, Узбекистан, Грузия), рынки Европейского Союза (Чешская Республика, Словакия, Испания, Нидерланды) и мировые рынки (Саудовская Аравия, Австралия, Ирак, ЮАР, Египет, Кувейт, Македония). Благодаря развитой сети дистрибьюторов и посредников производимое компанией оборудование безотказно работает на многих энергообъектах во всем мире.

Удовлетворять потребности требовательных заказчиков APS Energia в Беларуси помогают высококвалифицированные инженеры фирмы ООО «Электротехническая компания «ЭКНИС», прошедшие обучение и имеющие большой опыт работы с оборудованием APS Energia. Именно благодаря наличию местной сервисной группы можно гарантировать быстрое решение всех проблем, хотя и возникают они крайне редко.

О БУДУЩЕМ КОМПЬЮТЕРНЫХ СЕТЕЙ

Компьютерные сети переживают глубокий кризис, связанный с тем, что лежащая в их основе архитектура безнадежно устарела. Несколько лет назад возникло движение по созданию сетей нового поколения.

Лежащая в основе современных сетей архитектура закладывалась полвека назад, когда мир был совсем другим. Объемы трафика растут гораздо быстрее, чем увеличивается пропускная способность систем передачи данных, а его структура все время усложняется.

О перспективных методах решения этих проблем рассказал директор по науке и образованию Центра прикладных исследований компьютерных сетей (ЦПИ КС), член-корреспондент РАН, профессор МГУ им. М.В. Ломоносова, д.ф.-м.н. и заслуженный деятель науки Российской Федерации Руслан Смелянский.

– Не могли бы вы рассказать о вашем центре: когда он был создан и каковы его цели и задачи?

– Центр юридически был создан в январе 2012 года и уже в феврале ему был присвоен статус резидента «Сколково». Мы стали, насколько мне известно, первой некоммерческой организацией-резидентом, которая полностью ориентирована на научные исследования.

Основных направлений деятельности три. Во-первых, это установление партнерских взаимоотношений с международными исследовательскими центрами. Второе направление нашей деятельности – работа с НИИ Академии Наук и российскими университетами с целью инициировать там научные исследования в области компьютерных сетей. Третье направление деятельности – это продвижение новых технологий, демонстрация их преимуществ нашей отечественной промышленности, создание совместных проектов и проверка эффективности этих технологий на реальных задачах.

– Говоря о технологиях, вы имеете в виду только программно-конфигурируемые сети (SDN), или Центр занимается чем-то еще?

– Мы занимаемся компьютерными сетями, которые сегодня переживают глубокий кризис. Если очень кратко, то причина этого кризиса, во-первых, в том, что лежащая в основе современных сетей архитектура закладывалась в конце шестидесятых – начале семидесятых годов прошлого века и безнадежно устарела.

Во-вторых, недавно в области компьютерных сетей появились новые тенденции. Темпы роста трафика и структура этого трафика стали существенно обгонять те возможности, которые предоставляют системы передачи данных. Раньше темпы роста пропускной способности систем передачи данных практически удваивались каждый год, и это всех устраивало. Сейчас объем трафика увеличивается существенно быстрее пропускной способности каналов, и это начинает создавать проблемы.

И наконец, в-третьих, что является, в какой-то степени, следствием первого: компьютерные сети – одна из наиболее сложных инженерно-технических систем, поведение которой сложно предсказать. Достаточно сказать, что в современных сетях активно используется более шестисот разных протоколов. Связать их между собой по управлению и данным надежно и безопасно – это задача чрезвычайно трудная. Именно поэтому администрирование компьютерных сетей – скорее, искусство, нежели инженерия. И понятно, что с таким положением уже никто не хочет мириться.

Лет пять или шесть назад в США возникло движение по созданию компьютерных сетей (или, если хотите, Интернета) нового поколения. И начались поисковые исследования. Одно из таких направлений, которое на сегодняшний день стало лидирующим – программно-конфигурируемые сети (software defined network). Его мы и развиваем сейчас в России.

– Если конкретизировать проблемы компьютерных сетей, то четко вырисовываются две проблемы. Это отставание темпов роста пропускной способности каналов от темпов роста трафика и сложная гетерогенная структура, которой очень тяжело управлять. Это основные факторы?

– Не только это. Есть и третий фактор. На сегодняшний день оборудование, с помощью которого строятся компьютерные сети, является проприетарным. То есть для системного администратора это «черный ящик», в который он втыкает провода и функциональностью которого он пользуется. Там есть прошитый производителем софт, но пользователь без производителя ПО сменить не может.

– Но разве это проблема? Допустим, оборудование – «черный ящик», но все входы и выходы из этого ящика описаны – протоколы стандартизированы.

– Протоколов более шестисот. Действительно, каждый из них имеет стандарт, но подавляющее большинство используемых в Интернете протоколов стандартом де-юре не является.

– Но это, в основном, протоколы уровня приложений?

– Вы глубоко заблуждаетесь. Протоколы TCP/IP не являются стандартами де-юре с точки зрения международного права – это стандарты де-факто.

– Но они являются отраслевыми техническими стандартами де-факто, описанными в RFC.

– Они не являются отраслевыми стандартами. Понятие отраслевого стандарта совсем другое. Они являются стандартами де-факто, выпущенными неким образованным в США society и ставшими международными. Но де-юре – это не стандарты.

Мало того, чтобы согласовать протоколы между собой, одного стандарта недостаточно. Он не определяет многих технических параметров и не описывает (да и не должен описывать, чтобы не связывать по рукам и ногам производителей) очень многих деталей. Даже при наличии стандарта могут возникать ситуации нестыковки протоколов между собой. Но самое главное, организация сетей в виде «черных ящиков» блокирует возможности проведения экспериментов с новыми протоколами.

– Один из предлагаемых вариантов решения этих проблем – программно-конфигурируемые сети. В чем суть этой технологии?

– Проблем, конечно, не три, их намного больше. В основной массе использующихся сейчас протоколов смешаны два разных вида данных: управляющие (с помощью которых происходит управление обменом информации) и, собственно, передающие данные. Программно-конфигурируемые сети их разделяют, то есть управление здесь происходит по совершенно другим каналам, нежели, сама передача.

Кроме того, передача данных идет не через отдельно маршрутизируемые пакеты, как это сделано, скажем, в IP, а в виде потока. Если мы вспомним IPv4 и IPv6, то там есть такое поле – идентификатор потока. Здесь оно существенно используется.

– Но данные одного потока все равно оформлены в виде пакетов?

– Совершенно верно, данные идут по виртуальному соединению, которое маршрутизируется в момент своего создания. И дальше все относящиеся к нему пакеты идут по одному маршруту. Как только появляется пакет, маршрутизатор смотрит, знает ли он поток, к которому тот относится. Если устройство в своих таблицах маршрутизации находит соответствующие правила (то есть, знает этот поток), то пакет обрабатывается в соответствии с ними. Если такого правила нет, то устройство обращается к так называемому контроллеру, который принимает решение об открытии нового потока и формирует правила. Они загружаются с помощью протокола Openflow в таблицы маршрутизатора и используются при обработке всех пакетов нового потока.

Таким образом, решается очень серьезная проблема – на ранних этапах развития Интернета было принято решение, что каждый пакет может маршрутизироваться самостоятельно. Оно было следствием сформулированной в конце шестидесятых годов в DARPA проблемы: военным были нужны высоковживучие системы передачи данных, чтобы уничтожение отдельного узла не приводило к выходу из строя всей сети. Пока каналы были медленными и потоки данных небольшими – это было приемлемо.

Каждый узел сети при этом должен решать две задачи: построение маршрута для пакета и коммутация пакета в соответствии с принятым маршрутом. С точки зрения вычислительной сложности эти задачи принципиально разные. Проблема была осознана только в девяностые годы, и люди стали пытаться их разделить, то есть вытащить все, что связано с расчетом маршрута и его оптимизацией, в отдельные специализированные сервера, оставив основной части узлов сети только задачи маршрутизации.

Именно на этом принципе построены так называемые MPLS-сети или маршрутизация по меткам. В программно-конфигурируемых сетях решение доведено до логического конца – сформулировано понятие сетевой операционной системы, управляющей ресурсами сети (маршрутизаторами, свичами, линками), как традиционная ОС управляет ресурсами компьютера. Поверх нее работают различные приложения: по маршрутизации, по балансировке потоков, по оптимизации пропускной способности каналов и т. д. Эти приложения работают на так называемых контроллерах, а остальные устройства просто решают задачи маршрутизации и коммутации.

– Насколько я понимаю, здесь активно используются технологии виртуализации? Живучесть контроллера (на котором очень многое завязано) мы не теряем?

– Естественно, этот подход открывает новые возможности по виртуализации.

– Но разве программно-конфигурируемые сети могут решить проблему с пропускной способностью каналов? Ведь, скажем, 2 Мб есть 2 Мб и ничего с этим не сделаешь на программном уровне.

– Они решают задачу балансировки нагрузки и более эффективного использования каналов, подобно тому, как виртуальные машины позволяют повысить эффективность использования вычислительных ресурсов дата-центров. Здесь, то же самое – более эффективно используются имеющиеся каналы, ресурсы коммутаторов и другого оборудования. Проблема недостаточно быстрого роста пропускной способности каналов решается за счет динамического перераспределения нагрузки.

– Перейдем к экономическим аспектам проблемы. Что в данном случае важнее: технические или экономические причины?

– Промышленность никогда не будет инвестировать в то, что не приносит доход. Основной экономический фактор здесь – на 30 % снижается стоимость владения компьютерными сетями за счет сокращения расходов на управление. И за счет динамической балансировки и перераспределения нагрузки вы можете повысить на 20 % фактическую пропускную способность своих каналов. А значит – пропустить больше трафика и заработать больше денег.

– А какие-то недостатки и узкие места у программно-конфигурируемых сетей есть?

– Конечно. Это новое явление и у него пока отсутствует то, что называется научным бекграундом – здесь еще нужно много работать математикам по построению соответствующего аппарата. Кроме того, не до конца решены вопросы построения распределенных контроллеров – сетевые операционные системы пока не обладают должной гибкостью.

– Но ведь есть наработки, которые уже пошли в серию? Насколько мне известно, протокол OpenFlow поддерживается рядом крупных производителей сетевого оборудования.

– Количество производителей, которые выпускают поддерживающее OpenFlow оборудование, растет постоянно. На сегодняшний день это и Nec, и HP, и Huawei, и Juniper, и Marvell – их много. IBM заявила о поддержке OpenFlow – эта когорта постоянно растет.

– SDN, если по-русски ПКС, базируются на открытых технологиях и стандартах?

– Есть уже несколько версий открытого стандарта OpenFlow.

– Но реализации протокола у разработчиков проприетарного железа могут быть основаны на закрытом коде?

– Этот протокол позволяет контроллеру общаться с активным сетевым оборудованием. Важно, чтобы сформулировавшее политику маршрутизации приложение могло загрузить соответствующее правило в память устройства (свича или маршрутизатора), – оборудование должно быть программно управляемым. Вот эта загрузка происходит по открытому протоколу OpenFlow. Само устройство может быть проприетарным, но, главное, оно становится программируемым. И, что самое важное – оно программируемо вами – вы сами описываете политику.

– Что изменится, с точки зрения конечного пользователя, с массовым внедрением программно-конфигурируемых сетей?

– Нововведения коснутся, в первую очередь, активного сетевого оборудования на стороне оператора, а также корпоративных сетей. Домашним же пользователям могут предложить новые услуги, например, администрирование сетей.

– То есть оборудование с поддержкой OpenFlow будет доступно и частным лицам?

– Вы можете поставить SDN-свич у себя дома и прописать, как вы хотите связать разные устройства. Например, если в квартире несколько комнат, можно перенаправить аудиопоток из одной в другую. Более того, провайдер сможет удаленно вам помогать, устранять проблемы, подгружать софт с новым функционалом и т.д.

– И, наверное, SDN позволит снизить тарифы?

– Прежде всего, технология позволит снизить себестоимость услуг для провайдера. И отсюда повышение рентабельности его бизнеса. Одна из проблем заключается в том, что стоимость услуг повышать из-за жесткой конкурентной борьбы нельзя, а их себестоимость постоянно растет, потому что дорожает труд, оборудование усложняется, повышается его стоимость и т.д.

– Каковы ваши прогнозы развития ситуации? Настолько быстро будут внедряться технологии SDN?

– Думаю, в ближайшие три-пять лет это станет общей практикой. Если на рынке начались процессы слияния и поглощения, то есть происходит покупка занимающихся программно-конфигурируемыми сетями компаний (причем, за ощутимые деньги), то тенденции очевидны. Самое главное для нас, что открылось окно – если мы в ближайшие годы сможем наладить производство своих продуктов (хотя бы софтверных), то из потребителя технологий Россия превратится в поставщика.

– И дело не только в бизнесе – это ведь связано и с технологической независимостью и национальной безопасностью?

– Вы абсолютно правы, потому что ПКС-сети позволяют решать многие вопросы безопасности более эффективно, чем традиционные. Что касается национальной безопасности – достаточно привести пример Бушерской АЭС – в 2010 году атомная программа Ирана была в центре международного внимания. Если вредоносный код может выводить из строя технологическое оборудование, то кибервойны вполне реальны.

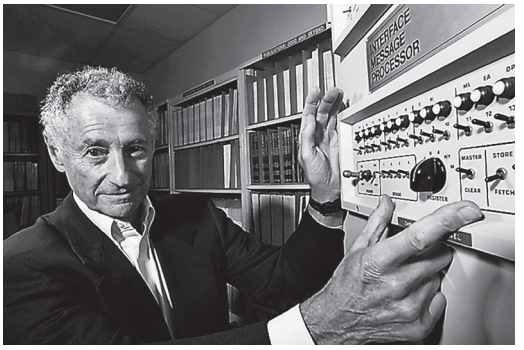
Евгений Крестников
old.computerra.ru

«ИНТЕРНЕТ, КАК ВЕЗДЕСУЩАЯ И НЕВИДИМАЯ СЕТЬ»

Леонард Клейнрок является профессором вычислительной техники Калифорнийского университета в Лос-Анджелесе.

Принципы коммутации пакетов, разработанные им в 1961 году, стали предвестниками глобальной Сети, а в 1969 году его компьютер стал первым Интернет-узлом.

Леонард Клейнрок является основателем и председателем компании Nomadix Inc., предоставляющей широкий спектр услуг по широкополосному доступу в Интернет.



– Что подтолкнуло вас начать научную деятельность в области сетевых технологий?

– Будучи в 1959 году студентом, я заметил, что большинство моих сокурсников стали заниматься исследованиями в области теории информации и кодирования. В нашем университете работал замечательный исследователь Клод Шеннон, в область научных интересов которого входили подобные проблемы. К тому времени у него уже было много наработок в этой области, а нерешенные задачи представляли большую сложность и к тому же, как мне казалось, не были принципиальными. Мне хотелось начать деятельность в той области, в которой пока не было фундаментальных разработок. В процессе учебы меня окружало множество вычислительных машин, и я понял, что недалек тот час, когда потребуется организовать общение между ними. В то время еще не было эффективных средств, позволяющих это сделать, поэтому я решил разработать технологию эффективного обмена данными между компьютерами.

– Какова была ваша первая работа в области компьютерных технологий? Какую роль она сыграла для вас?

– В 1951–1957 годах я учился на бакалавра по специальности инженер-электрик. Сначала я работал техником в маленькой промышленной компании Photobell, занимавшейся электроникой, а через некоторое время получил в ней должность инженера. За это время я успел разработать цифровую технологию для линии продуктов компании. В сущности, мы занимались интеграцией цифровых технологий в систему детекции с использованием фотоэлектрических элементов. Позже наши разработки трансформировались в то, что теперь принято называть коммутаторами.

– Каковы были ваши впечатления от первого опыта передачи сообщения?

– Откровенно говоря, я был разочарован. Гораздо более приятным воспоминанием для меня служит 2 сентября 1969 года, когда интерфейсный процессор сообщений впервые соединился с моим компьютером в Лос-Анджелесе. Эту дату я считаю началом Интернета. Годом ранее я заявил, что, как только нам удастся обеспечить стабильную работу сети, мы сможем предоставить доступ к ней из наших домов и офисов так же легко, как к обычной телефонной или электрической сети. В то время я представлял себе перспективу Интернета как вездесущей и невидимой сети, доступной каждому человеку в любой точке пространства в любое время. Но я не мог и предположить, что моя 94-летняя мама будет пользователем Интернета, тем не менее, это так!

– Каковы ваши прогнозы на будущее компьютерных сетей?

– Наиболее важные направления развития – мобильные технологии связи и, как следствие, устройства для их применения. Легкие, недорогие, высокотехнологичные портативные устройства приобретают все большую доступность. Мобильные технологии – это средства, позволяющие пользователям получать доступ к компьютерной сети и ее службам без территориальных ограничений. Но я думаю, что это только первый шаг в раскрытии истинного потенциала компьютерных сетей. Далее компьютерные сети в виде сенсорных датчиков, цифровых камер, микрофонов, дисплеев, накапливающих и обрабатывающих информацию устройств проникнут в то, что мы называем окружающей обстановкой: в стены, в предметы интерьера, в машины и т.д. Например, когда я буду заходить в комнату, она будет «знать» о том, что я вошел. Я смогу общаться с окружающими предметами также, как с живыми людьми, а они будут отвечать мне, подобно серверам, у которых я запрашиваю нужную мне web-страницу.

С технологической точки зрения, на мой взгляд, произойдет распространение множества ключевых сетевых компонентов. Так, к примеру, возможно создание интеллектуальных программных систем, которые будут самостоятельно заниматься сбором информации, ее обработкой, проведением сложных аналитических операций и динамическим выполнением назначенных заданий. Большая часть сетевого трафика будет генерироваться не людьми, а различными встроенными устройствами, в частности, этими интеллектуальными системами. В целом, Интернет станет во многом саморегулирующейся системой. Его ожидают огромные потоки информации, передающейся в разные концы планеты, где эта информация будет подвергаться обработке и фильтрации. Интернет способен в будущем стать гигантским «живым» организмом, способным воспринимать, думать и реагировать на действия людей.

– Кто были вашими профессиональными вдохновителями?

– Я многим обязан Клоду Шеннону, замечательному исследователю, благодаря своей потрясающей интуиции воплотившему в реальную жизнь абстрактные математические выкладки. Он был одним из членов ученого совета, принимавшего мою диссертацию.

– Можете ли вы дать совет молодым студентам, желающим стать специалистами в области компьютерных сетей?

– Интернет с его богатым потенциалом является замечательным полем для научной деятельности. Не нужно думать, что технологии сегодняшнего дня – предел совершенства. Ставьте себе новые цели и достигайте их!

conlex.kz

ОРГАНИЗАЦИЯ СБОРА ДАННЫХ СОЦИАЛЬНОЙ НАПРАВЛЕННОСТИ НА ПРИМЕРЕ ДАННЫХ МИКРОБЛОГОВ «TWITTER»

А.В. Ахрамович, г. Минск

Введение

Развитие информационных технологий и информатизация общества приводит к генерации все большего объема данных социального общения и взаимодействия личностей в рамках глобальной сети хранения, обмена и обработки информации Интернет. Современный уровень использования «глобальной паутины» уже вышел за рамки рассмотрения сети как средства получения информации, продвижения новых продуктов, обмена данными. Основным эволюционным признаком Интернета стало социальное ориентирование на взаимодействие между индивидами (открытые социальные данные) и группами индивидуумов (закрытые сообщества, корпоративные сети).

В каждой социальной сети существуют специализированные механизмы взаимодействия между учетными записями. Эти механизмы определяют динамику генерации информации социального общения, а также ее основные типы данных.

В соответствии с последним отчетом международного союза электросвязи (ITU) общее количество пользователей, которые, так или иначе, пользуются сервисами, содержащими элементы социальной сети, превысило один миллиард. С каждым годом количество пользователей, использующих мобильные устройства для доступа к социальным сетям, увеличивается, что отражается на частоте обновления информации в социальных сетях. Использование социальных сетей присуще не только обыкновенным пользователям. Большинство политиков, представительств компаний, средств массовой информации, так или иначе, представлены в социальных сетях. В наше время практически каждое важное событие, происходящее в мире, обширно обсуждается на различных информационных ресурсах с элементами социальных сетей. Таким образом, важность социальных сетей в жизни современного общества трудно недооценить.

Данные социальных сетей представляют огромный интерес для аналитиков. Например, при анализе предпочтений пользователей существует возможность построения маркетинговой программы для выбранного продукта. Анализируя графы общения пользователей можно выявить группы людей со схожими интересами, увлечениями и политическими взглядами. Данные результаты могут быть использованы для проведения социальных и политических программ. Данные, накопленные социальными сетями, а также результаты их анализа можно применять при проведении розыскных мероприятий в правоохранительных органах или же при построении психологического портрета личности.

На сегодняшний день данные социальных сетей представлены в открытом доступе в виде, не позволяющем проводить их анализ напрямую. Для возможности проведения анализа, данные социальных сетей необходимо собрать, агрегировать и обработать. Именно качество полученного хранилища данных будет впоследствии определять максимально возможный уровень и достоверность анализа.

Концепция социальных сетей

Социальная сеть – платформа, онлайн-сервис или Интернет-ресурс, предназначенные для построения, отражения и организации социальных взаимоотношений. Термин «социальная сеть» был введен в 1954 году социологом из «Манчестерской школы» Джеймсом Барнсом в работе «Классы и собрания в норвежском островном приходе», вошедшую в сборник «Человеческие отношения». Он развил и дополнил изобретенный в 30-е годы подход к исследованию взаимосвязей между людьми с помощью социограмм, то есть визуальных диаграмм, в которых отдельные лица представлены в виде точек, а связи между ними – в виде линий. На этом подходе основана социометрия – психологический метод для выяснения распределения ролей в коллективе. К 70-м годам окончательно сформировался комплекс социологических и математических методов исследований, которые составляют научный фундамент современного анализа социальных сетей [1].

Сеть социальных взаимодействий состоит из совокупности пользователей и набора связей между ними. В качестве пользователей могут выступать индивиды, социальные группы, организации, города, страны. Под связями понимаются не только коммуникационные взаимодействия между пользователями, но и связи по обмену различными ресурсами и деятельностью, включая конфликтные отношения. Полученная сеть взаимодействий может быть проанализирована различными методами теории графов, теории информации, математической статистики.

Можно выделить типичные и индивидуальные особенности социальных сетей. К типичным особенностям социальных сетей можно отнести:

– возможность создавать общедоступные и закрытые индивидуальные профили, в которых будет содержаться определенная информация о пользователе;

– возможность формировать список пользователей, имеющих связи определенного вида (например, дружба или наличие общих интересов);

– взаимодействие пользователей (посредством просмотра профилей друг друга, внутренней почты, комментариев и пр.);

– возможность достижения совместной цели путем кооперации (например, целью социальной сети может быть поиск новых друзей, ведение группового блога и пр.);

– обмен ресурсами (например, ссылками на сайты);

– возможность удовлетворения потребностей за счет накопления ресурсов (например, путем участия в социальной сети можно обзаводиться новыми знакомыми и тем самым удовлетворять потребность в общении).

К распространенным индивидуальным особенностям социальных сетей можно отнести:

– возможность пользователям объединяться в группы в соответствии с общими интересами или принадлежностями к различным группам в реальном мире;

– возможность общаться в более сложной форме (например, посредством потокового видео или через форумы);

– наличие элементов геолокации (например, в некоторых социальных сетях вы можете получить географическую информацию о месте, в котором находился интересующий вас пользователь).

Twitter

Twitter – система, позволяющая пользователям отправлять короткие текстовые заметки (до 140 символов), используя веб-интерфейс, SMS, средства мгновенного обмена сообщениями или сторонние программы-клиенты. Отличительной особенностью социальной сети Twitter является публичная доступность размещенных сообщений, что роднит его с блогами.

Запуск системы состоялся в 2006 году. Сегодня сервис насчитывает более 500 миллионов пользователей. Более 140 миллионов пользователей проявляют активность хотя бы раз в месяц. В день создается более 175 миллионов новых сообщений.

Пользователи могут объединять группу сообщений по теме или типу с использованием hashtag – слова или фразы, начинающиеся с#. Кроме того, буква d перед именем пользователя позволяет отправлять сообщения в частном порядке, приватно. Наконец, знак @ перед именем пользователя используется для упоминания или ответа другим пользователям. В конце 2009 года была добавлена функция «Списки Twitter», что позволяет пользователям следить (а также отмечать и отвечать) за списками авторов, а не за отдельными авторами.

Стратегия сбора данных

Процесс сбора данных из внешних источников можно разделить на три этапа: извлечение данных из различных внешних источников; очистку данных, преобразование и трансформация для наилучшего описания выбранной модели; загрузку данных в хранилище [2].

Извлечение данных

На этом этапе данные извлекаются из одного или нескольких источников и подготавливаются к этапу преобразования. Среди особенностей процесса извлечения данных можно выделить:

– идентификация источников данных;

– определение метода извлечения данных;

– определение частоты и условий извлечения данных.

Существуют две различные стратегии извлечения данных для анализа социальных сетей. Первая направлена на получение данных о взаимодействиях всех пользователей сети, например, всех компаний на рынке информационных систем. В этом случае возникает проблема определения границ сети: если существует сторонний пользователь, интенсивно взаимодействующий с представителями данной сети, необходимо определить, являются ли эти взаимодействия существенными для структуры сети, и, если это так, то указанный пользователь должен быть включен в состав анализируемой социальной сети. В то же время ее границы могут задаваться границами исследуемой социальной общности, например, социальная сеть может содержать информацию о взаимодействиях землевладельцев только в пределах данного района страны, при этом их внешние связи не будут рассматриваться.

Вторая стратегия заключается в сборе данных обо всех взаимодействиях, в которые включен определенный пользователь. Эта стратегия особенно часто используется при построении социальной сети по результатам опроса респондентов. В этом случае мы можем получить полную информацию о взаимодействиях респондента за определенный промежуток времени и, сравнивая структуры этих взаимодействий, выявить их зависимость от характеристик пользователей и внешних воздействий.

Очистка, преобразования и трансформация данных

Для правильной обработки в рамках проведения анализа часть данных необходимо должным образом обработать и трансформировать в различные формы. Это связано с тем, что в силу тех или иных причин данные хранятся в операционных базах данных в непригодном для анализа виде.

Фильтрация данных. В задачу процесса фильтрации данных входит удаление дубликатов записей, неизбежно присутствующих на первом уровне хранилища, с учетом даты последнего обновления. Например, при необходимости подготовки итогового хранилища, содержащего данные, относящиеся к конкретному пользователю и его окружению, необходимо данные, хранящиеся в хранилище первого уровня, отфильтровать в соответствии с ограничениями запроса конечного пользователя. Это также является задачей процесса фильтрации данных [3].

Нормализация данных. Информация, хранимая на первом уровне хранилища, зачастую находится в ненормализированом виде. Так, например, упоминания имени одного пользователя в сообщениях, созданных в разное время, может отличаться, в связи с тем фактом, что пользователь мог изменить свое имя в заданный период.

Агрегация данных. Зачастую информация, полученная при помощи Twitter API, не предоставляет особого интереса для последующего анализа в связи со своей неполнотой. В таких случаях данные подвергаются опереженной постобработке с целью получения полной информации об объекте.

Так, например, для адресов внешних ресурсов, присутствующих, как в профиле пользователя, так и в самих сообщениях, можно и необходимо получить ключевые слова, заголовок, описание, а также рейтинг данного ресурса в глобальной поисковой сети Google. Схема данного процесса представлена на рисунке 1.

В качестве источников дополнительных данных можно использовать сторонние веб-сервисы (такие как сервисы от компании Google, Microsoft, Wolframalpha). Также некоторые данные можно анализировать напрямую (например, семантический анализ сообщений пользователей или получение информации о внешнем ресурсе, анализируя его исходный код).

Таким образом, фильтрация, нормализация и дополнение данных позволяют получить итоговое хранилище данных в наиболее удобной для последующей работы форме.

Загрузка данных

Данный этап предполагает запись преобразованных, интегрированных и очищенных данных в соответствующую систему хранения. Основной проблемой на этом этапе ста-

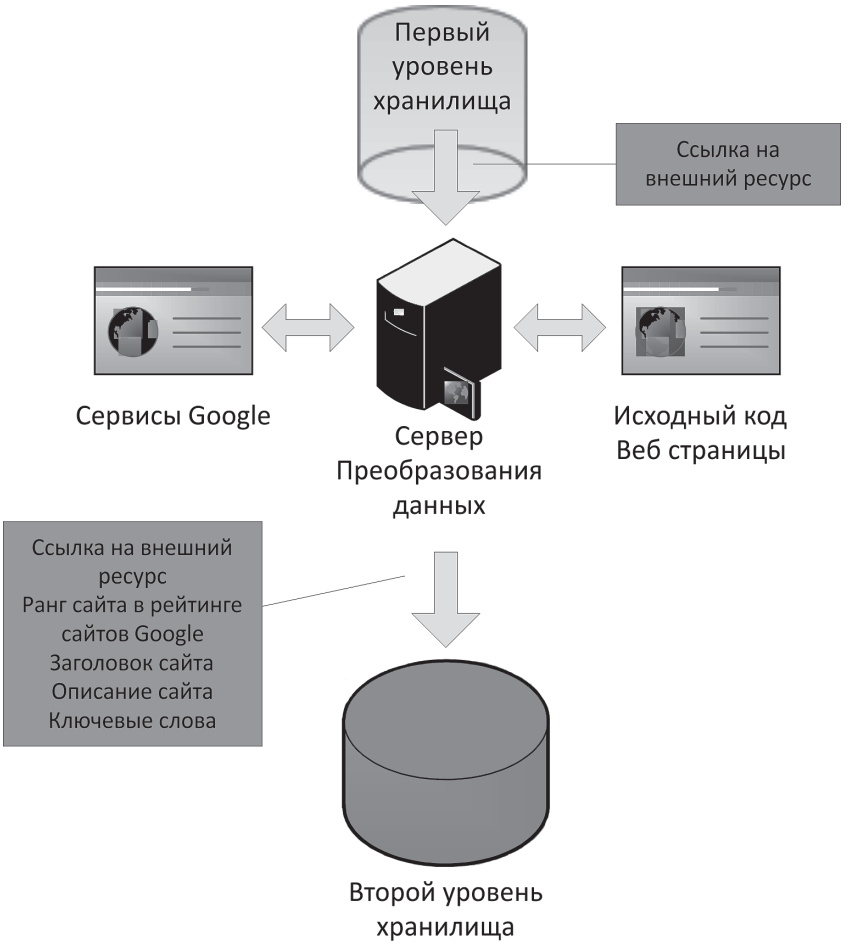


Рисунок 1 – Пример дополнения данных

новится определение типа обновления данных в хранилище, а также частота этого обновления.

По виду загрузки данных чаще всего выделяют следующие классы процессов:

– процесс начальной загрузки (Initial load);
– процесс обновляющей загрузки (Refreshing load).

Обзор существующих средств сбора данных

На данный момент в Интернете представлены различные специализированные средства для сбора данных из внешних источников. Среди них есть как открытые, так и закрытые программные средства. Наиболее распространенными программными средствами являются:

– Talend Open Studio;
– Pentaho Kettle;
– CloverETL;
– Jaspersoft ETL.

Рассмотрим каждый из них в отдельности.

Talend Open Studio

Программный продукт Talend Open Studio разработан для поддержки операций ETL, синхронизации и репликации баз данных, обеспечения качества данных для нужд анализа. В качестве описания процессов используется репозиторий метаданных [4].

Данный продукт распространяется в соответствии с лицензией GNU General Public License, что не позволяет использовать его совместно с продуктами, распространяющимися в соответствии с другими типами лицензий.

Последний релиз данной программы был выпущен 7 мая 2012 года, что позволяет сделать вывод, что программный продукт развивается. Данный продукт частично поддерживает параллельное выполнение заданий. Работа осуществляется в специальной программной среде (рисунок 2).

Основные поддерживаемые типы источников данных:

– MySQL, PostgreSQL, SQLite, MS SQL, Sybase, Oracle и другие;
– CSV и другие плоские файлы;
– Excel таблицы, XML файлы.

Основными недостатками данного продукта являются ограниченная функциональность; невозможность использования совместно с программными продуктами, распространяемыми в соответствии с лицензиями, отличными от GNU GPL; ограниченный набор поддерживаемых источников данных, сложность построения алгоритма сбора данных.

Pentaho Kettle

Программа Kettle (Pentaho Data Integration) – разрабатывалась как средство обеспечения интеграции Pentaho с исходными данными из прочих систем, но вскоре стала самостоятельным продуктом, поддерживающим основные ETL операции [5]. Работа осуществляется в специальной программной среде (рисунок 3).

Данный продукт распространяется в соответствии с лицензией GNU General Public License, что не позволяет использовать его совместно с продуктами, распространяющимися в соответствии с другими типами лицензий. Последний релиз данной программы был выпущен в сентябре 2011 года, что говорит о низкой скорости развития программы.

Основные поддерживаемые типы источников данных:

– MySQL, PostgreSQL, SQLite, MS SQL, Sybase, Oracle и другие;
– CSV и другие плоские файлы;
– Excel таблицы, XML-файлы.

Среди преимуществ данного программного обеспечения можно выделить полноценную поддержку параллельного выполнения и условную бесплатность. Основными недостатками данного продукта являются ограниченная функциональность; сложность расширения функционала; невозможность использовать совместно с программными продуктами, распространяемыми в соответствии с лицензиями, отличными от GNU GPL; ограниченный набор поддерживаемых источников данных.

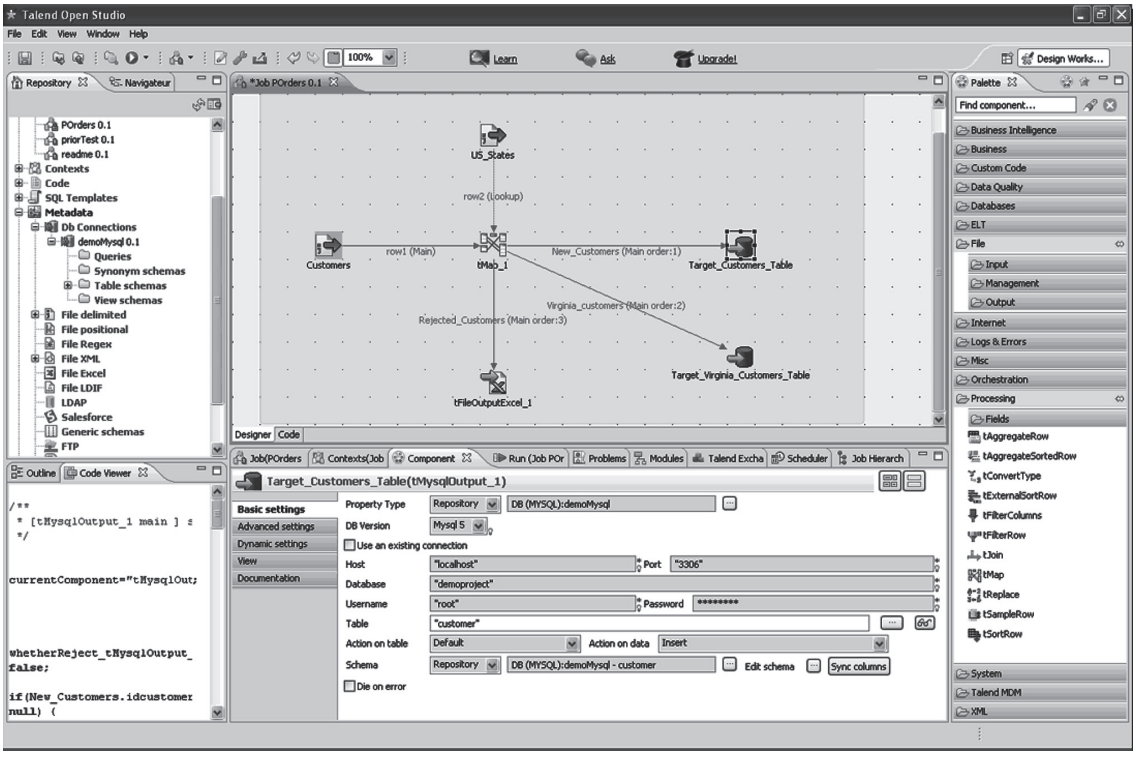
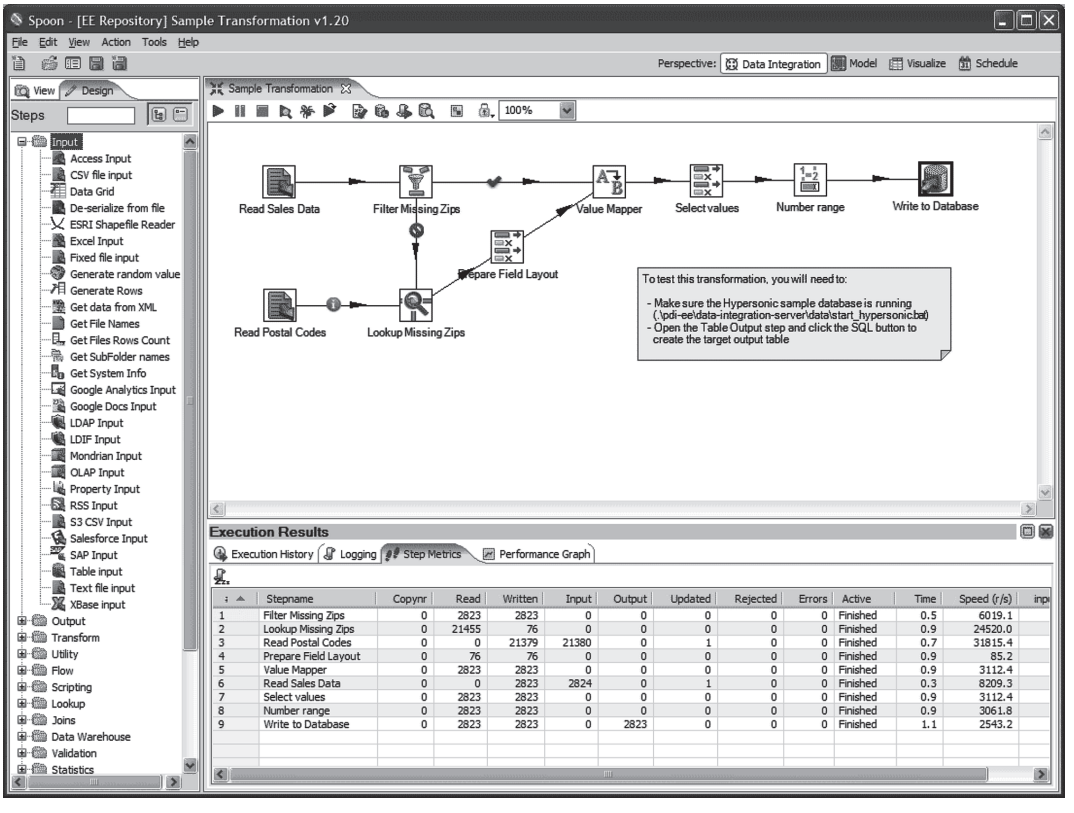


Рисунок 2 – Пример рабочего окна окружения продукта Talend Open Studio



	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Filter Missing Zips	0	2823	2823	0	0	0	0	0	Finished	0.5	6019.1
2	Lookup Missing Zips	0	21455	76	0	0	0	0	0	Finished	0.9	24520.0
3	Read Postal Codes	0	0	21379	21380	0	1	0	0	Finished	0.7	31815.4
4	Prepare Field Layout	0	76	76	0	0	0	0	0	Finished	0.9	85.2
5	Value Mapper	0	2823	2823	0	0	0	0	0	Finished	0.9	3112.4
6	Read Sales Data	0	0	2823	2824	0	1	0	0	Finished	0.3	8209.3
7	Select values	0	2823	2823	0	0	0	0	0	Finished	0.9	3112.4
8	Number range	0	2823	2823	0	0	0	0	0	Finished	0.9	3061.8
9	Write to Database	0	2823	2823	0	2823	0	0	0	Finished	1.1	2543.2

Рисунок 3 – Пример рабочего окна окружения продукта Pentaho Data Integration

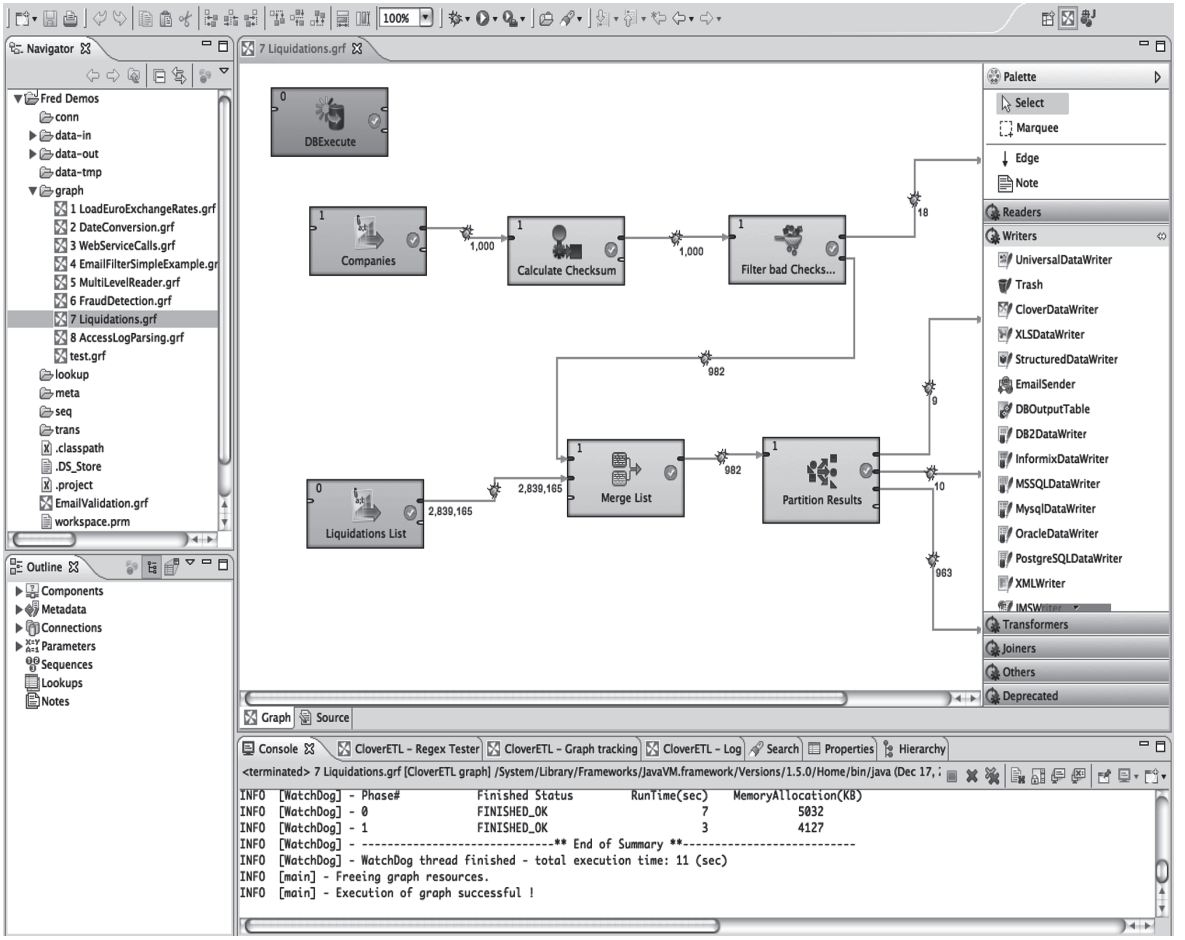


Рисунок 4 – Пример рабочего окна окружения продукта CloverETL

CloverETL

Программа CloverETL может использоваться в качестве отдельного приложения для выполнения операций ETL, а также может быть интегрирована с другими приложениями (в качестве библиотеки Java) [6].

Данный продукт распространяется в соответствии с лицензией GNU Lesser General Public License, что позволяет использовать его совместно с продуктами, распространяющимися в соответствии с другими типами лицензий. Последний релиз данной программы был выпущен в сентябре 2011 года, что говорит о низкой скорости развития программы. Работа осуществляется в специальной программной среде (рисунок 4).

Поддерживаемые типы источников данных:

– MySQL, PostgreSQL, SQLite, MS SQL, Sybase, Oracle, and Derby базы данных;

– CSV и другие плоские файлы;

– Excel таблицы;

– XML-файлы.

Основными преимуществами данной программы являются широкая поддержка мультизадачности и большое число поддерживаемых источников данных. Среди недостатков данного продукта стоит выделить ограниченность предоставляемого функционала и невозможность его расширения. Так же стоит отметить слабо развитое сообщество разработчиков.

Jaspersoft ETL

Данный продукт распространяется в соответствии с лицензией GNU General Public License, что не позволяет использовать его совместно с продуктами, распространяющимися в соответствии с другими типами лицензий [7]. Основные поддерживаемые типы источников данных: MySQL, PostgreSQL, Sybase, Oracle, CSV, Excel, XML-файлы. Работа осуществляется в специальной программной среде (рисунок 5).

Программный пакет Jaspersoft ETL отличается широкой поддержкой различных CRM систем, удобным дизайном процессов и диаграмм ETL. Основным недостатком является отсутствие доступной документации для open source версии.

Основываясь на данном обзоре, можно сделать вывод, что ни один из доступных программных комплексов не удовлетворяет в полной мере существующим требованиям. Представленные системы сложны в освоении, имеют строго ограниченные возможности, которые сложно или вовсе невозможно расширить. Также данные системы не поддерживают работу с нереляционными базами данных. Проектировать системы адаптивного сбора данных на основе таких систем практически невозможно.

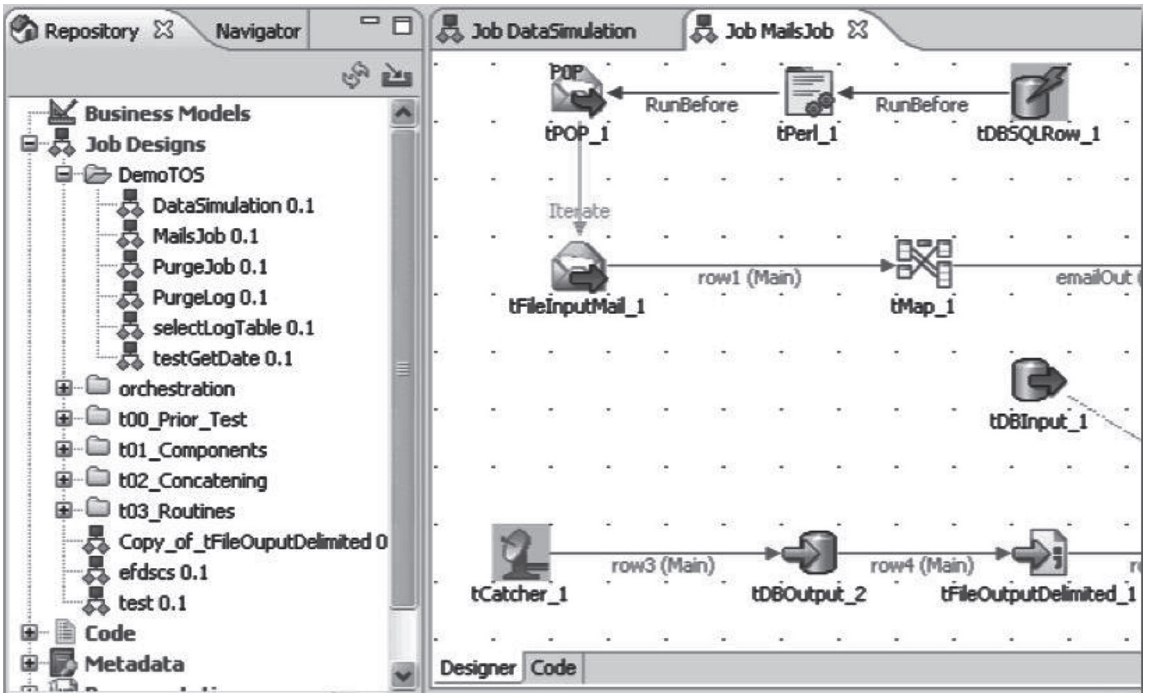


Рисунок 5 – Пример рабочего окна окружения продукта Jaspersoft ETL

Анализ существующих способов получения данных микроблогов Twitter

Существует несколько возможных подходов к сбору данных микроблогов социальной сети Twitter. В данном разделе будут рассмотрены некоторые из них.

Использование существующих хранилищ данных

В настоящее время в Интернете существуют несколько хранилищ данных, содержащих данные микроблогов Twitter. Некоторые из этих источников доступны для свободного скачивания и могут выступать в качестве первичных источников данных для системы сбора и хранения данных социальных ресурсов. Достоинством такого рода источников данных для разработчика является то, что самый долгий и трудоемкий этап сбора данных в случае использования таких хранилищ можно упростить или пропустить. Но, к сожалению, динамичность социальной сети Twitter не позволяет разработчикам таких хранилищ поддерживать актуальность хранимых данных. Таким образом, существующие хранилища данных превосходно подходят для задач обобщенного анализа социальной сети в целом и плохо годятся для подробного и актуального анализа какой-либо ее части в отдельности.

Сбор данных посредством поискового робота

Существует возможность сбора данных посредством специально спроектированного поискового робота. Данный поисковый робот может имитировать поведение обыкновенного пользователя. Таким образом, ему будет доступна вся информация пользователей социальной сети, находящаяся в открытом доступе. Главным преимуществом данного подхода является отсутствие каких-либо специфических ограничений на скорость сбора данных, накладываемых разработчиками социальной сети Twitter. Среди недостатков данного подхода можно выделить то, что сбор данных при помощи поискового робота будет наиболее трудоемким, а так же этот метод сбора данных запрещен в соответствии с лицензионным соглашением социальной сети Twitter.

Использование открытого программного интерфейса программирования

Социальная сеть микроблогов Twitter предоставляет разработчикам удобный и рекомендуемый способ доступа к накопленным данным посредством открытого программного интерфейса программирования (Twitter API) [8]. Twitter API состоит из следующих сервисов:

Search API – специализированный интерфейс прикладного программирования, предназначенный для поиска сообщений (статусов) пользователей по определенным ключевым выражениям, а так же поиска сообщений, ссылающихся на конкретного пользователя или событие.

Среди преимуществ данного интерфейса можно выделить удобный инструмент для поиска по данным, накопленный социальной сетью Twitter. Недостатками такого интерфейса является то, что:

– данные доступны за короткий промежуток времени (6–9 дней);

– присутствует ограничение на сложность поисковых запросов;

– нет возможности поиска по защищенным данным;

– возможно исключение некоторых сообщений / пользователей из результатов поиска вследствие оптимизации последнего;

– существует ограничение запросов по их частоте.

Rest API – интерфейс прикладного программирования, предназначенный для доступа к данным совершенно различного рода, таких как:

– базовая информация о пользователе;

– информация о подписчиках пользователя;

– информация о друзьях пользователя;

– список статусов пользователей;

– специализированные списки пользователей;

– мультимедийная составляющая сообщений пользователей.

Преимуществами данного интерфейса являются:

– обширный набор доступных методов, покрывающий совершенно различные данные системы;

– для доступа к информации стороннего пользователя нет необходимости авторизовать приложения у этого пользователя;

– доступны данные за весь период без ограничения по сроку давности.

Среди недостатков данного интерфейса можно выделить:

– ограниченный доступ к защищенным данным;

– существование ограничения запросов к интерфейсу по их количеству в час (150 запросов в час).

Streaming API – набор специализированных интерфейсов прикладного программирования, позволяющих получать данные практически в режиме реального времени. На данный момент этот интерфейс состоит из интерфейсов User Streams и Site Streams.

User Streams – интерфейс прикладного программирования, предназначенный для получения всей информации, которая, так или иначе, касается определенного пользователя. Преимущества данного интерфейса: позволяет получать данные практически в момент их появления в самой сети Twitter; ограничения по частоте запросов практически отсутствуют. Недостатки данного интерфейса: приложение, использующее данный интерфейс должно быть авторизировано пользователем, чьи данные оно пытается получить; есть ограничения, накладываемые на количество пользователей, чьи данные могут получаться программой с одного сервера.

Site Streams – интерфейс прикладного программирования, предназначенный для получения обновлений о статусах сразу группы пользователей. Преимущества данного интерфейса включают в себя преимущества, присущие интерфейсу User Streams, а так же возможность работы одновременно с большой группой пользователей. Среди недостатков данного интерфейса стоит выделить то, что приложение, использующее данный интерфейс, должно быть авторизировано пользователем, чьи данные оно пытается получить; интерфейс находится в стадии Beta тестирования.

Алгоритм сбора данных

Цепочка операций процесса сбора данных на основании информации учетной записи Twitter предполагает выполнение следующих шагов [9]:

1. Формирование запросов к системе микроблогов, согласно импорт-параметрам удаленных функций открытого интерфейса прикладного программирования (Twitter APIs).

2. Получение необработанных данных удаленных функций и их запись на первом уровне хранилища данных.

3. Постобработка и выгрузка данных из первого уровня во второй уровень хранилища.

4. Формирование витрин данных определенной структуры, которые содержат информацию, позволяющую провести анализ данных определенной учетной записи Twitter на основе данных о статусах, подписках и подписчиках с заданным уровнем детальности сбора данных.

В данной работе в качестве основного канала получения данных использовался HTTP REST стиль архитектуры программного обеспечения обмена данными и Интернет-сервисы открытого интерфейса прикладного программирования Twitter APIs. Для получения базовой информации о пользователях, связях между пользователями их статусами использовались следующие методы:

– GET users/show. Данный метод позволяет получить информацию о пользователе по его уникальному идентификатору или имени, а именно: имя пользователя, дата регистрации в сети Twitter, изображение пользователя (аватар), страна проживания, персональный сайт пользователя, часовой пояс (временная зона), количество подписчиков, количество друзей, используемый язык, описание пользователя, количество статусов (сообщений) пользователя, текущий статус пользователя, параметры оформления его домашней страницы (например, цвета различных элементов оформления).

– GET statuses/user_timeline. Данный метод позволяет получить список статусов (сообщений) пользователя, отсортированный по дате их публикации. Каждый статус имеет следующие атрибуты: текст сообщения, место, откуда статус был создан (если доступно), дата создания, используемые ссылки на внешние ресурсы, используемые hashtag'и (ссылки на события или темы), список пользователей, которые упоминались в сообщении, если сообщение является ответом на другое сообщение, то можно узнать уникальный идентификатор исходного сообщения, количество «перепостов» (retweet'ов).

– GET users/lookup. Данный метод позволяет получать информацию о группе пользователей по их уникальным идентификаторам.

– GET followers/ids. Метод позволяет получить список пользователей, которые подписаны на обновления конкретного пользователя.

– GET friends/ids. Метод позволяет получить список друзей конкретного пользователя.

– GET favorites. Метод позволяет получить список статусов, которые были отмечены как избранные конкретным пользователем.

Общий алгоритм сбора данных можно представить следующим образом (рисунок 6).

Пользователи регистрируют заявки на анализ данных. Заявка включает в себя имя пользователя социальной сети Twitter и некоторое количество ограничивающих параметров, таких как, например, максимальная глубина поиска. Далее система, используя прикладной программный интерфейс, начинает анализировать ближайшее окружение рассматриваемого пользователя, затем окружение его окружения и т.д. до тех пор, пока поиск будет удовлетворять ограничивающие параметры. В случае выявления пересечения с данными, собранными по другим запросам пользователей, поиск в данной ветви автоматически останавливается, что позволяет сильно сократить количество обращений к прикладному программному интерфейсу.

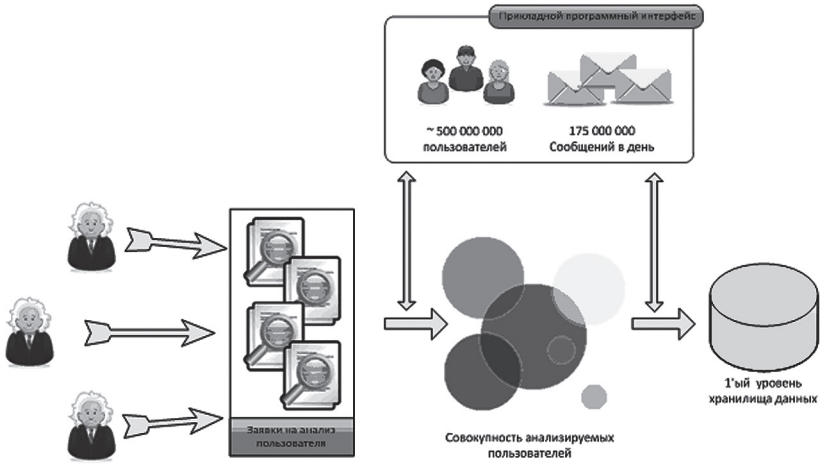


Рисунок 6 – Алгоритм сбора данных в соответствии с заявками пользователей

Параллельно с этим процессом в автоматическом режиме происходит сбор подробных данных о пользователях, которые были найдены на предыдущем этапе, их сообщениях и т.д. Данный процесс так же оптимизирован, а именно, благодаря итерационности алгоритма в течение каждой итерации из социальной сети Twitter собираются только те данные, которые были изменены по сравнению с прошлой итерацией. Таким образом, после прохождения N итераций на первом уровне хранилища данных система будет хранить наиболее актуальную информацию, покрывающую все зарегистрированные пользователями запросы [10].

Литература:

1. Charles Kadushin, Understanding Social Networks: Theories, Concepts, and Findings / Charles Kadushin. – Oxford University Press, 2012. – 264 p.

2. Ian H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, Third Edition / Ian H. Witten, Eibe Frank, Mark A. Hall. – Morgan Kaufmann, 2011. – 664 p.

3. Холод, И.И. Методы и модели анализа данных: OLAP и Data Mining / И.И. Холод, М.С. Куприянов, В.В. Степаненко. – Санкт-Петербург: БХВ-Петербург, 2004. – 336 с.

4. Talend, Talend Open Studio User's Guide, 2007. – 570 p.

5. Roldan, M. Pentaho 3.2 Data Integration: Beginner's Guide, 2010. – Packt publishing. – 492 p.

6. CloverETL, CloverETL Designer User's Guide / [Electronic source] / Access mode: http://www.cloveretl.com/documentation/UserGuide. – Date 04.07.2011.

7. JasperSoft, JasperETL User's Guide version 2.1.1, JasperSoft, 2007. – Talend Inc. – 348 p.

8. Twitter Developers Documentation / [Электронный ресурс] / Access mode: https://dev.twitter.com/docs. – Date: 05.21.2012.

9. Ахрамович, А.В. Модель аналитической системы мониторинга операционной деятельности в учреждениях государственного сектора с использованием OLAP-технологии / А.В. Ахрамович, С.В. Валенда, И.Л. Чваркова, С.Г. Тихоненко // Журнал «Электроника Инфо». – Минск, 2011. – №3. – С. 57–61.

10. Информационные технологии и системы 2011 (ИТС 2011): материалы научной конференции, БГУИР, Минск, Беларусь, 26 октября 2011 / редкол.: Л.Ю. Шилин [и др.]. – Минск: БГУИР, 2011. – 306 с. – ISBN 978-985-488-816-3 – С. 104–105.

НОВОСТИ

ДВУХКАНАЛЬНЫЙ ОПЕРАЦИОННЫЙ УСИЛИТЕЛЬ

Прецизионный, малопотребляющий, двухканальный операционный усилитель ADA4500-2 обладает максимальным напряжением смещения постоянной составляющей 120 мкВ, шириной полосы 10 МГц, шумом 14,5 нВ/√Гц, а также rail-to-rail диапазонами входных и выходных напряжений (диапазон напряжения включает в себя напряжения питания). Инновационная топология схемы с нулевыми искажениями при переходе через ноль обеспечивает высокую линейность во всем диапазоне входных напряжений, подавление пульсаций напряжения питания не менее 98 дБ и ослабление синфазного сигнала не менее 95 дБ. Эта комбинация характеристик делает данный усилитель идеальным вариантом для аналогового преобразования сигналов прецизионных датчиков, входного интерфейса с аналого-цифровыми преобразователями и портативных измерительных приборов. ADA4500-2 работает от одного напряжения питания в диапазоне от 2,7 В до 5,5 В, потребляя максимальный ток 1,8 мА во всем диапазоне рабочих температур. Компонент выпускается в 8-выводных корпусах LFCSP и MSOP, рабочий температурный диапазон составляет от –40°C до +125°C.

ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ: 10 ПРОГНОЗОВ НА 2013 ГОД

В 2012 году не было никаких сомнений, что благодаря облачным вычислениям в распоряжении организаций появился новый и более совершенный способ управления ИТ-ресурсами. Тысячи компаний и организаций подготовились к переносу своих ИТ-ресурсов в облако.

Преимущества облака для любых предприятий, от огромных корпораций до госучреждений и малых вебсайтов очевидны, и поставщики спешат удовлетворить потребности своих клиентов.

В этом году все больше изготовителей и поставщиков решений соперничают, предлагая потребителям простые в использовании службы с повышенной надежностью, низкой ценой и более совершенными функциями управления. В 2013 году по прогнозам облако должно получить широкое распространение, должны произойти новые революционные применения, которые было трудно себе представить еще совсем недавно.

В данном обзоре были собраны прогнозы на 2013 год, сбылась ли хоть часть из них?

Amazon меняется

После неудач, вызванных сбоями в обслуживании клиентов, в том числе в результате известного рождественского отключения, из-за которого пострадал сайт Netflix, назначен новый директор по надежности Amazon Web Services. Компания также выделит десятки, если не сотни миллионов долларов на обновление облачной инфраструктуры, в частности переоборудование центра в Виргинии, где в 2012 году произошли три отключения.

Каждому – частное облако

Компании готовы к переходу в облако, но на своих условиях. Как следствие, они получают преимущества от огромного разнообразия внедряемых облачных вариантов. Общедоступные облака, предоставляемые такими ведущими поставщиками, как Amazon Web Services, сохранят популярность, наряду со смешанными частными и общедоступными облаками.

Но частная модель, в которой основные ресурсы хранятся и управляются локально, в 2013 году будет расти быстрее других, учитывая стремление поставщиков удовлетворить требования рынка.

OpenStack и CloudStack объединяют усилия

OpenStack Foundation заключит кооперативное соглашение с открытым облачным конкурентом CloudStack, обе организации будут теснее сотрудничать с Amazon Web Services. В апреле поставщик облачных служб Citrix покинул OpenStack и перевел собственную программу облачного взаимодействия (CloudStack) в Apache Software Foundation. OpenStack, среди учредителей которой Rackspace и NASA, объявила себя независимой организацией OpenStack Foundation.

Но при огромном спросе на облачные службы и множестве посредников, стремящихся удовлетворить этот спрос, безупречное взаимодействие – обязательное требование, и поставщики облачных услуг потребуют устранения любых препятствий на пути облачного строительства.

В дело вступает государство

В 2012 году федеральное правительство, администрации штатов и местные власти начали движение к облаку, осознав потенциальную экономию. Эта тенденция будет усиливаться в 2013 году по мере того, как учреждения начнут пользоваться выгодами облачного размещения. В ответ системные интеграторы и другие поставщики решений на рынке госучреждений примутся перевооружать свой бизнес для предоставления облачных служб.

Hewlett-Packard: резкий разворот к облаку

Руководство Hewlett-Packard в сентябре 2012 года назвало стратегию конвергированного облака своим высшим приоритетом. Главный управляющий HP Мег Уитман пытается вдохнуть жизнь в технического гиганта после серии неудач 2012 года, и рассчитывает на рост облачного сегмента компании.

В сентябре HP организовала внутреннее подразделение, объединив разрозненные группы, занимавшиеся облачными продуктами и услугами, которое, как говорится в служебной записке, «будет работать с различными облачными инициативами HP, совершенствуя объединенный комплекс конвергентного облака, архитектуру, продукты/решения и инфраструктуру доставки, одновременно формируя центральную группу продуктов/разработки и технологии, чтобы ускорить выпуск облачных решений HP на рынок».

Новые подходы к хранению данных

В уходящие месяцы 2012 года ведущие поставщики услуг хранения данных в облаке Amazon Web Service, Google и Dropbox были вовлечены в ценовую войну в попытке увеличить свою долю рынка. Сверхнизкие цены могут означать только более широкое распространение услуг хранения информации в облаке и изменение способов использования и продажи систем хранения.

Малый и средний бизнес движется в облако

Представители малого и среднего бизнеса называют безопасность, сложность и цены главными причинами, обусловившими недоверие к облачным службам. Но в 2013 году их привлекут новые облачные услуги, предоставляемые HP, Microsoft, Oracle, IBM и поставщиками облачных решений.

Облако приведет к международному конфликту

Уже в течение нескольких лет китайские хакеры атакуют Интернет-компании, в частности, Google. Но по мере того, как поставщики облака начинают предлагать свои услуги во всем мире, будущие атаки почти неизбежно нанесут урон большему числу американских компаний и вынудят официальных лиц страны дать публичный ответ. В результате глобальный охват облака станет международной проблемой.

От IaaS к PaaS

В 2012 году поставщики инфраструктуры как услуги (IaaS) занимали ведущие позиции в развитии облачных вычислений. Amazon Web Services, Rackspace, Terremark и другие предоставляют клиентам доступ к хостингу и вычислениям.

В 2013 году поставщики платформы как услуги (PaaS), такие как RightScale, Engine Yard и Heroku, приобретут большое значение, так как компании захотят получить более совершенные инструменты для миграции все более сложных рабочих ИТ-нагрузок в облако.

Поставщики решений становятся облачными брокерами

2013-й станет годом, когда поставщики решений воспользуются предоставляющейся благодаря облаку возможностью стать брокерами по перемещению бизнеса в размещенную ИТ-среду. Это подтверждается исследованием, проведенным компанией IDC по заказу Microsoft и выпущенным в конце декабря 2012 года. Согласно выводам исследования, спрос на «готовых к работе в облаке» ИТ-специалистов будет расти на 26 % в год вплоть до 2015 года, и в мире откроется до 7 млн рабочих вакансий, связанных с облаком.

Джек Маккарти, CRN/США

НОВОСТИ

ОБЛАЧНЫЕ ТЕХНОЛОГИИ – ШВЕЙЦАРСКИЙ ВАРИАНТ

Intel и «Мирантис» намерены вместе развивать экосистему OpenStack, призванную объединить всех участников облачной индустрии.

Облачный джинн выпущен из бутылки – к 2015 году у облачных инфраструктур будет 3 млрд пользователей, к ним будет подключено 15 млрд устройств, а объемы данных, циркулирующих в облаках, будут удваиваться каждые два года. Все это стало причиной замешательства ведущих ИТ-компаний – одни бросились создавать свои собственные облака на базе проприетарных технологий, другие заявили, что ничего не происходит и они готовы решить любые проблемы облаков, а третьи сделали акцент на Open Source. По мнению Камиля Исаева, генерального директора по исследованиям и разработкам Intel в России, проприетарные технологии невозможно адаптировать к многообразию изменений. Облака несут массу вызовов современной ИТ-индустрии: сокращение времени реакции на происходящие изменения; требование теоретически бесконечного масштабирования как по производительности, так и по объемам хранения; оперативное устранение узких мест; обеспечение надежности и безопасности конфигураций на базе проактивной аналитики.

Если до 2015 года, по мнению Исаева, еще будут сохраняться частные и публичные облака, то затем индустрия войдет в эпоху «открытых облаков», характеризуемую интероперабельностью, опорой на открытые стандарты, мультивендорные решения и промышленные стандарты. Среди своих задач на пути движения к этой эпохе Intel называет прежде всего мониторинг и выявление потребностей пользователей, создание безопасных, надежных и унифицированных облачных платформ, а также активное участие в экосистеме, объединяющей всех участников современного облачного движения. Компания «Мирантис» получила от фондов Intel Capital, WestSummit Capital из Китая и Dell Ventures 10 млн долл. на удвоение численности своих инженерных подразделений на территории СНГ, занятых в развитии технологий платформы OpenStack. Эти инвестиции, по словам Исаева, как раз и являются частью программы движения корпорации к «открытым облакам». Пока в фокусе совместной работы три направления: управление энергопотреблением ЦОД с помощью пакета Intel Data Center Manager, оптимизация сетей на базе концепции программно-конфигурируемых сетей (Software-Defined Networking, SDN), использование платформы Apache Hadoop для развертывания облаков на базе технологий OpenStack и продуктов Intel.

Инициированный хостинг-провайдером RackSpace и НАСА, проект OpenStack по созданию облачных решений вырос в глобальное сообщество программистов, занятых созданием стандартизированной облачной ОС с открытым кодом. «Главная идея OpenStack – изначально создавать на базе открытого ПО все специально для облачной среды», – отметил Александр Фридланд, председатель совета директоров «Мирантиса».

Функция сообщества OpenStack, объединяющего на сегодняшний день около 6 тыс. членов, среди которых AT&T, RackSpace, HP, IBM, Red Hat, Intel, Dell, – выступать в качестве зонтичного бренда и управлять проектами разработки программного обеспечения с открытым кодом, распространяемого под лицензией Apache License 2.0. Иначе говоря, брать на рынке востребованные решения, упаковывать их в оболочку на базе открытых стандартов и включать в облачную платформу. Все это делает OpenStack похожим на фонд Apache, тем более что совместно с Dell и Intel компания «Мирантис» основала фонд OpenStack Foundation, задача которого – определять направления развития платформы построения облаков и содействовать распространению и росту экосистемы OpenStack.

Отсутствие совместимости между облачными системами разных разработчиков – одно из главных препятствий для широкого использования облаков, особенно в корпоративной среде, а с помощью OpenStack, как ожидается, любая организация может развернуть собственное облако. Однако обязательное условие этого – независимость фонда поддержки такой платформы от какого-либо производителя. По словам Фридланда, «Мирантис» и OpenStack Foundation являются сегодня своего рода Швейцарией в мире ИТ; взаимодействуя со всеми, они остаются при этом независимыми.

osp.ru

ОФИС ОНЛАЙН.
ПИШЕМ В ОБЛАКАХ

Офис онлайн или Офис на локальном компьютере. Что удобнее, в каких случаях и насколько это безопасно? Офис онлайн – это очередной шаг в облачные технологии. Заранее оговорюсь, что под офисом я имею ввиду определенный список приложений, который входит в пакет офиса, такие как средство написания текстов, средство создания электронных таблиц, средство создания презентаций. Такую двойственность слова Офис привнесла нам компания Microsoft, выпустив продукт Microsoft Office, в который входили такие приложения, как Word, Excel, PowerPoint. Так что под термином офис онлайн надо понимать возможность создавать и редактировать документы онлайн, не устанавливая никаких приложений на свой личный ПК.

Что такое офис онлайн?

Офис онлайн – это облачные сервисы, которые дают возможность создавать и редактировать документы онлайн, не устанавливая никаких приложений на свой личный ПК. Так же, Вы можете использовать хранение информации в Интернете, если у Вас офис в облаке. История этого SAAS сервиса уходит в начало 2010 года. Google стал первопроходцем, создав офис онлайн. Он дал возможность всем попробовать, удобно ли использовать офис в облаке. Этот первый сервис получил название Google Docs. Он давал возможность создавать и редактировать тексты и электронные таблицы. Так же была возможность их хранить в своем Google аккаунте. Позже, Google переименовал свой сервис в Google Drive, предоставив пользователям возможность не только создавать и редактировать электронные документы, но и организовать их хранение и синхронизацию на разных локальных компьютерах. Фактически, Google совместил два облачных сервиса – Хранение информации в Интернете и офис онлайн. Ну, а дальше сервисы онлайн редактирования документов начали создаваться другими крупными компаниями. Компания Microsoft выпустила продукт Office Web Apps и Office 365. Они перенесли функционал продукта Microsoft Office в SAAS сервисы. Еще в конкуренцию вступили такие производители, как ZOHO, ThinkFree и многие другие.

Рссмотрим плюсы и минусы онлайн офиса

Причины использовать или не использовать офис онлайн часто совпадают с аналогичными причинами применимо к облачным технологиям и онлайн сервисам. Удобно, что доступ к редактору документов есть ото всюду, где есть Интернет. Удобно что все документы хранятся в одном месте. В отсутствие Интернета можно пользоваться локальными версиями редакторов. Вопрос безопасности тут встает как и в других схожих тематиках, таких как бухгалтерия онлайн. Если Вам нечего скрывать, нечего и боятся. Если есть что скрывать, не храните и не редактируйте эти файлы в Интернете. Для юридических лиц переход на использование офиса онлайн более сложен. Он сопряжен с сложным выбором между платными и бесплатными продуктами. Еще могут возникнуть сложности с печатью.

digiclouds.ru

УПРАВЛЕНИЕ ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ И ОБЕСПЕЧЕНИЕ ОТКАЗОУСТОЙЧИВОСТИ IAAS-ОБЛАКА

УДК 004.75

Ю.И. Воротницкий, В.П. Кочин, БГУ, г. Минск
В.А. Волчок, А.И. Бражук, ГрГУ им. Я. Купалы, г. Гродно

Аннотация

Рассмотрены особенности управления программным обеспечением (ПО) и представлена схема формирования виртуальной машины в облаке IaaS (Infrastructure as a Service, англ. «инфраструктура как услуга»). Исследованы вопросы архитектурной организации управляющего ПО IaaS (OpenStack) и предложена обобщенная структура отказоустойчивого облака на основе универсальных узлов.

Введение

Одной из устойчивых мировых тенденций развития средств информатизации является миграция к так называемым «облачным» технологиям (cloud computing). Эти технологии основаны, как правило, на централизованном хранении и обработке информации в центрах обработки данных (ЦОД), на гибких механизмах управления ресурсами и выделения их удаленным пользователям. Внедрение облачных концепций меняет приоритеты в сфере информационных технологий (ИТ). Первостепенное значение приобретают информационные ресурсы, на разработку которых направляются основные усилия. Компьютерные устройства как таковые уходят на второй план. Независимо от типа, марки, производителя и местонахождения любое из них лишь должно обеспечивать доступ к сетевым сервисам. При этом облачный подход позволяет создать доступную информационную среду и обеспечить синхронизацию деятельности пользователей, осуществляемой с разных устройств (рабочая станция, домашний компьютер, личный планшет, смартфон и т.п.).

Использование облачных технологий на корпоративном и отраслевом уровнях нацелено, во-первых, на обеспечение потребителей современной ИТ-инфраструктурой, программными средствами, электронными ресурсами и сервисами; во-вторых, на оптимизацию затрат за счет использования информационных и вычислительных ресурсов «облака», гибко предоставляемых пользователям в соответствии с их потребностями [1]. В частности, в системе образования это позволит обеспечить мобильность и актуальность образовательных ресурсов. Учебные заведения получат возможность без дополнительных затрат на сопровождение локальных инфраструктур использовать современные и постоянно актуализируемые программные средства и сервисы, предоставляемые ЦОД. Облачные технологии также позволят вовлечь в образовательный процесс личные компьютерные устройства преподавателей, учащихся и их родителей.

В данной работе рассматриваются особенности управления ПО и элементы архитектуры отказоустойчивого облака, реализующего подход «инфраструктура как услуга» (Infrastructure as a Service – IaaS). Концепция IaaS предполагает обеспечение потребителей доступом к вычислительным ресурсам на уровне виртуальных машин (ВМ) с возможностью выбирать, устанавливать и настраивать ПО, как системное, так и прикладное [2]. Уровень «инфраструктура как услуга» является системообразующим при построении иерархических универсальных облачных структур и может служить платформой для развертывания сервисов облака более высокого уровня, таких как «платформа как услуга» (Platform as a Service), «прикладное ПО как услуга» (Software as a Service) и т. п.

Универсальность модели IaaS позволяет использовать ее не только в крупных ЦОД для вычислительных кластеров и суперкомпьютеров, но и для оптимизации корпоративной серверной и сетевой инфраструктуры (частные облака), а также для объединения программно-технических ресурсов различных организаций (совместно используемые или общие облака).

Проблемы внедрения облачных технологий и обеспечения информационного взаимодействия учреждений образования исследуются в рамках Государственной программы научных исследований на 2011-2015 годы «Научные основы и инструментальные средства информационных и космических технологий» (ГПНИ «Информатика и космос») по теме «Разработка модели информационного взаимодействия учреждений образования в рамках национальной информационной инфраструктуры ГРИД», осуществляемой Белорусским государственным университетом (Центр информационных технологий БГУ) совместно с Гродненским государственным университетом им. Я. Купалы (лаборатория Системотехники). В частности, в лаборатории Системотехники ГрГУ им. Я. Купалы на основе ПО с открытым исходным кодом (Openstack, Linux Ubuntu Server) и бюджетного технического обеспечения (стандартные рабочие станции) развернут тестовый стенд, позволяющий моделировать и исследовать различные IaaS-архитектуры.

Управление программным обеспечением в облаке IaaS

Облачные методы управления ПО основаны на традиционных подходах системного администрирования, но при этом нацелены на использование преимуществ технологий виртуализации и широких возможностей масштабирования и эластичности, предоставляемых управляющим ПО IaaS [3, 4].

Обобщенный процесс создания виртуальной компьютерной системы (экземпляра ВМ) в облаке представлен на рисунке 1.

Экземпляр (instance) – это виртуальная машина, работающая под управлением гипервизора (менеджера ВМ) на физическом узле. Перед первым запуском экземпляра в вычислительной среде создается (копируется из образа дисков ВМ) его файловая система. Образы дисков (images) – это фактически шаблоны ВМ, содержащие необходимое для работы ПО. С использованием одного шаблона может быть сформировано множество экземпляров.

При создании экземпляра ВМ определяются перечень и количественные характеристики виртуальных техни-

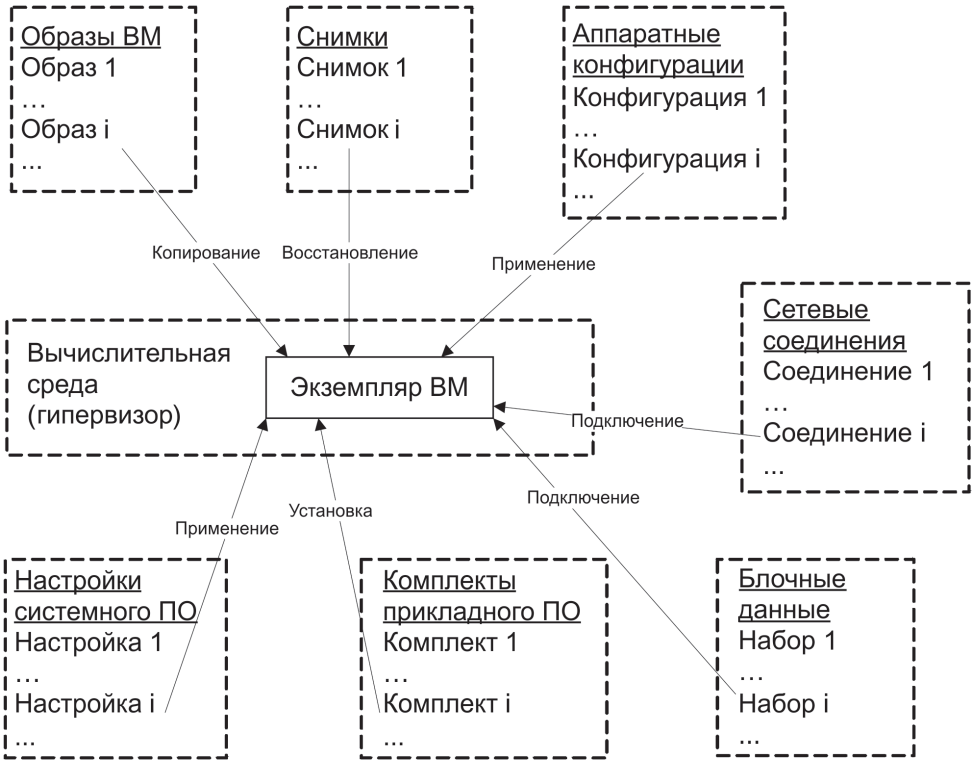


Рисунок 1 – Формирование экземпляра ВМ в облачной среде

ческих ресурсов – аппаратная конфигурация (в том числе и виртуальные сетевые адаптеры). Также на основе предопределенных сценариев (скриптов) может осуществляться первоначальная настройка системного ПО и установка необходимых комплектов прикладного ПО.

Как правило, файловая система экземпляра является локальной копией образа, поэтому ее изменения никак не влияют на него. Удаление экземпляра предполагает удаление его локальной копии. При этом, для сохранения данных ВМ могут быть использованы механизм копий состояний ВМ – снимков (snapshots) и (или) подключаемые постоянные (persistent) источники данных.

Облачный подход к управлению программным обеспечением предоставляет пользователям услуги IaaS ряд возможностей:

– автоматизацию развертывания виртуальных машин на основе библиотек (хранилищ) образов ВМ, виртуальных аппаратных и системных конфигураций и комплектов прикладного ПО, а также средств их создания, редактирования и упорядоченного хранения. Например, в настоящее время существуют и активно развиваются библиотеки образов ВМ с предустановленным прикладным ПО (в том числе и opensource), практически готовые к эксплуатации сразу после операции импорта в облако. Это существенно упрощает настройку ПО и снижает требования к квалификации пользователей услуги IaaS;

– настройку и манипуляцию конфигурациями ВМ. Например, механизм снимков предоставляет возможности сохранения состояния и быстрого восстановления работоспособности ВМ в случае программных ошибок и системных сбоев. Также путем создания шаблонов на основе снимков может быть организовано тиражирование системных конфигураций;

– использование механизмов и средств, обеспечивающих высокую доступность, отказоустойчивость, экспорт и импорт виртуальных машин (миграцию внутри облака и между облаками). Наличие этих функций во многом зависит от возможностей используемого гипервизора и степени интеграции управляющего ПО с ним;

– манипуляцию с сетевыми подключениями и создание произвольных (виртуальных) сетей в рамках концепции «Сеть как услуга» (Network as a Service).

Структура управляющего ПО IaaS-облака

Исходя из анализа существующего управляющего ПО IaaS, типовой состав облачной системы может включать следующие функциональные компоненты, как показано на рисунке 2:

– интерфейс пользователя. Позволяет получить доступ к функциям управления, предназначенным как для пользователей, так и для администраторов. Как правило, строится на основе протокола HTTP (HTTPS) на основе WEB-технологий (WEB-интерфейс) и (или) интерфейса командной строки (консоли);

– управление ВМ и ресурсами (вычислительная функция). Обеспечивает драйверы для взаимодействия с различными гипервизорами ВМ, планирование и оптимизацию доступных аппаратных ресурсов, а также мониторинг производительности, отказоустойчивость и высокую доступность облачных ресурсов;

– доступ к объектам и файлам. Позволяет работать с хранимыми данными. В существующих облачных системах

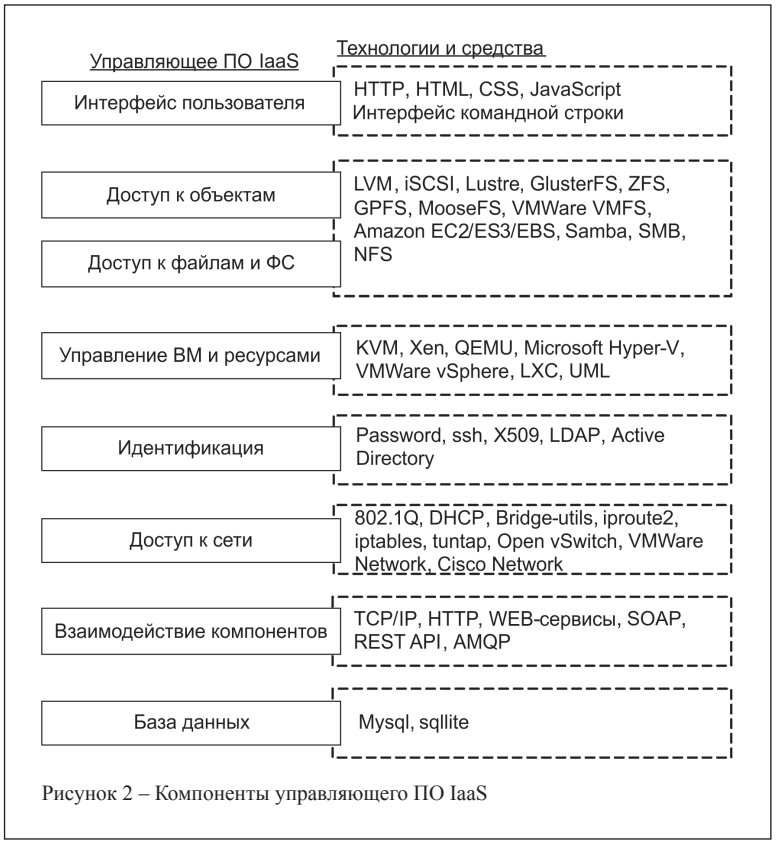


Рисунок 2 – Компоненты управляющего ПО IaaS

комбинируются различные подходы: на уровне объектов (образы виртуальных машин, пользовательские данные), на блочном уровне, с использованием распределенных и сетевых файловых систем;

– идентификация. Обеспечивает единый механизм и средства аутентификации и авторизации для пользователей, администраторов и сервисов облачной среды;

– доступ к сети. Позволяет пользователям создавать произвольные сети и подключать сетевые адаптеры ВМ к ним, обеспечивает взаимодействие ВМ и доступ к сети Интернет и т.п.;

– взаимодействие компонентов. Обеспечивает «внутренний» информационный обмен компонентов управляющего ПО;

– база данных. Реализует хранение внутренней информации облачной системы;

Как видно из рисунка 2 управляющее ПО IaaS может использовать широкий спектр существующих технологий и программных продуктов для реализации своих функций. К ключевым особенностям архитектуры, определяющим эффективность IaaS-системы, следует отнести:

– сервисный подход к внутренней организации. Модули системы организованы в виде сетевых сервисов (служб) и их функции доступны удаленно посредством соответствующих прикладных программных интерфейсов (application programming interface – API);

– комбинирование различных методов взаимодействия компонентов облачной платформы на основе протокола HTTP, технологий WEB-сервисов, специализированных интерфейсов (REST API), а также промежуточного ПО, ориентированного на обработку сообщений, в частности, на основе стандарта AMQP (Advanced Message Queuing Protocol);

– независимость сервисов от информации о состоянии. Данные о состоянии сервиса должны храниться отдельно от его экземпляра, что позволяет восстановить работоспособность при сбое, масштабировать службу и т.п.

Примером управляющего ПО, реализующего облачные возможности на достаточно хорошем уровне, является OpenStack [3], набор открытого программного обеспечения, предназначенный для установки и запуска облачной инфраструктуры вычислений и хранения информации, совместимый со стандартом Amazon EC2/S3 на уровне программных интерфейсов, хорошо документированный и поддерживаемый большинством IT-корпораций.

Обеспечение отказоустойчивости частного IaaS-облака

На практике меры по обеспечению отказоустойчивости (High Availability) компьютерной системы нацелены на минимизацию времени простоя и защиту данных от утраты и реализуются путем введения избыточных компонентов. На уровне управляющего ПО облачной платформы избыточность обеспечивают дополнительные (резервные) экземпляры сервисов, а также специализированные средства для переключения активного экземпляра и балансировки запросов и средства резервирования данных.

Существующие подходы к архитектуре IaaS на основе разделения узлов на функциональные группы (управляющие, вычислительные, хранения, оконечные и т.п.) [5] в отказоустойчивой конфигурации можно обобщить, введя понятие универсального узла облака. Таким образом, как показано на рисунке 2 архитектурными единицами облака OpenStack являются:

– универсальные узлы, способные совмещать вычислительные и сервисные функции, а также балансировку запросов API и HTTP;

– отказоустойчивый кластер Mysql/RabbitMQ, обеспечивающий хранение внутренней информации облака и информационный обмен между компонентами.

Исходя из задач облака и возможностей поддерживающей инфраструктуры, а также определив количество универсальных узлов, состав и количество запускаемых на них сервисов, можно обеспечить требуемый уровень отказоустойчивости IaaS-облака.

Вычислительные и сетевые сервисы (nova-compute, nova-scheduler, quantum-server и т.п.) содержат встроенную функциональность для объединения в отказоустойчивые пулы и могут обслуживаться простыми средствами (скриптами-сценариями).

Для мониторинга работоспособности, переключения экземпляров и балансировки запросов сервисов API (nova-api, glance-api, glance-registry, keystone, keystone-api, quantum-api и т.п.) используются технологии виртуальных IP-адресов (virtual IP address) или прокси транспортного (TCP) и прикладного (HTTP) уровней [6].

Официальное руководство по отказоустойчивости OpenStack [6] ссылается на ПО Pacemaker как универсальный кластерный стек для мониторинга, переключения сервисов и балансировки запросов для платформы Linux. В кластере Pacemaker коммуникации осуществляются посредством соответствующего уровня сообщений (Corosync), обеспечивающе-

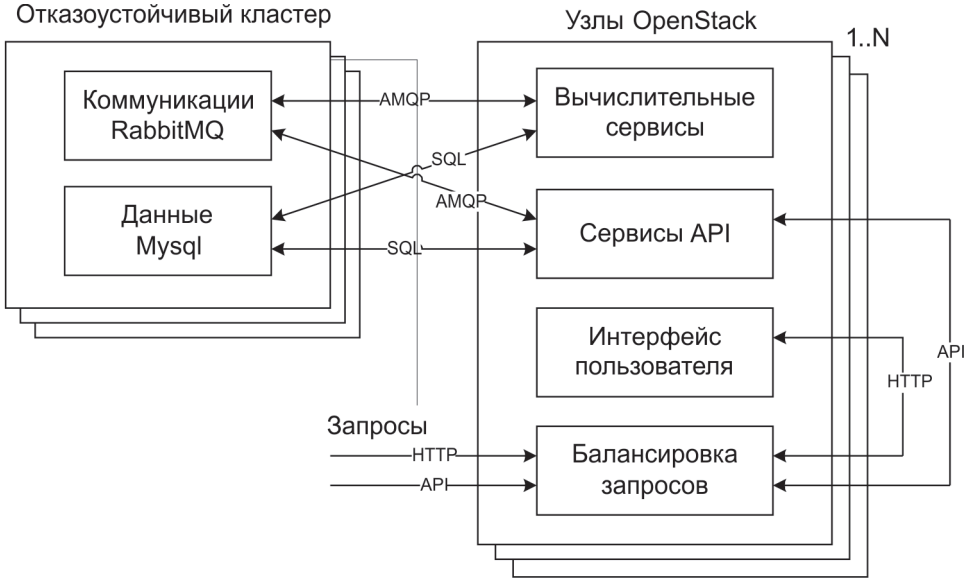


Рисунок 3 – Распределенная структура отказоустойчивого облака OpenStack

го кольцевой механизм доставки сообщений (Totem) на основе протокола UDP или технологии InfiniBand, кворум и членство в кластере. ПО Pacemaker взаимодействует с приложениями посредством ресурсных агентов (resource agent – RA).

В отказоустойчивых конфигурациях OpenStack может применяться бесплатный продукт HAProxy – балансировщик нагрузки транспортного (TCP) и прикладного (HTTP) уровней, а также подобные ему программные и технические решения.

Подсистемы внутренних коммуникаций и хранения данных являются сторонними продуктами к OpenStack, и для их отказоустойчивости существуют специализированные средства. В частности, для резервирования данных на бюджетном техническом обеспечении широко используется бесплатный продукт DRBD, реализующий по сути сетевой RAID1 («зеркалирование»). Простой отказоустойчивый кластер для Mysql и RabbitMQ на основе DRBD и Pacemaker можно собрать из двух компьютеров с дисками требуемого объема и дополнительными сетевыми картами. Использование специализированных программно-технических средств для хранения данных с функцией резервирования существенно упрощает задачу обеспечения отказоустойчивости. В этом случае можно использовать как Pacemaker, так и встроенные средства кластеризации RabbitMQ и сторонние продукты для Mysql (MySQL-MMM, Galera).

Заключение

Существующее управляющее ПО IaaS (в частности, OpenStack) позволяет масштабировать «размеры» облака в достаточно широких пределах, поэтому облачная инфраструктура системы образования может включать программно-технические средства республиканского отраслевого ЦОД [1], вычислительные ресурсы крупных образовательных учреждений (университетов, институтов), а также существующую вычислительную ГРИД-инфраструктуру.

В рамках реализации образовательной облачной среды на уровне IaaS следует выделить ряд актуальных технических задач, в частности формирования библиотек (репозиториев) образов виртуальных машин с готовым ПО учебного, отраслевого и научного назначение, например, позволяющего организовать распределенные вычисления.

Учитывая «молодость» технологий и ПО, недостаток документации и закрытость разработок многих производителей интерес представляют методики, руководства, рекомендации по развертыванию облаков на основе бесплатного ПО. В частности, унификация программной реализации узлов IaaS и инструментальных средств для их установки и настройки позволяет существенно упростить управление частными и общими облаками.

Литература:

1. Абламейко, С.В. Перспективы применения «облачных» технологий в системе образования Республики Беларусь / С.В. Абламейко, Ю.И. Воротницкий, Н.И. Листопад // Четвертая Международная научная конференция «Суперкомпьютерные системы и их применение» (SSA'2012), 23-25 октября 2012 года, Минск: доклады. – Минск: ОИПИ НАН Беларуси, 2012. – С. 29–36.

2. Листопад, Н.И. Модели функционирования «облачной» компьютерной системы / Н.И. Листопад, Е.В. Олизарович // Доклады Белорусского государственного университета информатики и радиоэлектроники. – 2012. – № 3. – С. 23–29.

3. OpenStack Compute Administration Guide Grizzly, 2013.1 [Electronic resource] / OpenStack Foundation, 2013. – Mode of access: http://docs.openstack.org/grizzly/openstack-compute/admin/bk-compute-adminguide-grizzly.pdf. – Date of access: 01.07.2013.

4. OpenNebula 4.0 Guides [Electronic resource] / Mode of access: http://opennebula.org/documentation:rel4.0. – Date of access: 01.07.2013.

5. Piotr Siwczak, Understanding your options: Deployment topologies for High Availability (HA) with OpenStack [Electronic resource] / Mode of access: http://www.mirantis.com/blog/117072/. – Date of access: 01.07.2013.

6. OpenStack High Availability Guide [Electronic resource] / OpenStack Foundation, 2012. – Mode of access: http://docs.openstack.org/trunk/openstack-ha/openstack-ha-guide-trunk.pdf. – Date of access: 01.07.2013.

Abstract

The features of the cloud software are examined, and a diagram of creation of a virtual machine in the cloud IaaS (Infrastructure as a Service) is introduced. The issues of architecture of cloud software (OpenStack) is examined, and a structure failover clouds on the basis of multipurpose nodes is proposed.

Поступила в редакцию 05.08.2013 г.

СЕМАНТИЧЕСКАЯ МОДЕЛЬ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

УДК 004.822:514

Д.Н. Корончик, БГУИР, г. Минск

Аннотация

В работе описана семантическая модель пользовательских интерфейсов интеллектуальных систем, которая является основной частью семантической технологии проектирования пользовательских интерфейсов интеллектуальных систем. Описанная модель позволяет проектировать мультимодальные пользовательские интерфейсы на основе готовых совместимых компонентов.

Введение

Эффективность использования программной системы зависит от ее пользовательского интерфейса. В большинстве случаев разработка пользовательского интерфейса в современных системах отнимает большую часть времени затрачиваемого на разработку всей системы [1]. Трудоемкость разработки обусловлена не столько сложностью пользовательского интерфейса, сколько отсутствием хорошо продуманных технологий их проектирования.

В большинстве своем пользовательские интерфейсы современных систем являются сложными, что обусловлено сложностью самих систем. Основной проблемой в них является то, что пользователю, имеющему низкий уровень квалификации, сложно разобраться. Это в свою очередь уменьшает количество пользователей и снижает эффективность их эксплуатации. Одним из важных факторов в этом является то, что все такие пользовательские интерфейсы имеют различные принципы организации. При переходе от одной системы к другой пользователю необходимо затратить некоторое время, чтобы освоить новую систему. Кроме того для их освоения необходимо читать большое количество документации.

Пользовательский интерфейс является единственным способом взаимодействия пользователя с программной системой. Поэтому они должны быть достаточно простыми и легкими в освоении [2]. Предлагаемая в данной работе технология направлена на решение описанных выше проблем.

Семантическая модель пользовательского интерфейса

В рамках проекта OSTIS [3] разрабатывается технология проектирования пользовательских интерфейсов [11]. Основной частью этой технологии является семантическая модель, которая описывает принципы, лежащие в основе проектируемых интерфейсов. В основу семантической модели положены следующие принципы:

– пользовательский интерфейс рассматривается как специализированная интеллектуальная система, которая направлена на получение сообщений от пользователя и вывода ему ответов системы. Другими словами, основной задачей пользовательского интерфейса является перевод сообщения от пользователя, полученного на некотором внешнем языке, на язык понятный системе, а также вывод ответа системы на некотором внешнем языке, понятном пользователю;

– в основе графических интерфейсов лежит SCg-код (SemanticCodegraphical, который является одним из возможных способов визуального представления текстов SC-кода) [4]. Объекты, отображаемые на экране с помощью SCg-кода, будем называть sc.g-элементами. Основным принципом, положенным в его основу, является то, что все изображенные на экране объекты(sc.g-элементы), в том числе и элементы управления, являются изображением узлов семантической сети. Другими словами каждому изображенному на экране объекту соответствует узел в семантической сети (базе знаний);

– выделение семантики в пользовательских действиях, с последующим анализом, а также их унификация и четкая типология.

Так как пользовательский интерфейс представляет собой интеллектуальную систему, построенную с помощью семантической технологии компонентного проектирования интеллектуальных систем, то он точно так же как и любая другая система строится с использованием компонентов.

Выделены следующие типы компонентов, которые используются при построении пользовательского интерфейса:

– трансляторы с текстов различных внешних языков в тексты SC-кода (обеспечивают понимание системой информации, которая поступает от пользователя);

– трансляторы с текстов SC-кода в тексты различных внешних языков (обеспечивают перевод информации на понятный пользователю внешний язык);

– компоненты вывода информационных конструкций пользователю;

– компоненты ввода информационных конструкций.

Каждый компонент пользовательского интерфейса состоит из некоторой базы знаний необходимой для его работы и набора агентов. К примеру, для работы транслятора русского языка в SC-код и обратно, необходима база знаний, описывающая русский язык.

Наиболее развитыми в текущее время являются графические пользовательские интерфейсы. Графический интерфейс интеллектуальных систем представляет собой мультимодальный оконный интерфейс [10]. Под окном будем понимать скроллируемую область экрана, которая ограничена прямоугольником произвольного размера (как и в современных графических интерфейсах).

Взаимодействие пользователя с системой осуществляется в рамках главного окна. Главное окно принадлежит к множеству sc.g-окно (окна, внутри которых для визуализации знаний используется SCg-код). В рамках главного окна могут присутствовать другие окна. Их будем называть дочерними окнами. Визуализируются они с помощью sc.g-рамок (рисунок 1), внутри которых отображается содержимое окна на одном из внешних языков.

С помощью sc.g-рамок отображаются окна в развернутом виде. В свернутом виде они изображаются с помощью sc.g-узла с квадратом в правом нижнем углу. При работе с системой пользователь может создавать окна, относящиеся к различным классам (sc.g-окна, видео-окна и т.д.). Такие

Рисунок 1 – Пример изображения sc.g-рамки

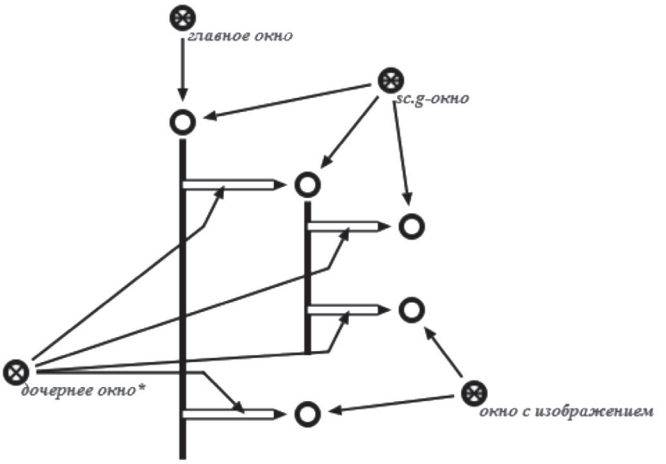


Рисунок 2 – Пример использования отношения «дочернее окно»

окна будем называть дочерними окнами. Они являются частью sc.g-конструкции в рамках некоторого sc.g-окна (в частности, главного окна). Главное окно не может быть дочерним окном по отношению к другим окнам. И одно окно не может иметь несколько родительских окон. Таким образом, они образуют древовидную иерархию, которая, в рамках базы знаний пользовательского интерфейса, задается с помощью отношения дочернее окно (рисунок 2).

Графический пользовательский интерфейс строится с использованием уже разработанных компонентов. К компонентам вывода информационных конструкций относятся просмотрщики, которые в свою очередь могут быть двух типов:

– просмотрщики содержимого sc-ссылок. Позволяют просматривать содержимое sc-ссылок (файлов), записанное в некотором формате (не SC-код);

– просмотрщики фрагментов базы знаний, представленных с помощью SC-кода. Позволяют просматривать с помощью внешнего языка информационные конструкции, представленные в SC-коде (к примеру, фрагмент базы знаний, описывающий геометрический чертеж с помощью SC-кода, может изображаться в виде sc.g-текста и в виде геометрического чертежа).

К компонентам ввода информационных конструкций относятся редакторы. Они позволяют редактировать содержимое некоторой sc-ссылки.

Главное окно представляет собой компонент просмотра фрагмента базы знаний с помощью SCg-кода и набора команд редактирования базы знаний. Другими словами, при инициировании команд редактирования в рамках главного окна происходит редактирование не на уровне внешнего представления, а напрямую в базе знаний, на уровне SC-кода, а эти изменения отображаются в главном окне.

Основной проблемой при использовании компонентов является их интеграция. Чтобы решить эту проблему, предлагается осуществлять взаимодействие между компонентами только через базу знаний. Таким образом, взаимодействуя между собой, компоненты могут лишь использовать общие ключевые узлы (понятия) в базе знаний. Такой способ интеграции компонентов позволяет разрабатывать их параллельно с минимальными зависимостями, что значительно сокращает сроки проектирования. Идея в том, что компоненты ничего не знают друг о друге. Каждый компонент лишь имеет набор агентов, которые реагируют на появление некоторой ситуации в базе знаний (сформированная пользовательская команда, вопрос и т.д.).

В пользовательском интерфейсе существует базовый способ диалога пользователя с системой. Пользователь формирует сообщения системе, явно рисуя их с помощью SCg-кода после чего данные попадают в базу знаний. На появление сообщения реагируют агенты, обрабатывая полученное сообщение. Ответы, как и вопросы, выводятся с помощью SCg-кода (в виде sc.g-конструкций). Частным видом ответа является sc.g-конструкция, состоящая из одной sc.g-рамки, содержимое которой представляется на каком-то внешнем языке. Другими словами, диалог сводится к обмену сообщениями (представленными с помощью SCg-кода) между пользователем и системой.Таким же образом пользователь может формировать команды (сворачивание, разворачивание, создание, удаление окна и т.д.). Такой способ формирования команд и вопросов является базовым и универсальным.

В процессе формирования сообщений базовым способом, пользователю приходиться выполнять часто повторяющиеся элементарные пользовательские действия (самые простейшие манипуляции с устройствами ввода: нажатие клавиши, перемещение мыши и т.д.). Очевидно, что постоянный набор сообщений вручную с помощью SCg-кода отнимает много времени и не является очень эффективным. Можно ускорить процесс набора сообщений, за счет улучшения и расширения команд редактирования sc.g-конструкций. Но это не даст значительного прироста, потому что при формировании сообщения, необходимо будет указывать дополнительную информацию. Чтобы избавиться от этой проблемы, в модель введены пользовательские команды. Пользовательская команда – последовательность элементарных пользовательских действий, которые инициируют некоторое действие в системе. К элементарным дей-

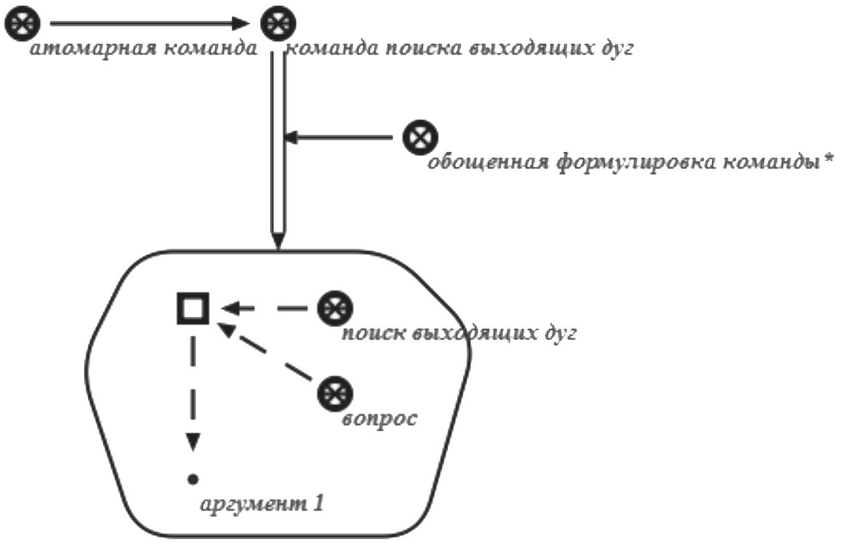


Рисунок 3 – Пример описания пользовательской команды, которая инициирует поиск всех выходящих дуг из указанного элемента

ствиям пользователя относятся: перемещение мыши, нажатие клавиш мыши и клавиатуры. Как и вся другая информация, пользовательские команды представлены с помощью SC-кода в базе знаний (рисунок 3). Инициирование пользовательской команды происходит при нажатии левой клавиши мыши на sc.g-узле, который ее изображает. При визуализации с помощью SCg-кода, пользовательские команды изображаются в виде элементов управления. Рекомендуется изображать элементы управления в виде прямоугольников, внутри которых присутствует графический или текстовый идентификатор команды. Чтобы различать различные классы команд в прямоугольнике изображается графический идентификатор класса команды.

Выделяются следующие классы пользовательских команд по типу инициируемого действия:

– команды, которые инициируют вопрос в системе;

– команды, которые инициируют действия, связанные с редактированием;

– команды, которые инициируют действия, связанные с просмотром (закрытие или открытие окон, перемещение в рамках окна, масштабирование и т.д.);

– команды, которые инициируют вывод декомпозиции* или разбиения* объекта.

Пользовательские команды могут быть атомарными и неатомарными. Неатомарная пользовательская команда – это команда, инициирование которой приводит к выводу на экран ее декомпозиции (аналог меню в дочерних подменю). Инициирование атомарных команд формирует некоторое сообщение в базе знаний системы.

По способу определения аргументов, выделяются следующие классы пользовательских команд:

– команды с заранее определенными аргументами (объектами, над которыми будет производиться инициируемое действие);

– команды с дополнительно указываемыми аргументами.

В современных пользовательских интерфейсах существующих приложений, как и в предлагаемом подходе, также есть команды для инициирования которых необходимо указывать аргументы и команды, для которых аргументы уже заранее определены. В качестве примера команд с известными аргументами (объектами действий), можно привести следующие команды: закрытие окна (окно заранее известно), сворачивание окна и т.д. В качестве команд, для которых необходимо указать аргументы, можно привести следующие: указание фрагмента текста при форматировании в редакторе Word и т.д. Проблемой является то, что в различных системах указание аргументов происходит различным (не универсальным) способом. Это в свою очередь требует от пользователя некоторого времени для освоения. В предлагаемой модели указание аргументов команд происходит явно (при этом пользователь всегда может узнать, какие аргументы необходимы для инициирования команды, задав соответствующий вопрос системе). Процесс инициирования команды сводится к следующей последовательности действий:

– указание аргументов команды (с помощью мыши);

– инициирование команды нажатием левой клавиши мыши на sc.g-узле, изображающем ее.

Все инициируемые пользователем команды попадают в базу знаний, где хранятся в протоколе пользовательских действий. Этот протокол дает возможность анализировать действия пользователя, чтобы на основе этого анализа система могла подстроиться под конкретного пользователя, для повышения качества диалога с ним [5]. За это отвечает подсистема управления диалогом.

Пользовательский интерфейс интеллектуальной системы обязан быть сам по себе интеллектуальной системой. Поэтому у него есть своя база знаний и машина обработки знаний. База знаний пользовательского интерфейса состоит из следующих разделов:

– описание синтаксиса и семантики всех используемых внешних языков;

– описание пользовательских команд;

– описание принципов работы самого пользовательского интерфейса;

– описание используемых компонентов – используются интерфейсом для организации диалога с пользователем. К примеру, при установке содержимого sc-рамки пользователю отображается список редакторов, которые позволяют его редактировать;

– протокол действий пользователя.

Машина обработки знаний пользовательского интерфейса имеет свою специфику. Кроме агентов, работающих только над базой знаний, она содержит эффекторные и рецепторные агенты. Эффекторные агенты, реагируя на изменения в базе знаний, производят изменения во внешней

среде (вывод информации). Рецепторные агенты, реагируя на изменения во внешней среде, производят изменения в базе знаний (ввод информации).

Выводы

Описанные выше принципы позволяют проектировать пользовательские интерфейсы достаточно быстро и легко. При этом созданные таким образом пользовательские интерфейсы с точки зрения пользователя внешне мало чем отличаются от тех, к которым он привык, так как:

– проектируемый пользовательский интерфейс является оконным, что присутствует во всех системах. Главное окно можно сравнить с рабочим столом операционной системы. В рамках этого окна создаются дочерние окна, в которых происходит просмотр или редактирование некоторых файлов;

– в рамках главного окна (рабочего стола) присутствуют некоторые команды, которые могут быть инициированы пользователем с указанием аргументов или же без них.

Однако предлагаемый подход имеет ряд преимуществ:

– так как в его основе лежит SCg-код, то у пользователя появляется возможность указывать и элементы управления в качестве аргументов команд. Это значительно снижает требования, предъявляемые к начальной квалификации пользователя. Ему лишь необходимо знать, как задать вопрос. Умея задавать вопросы, он может легко изучить команды и научиться работать с системой, не прибегая к чтению большого количества документации;

– унификация процесса формирования пользовательских команд, также упрощает освоение пользовательского интерфейса конечным пользователем, так как при переходе от одной системы к другой, ему не нужно заново осваивать пользовательский интерфейс;

– благодаря тому, что база знаний пользовательского интерфейса содержит описание команд и используется SCg-код, появляется возможность производить демонстрации работы с теми или иными командами. Пользователь может попросить систему показать, как пользоваться той или иной командой. При этом система, основываясь на описании этой команды, построит последовательность элементарных пользовательских действий, которая потом будет выполнена специализированными агентами. Другими словами, пользователю будет явно показано, как это сделать (система возьмет под свой контроль мышь и клавиатуру). Это также дает целый ряд преимуществ, при тестировании проектируемых интерфейсов. Так как для этого не надо записывать огромные протоколы с явным указанием элементарных действий (как это делается при тестировании современных интерфейсов). Достаточно лишь описать последовательность команд, остальное сделает сама система;

– полное описание пользовательского интерфейса с помощью SC-кода в базе знаний позволяет также решить еще одну очень важную задачу. Сейчас очень остро стоит проблема написания документации к разрабатываемым системам и их пользовательскому интерфейсу. По сути, происходит следующее: проектировщик (дизайнер) описывает пользовательский интерфейс (создает техническое задание), в котором описываются все команды, внешние языки и т.д. Далее разработчик (программист) реализует описанные команды и принципы. А на заключительном этапе технический писатель пишет по ним документацию.На всех трех этапах одни и те же вещи описываются на различных языках. В предлагаемом подходе описания команд, которые делаются разработчиком на формальном языке, сразу же и являются частью справочной системы, которая, в свою очередь, позволяет задавать вопросы по использованию самой системы. Это значительно сокращает сроки проектирования. Так как уже разработанный интерфейс и является частью справочной системы по его эксплуатации.

– разбиение пользовательского интерфейса на компоненты, которые взаимодействуют между собой через базу знаний, используя общие ключевые узлы и набор sc-операций, сводит зависимость между ними к минимуму. Снижение зависимостей между проектируемыми компонентами значительно сокращает их срок разработки, и упрощает дальнейшую поддержку;

– благодаря тому, что все действия пользователя семантически интерпретируются и записываются в протокол, появляется возможность их анализа, что делает возможным помощь пользователю при освоении пользовательского интерфейса (к примеру, система может подсказать пользователю, что некоторые действия можно делать гораздо проще). Также это дает возможность системе подстраивать пользовательский интерфейс под конкретного пользователя (показывать наиболее часто используемые команды, использовать более удобные принципы размещения элементов управления и т.д.).

Заключение

Описанные выше принципы и приемы позволяют проектировать интеллектуальные пользовательские интерфейсы для интеллектуальных систем, которые легко интегрируются в них и строятся на основе уже имеющихся компонентов. На основе предлагаемой технологии уже проектируются пользовательские интерфейсы некоторых прикладных систем [6]. Что дает возможность говорить о работоспособности предлагаемого подхода. Все результаты, в том числе и исходные коды компонентов и ядра пользовательских интерфейсов, представлены на сайте github [7, 8].

Литература:

1. Myers, B.A. Survey on User Interface Programming, Proceedings SIGCHI'92: Human Factors in Computing Systems / B.A. Myers, M.B. Rosson // Monterrey, CA, 1992. – P. 195–202.

2. Поспелов, Д.А. Интеллектуальные интерфейсы для ЭВМ новых поколений / Д.А. Поспелов // Электронная вычислительная техника. Сборник статей. Вып.3. – М.: Радио исвязь, 1989. – С. 4–20.

3. Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2013. – Режим доступа: http://www.ostis.net. – Дата доступа: 02.03.2013.

4. Голенков, В.В. Представление и обработка зна-

ний в графодинамических ассоциативных машинах / В.В. Голенков [и др]; – Минск: БГУИР, 2001.

5. Курзанцева, Л.И. Об адаптивном интеллектуальном интерфейсе «Пользователь – система массового применения» / Л.И. Курзанцева // Комп'ютернізасоби, мережі та системи. – 2008. – №7. – С. 110–116.

6. Интеллектуальная система поддержки проекта OSTIS [Электронный ресурс]. – 2013. – Режим доступа: http://www.ims.ostis.net. – Дата доступа: 02.03.2013.

7. Исходные тексты ядра пользовательских интерфейсов [Электронный ресурс]. – 2013. – Режим доступа: https://github.com/deniskoronchik/pyUI. – Дата доступа: 02.03.2013.

8. Исходные тексты web-ориентированного ядра пользовательских интерфейсов [Электронный ресурс]. – 2013. – Режим доступаhttps://github.com/deniskoronchik/sc-web. – Дата доступа: 02.03.2013.

9. Графы в программировании: обработка, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев. – Научное издание, 2003.

10. Multi-Modal Human Interactions with an Intelligent Interface Utilizing Images, Sounds and Force Feedback / Fei He, Arvin Agah // Journal of Intelligent and Robotic Systems. – Kluwer Academic Publisher, 2001. – P. 171–190.

11. Голенков, В.В. Проектирование пользовательских интерфейсов интеллектуальных систем / В.В. Голенко, В.А. Житко, Д.Г. Колб, Д.Н. Корончик // Нечеткие системы и мягкие вычисления (НСМВ – 2009). Сборник статей Третьей Всероссийской научной конференции. Том II. – С. 181–188.

Abstract

This article describes semantic model of user interface, that used to develop user interfaces for intelligent system. Described model allows to design multimodal user interfaces based on components. There are some systems that use described model for their user interfaces, one of them is ims.ostis.net.

Поступила в редакцию 04.04.2013 г.

ТРЕБОВАНИЯ К НАУЧНЫМ СТАТЬЯМ, ПУБЛИКУЕМЫМ В РАЗДЕЛЕ «РЕЦЕНЗИРУЕМЫЕ СТАТЬИ»

1. Научная статья – законченное и логически цельное произведение по раскрываемой теме – должна соответствовать одному из следующих научных направлений: информационные технологии и системы, оптоэлектроника, микро- и наноэлектроника, приборостроение.

2. Объем научной статьи не должен превышать 0,35 авторского листа (14 тысяч печатных знаков, включая пробелы между словами, знаки препинания, цифры и другие), что соответствует 8 страницам текста, напечатанного через 2 интервала между строками (5,5 страницы в случае печати через 1,5 интервала).

3. Статьи в редакцию представляются в двух экземплярах на бумаге формата А4 (220015, г. Минск, пр. Пушкина, 29Б), а также в электронном виде (e-mail: sadov@bsu.by). К статье прилагаются сопроводительное письмо организации за подписью руководителя и акт экспертизы. Статья должна быть подписана всеми авторами.

Статьи принимаются в формате doc, rtf, pdf, набранные в текстовом редакторе word, включая символы латинского и греческого алфавитов вместе с индексами. Каждая иллюстрация (фотографии, рисунки, графики, таблицы и др.) должна быть представлена отдельным файлом и названа таким образом, чтобы была понятна последовательность ее размещения. Фотографии принимаются в форматах tif или jpg (300 dpi). Рисунки, графики, диаграммы принимаются в форматах tif, cdr, eps или jpg (300 dpi, текст в кривых). Таблицы принимаются в форматах doc, rtf или Excel.

4. Научные статьи должны включать следующие элементы:

аннотацию;

фамилию и инициалы автора (авторов) статьи, ее название;

введение;

основную часть, включающую графики и другой иллюстративный материал (при их наличии);

заключение;

список цитированных источников;

индекс УДК;

аннотацию на английском языке.

5. Название статьи должно отражать основную идею выполненного исследования, быть по возможности кратким, содержать ключевые слова, позволяющие индексировать данную статью.

6. Аннотация (100–150 слов) должна ясно излагать содержание статьи и быть пригодной для опубликования в аннотациях к журналам отдельно от статьи.

В разделе «Введение» должен быть дан краткий обзор литературы по данной проблеме, указаны не решенные ранее вопросы, сформулирована и обоснована цель работы.

Основная часть статьи должна содержать описание методики, аппаратуры, объектов исследования и подробно освещать содержание исследований. Полученные результаты должны быть обсуждены с точки зрения их научной новизны и сопоставлены с соответствующими известными данными. Основная часть статьи может делиться на подразделы (с разъяснительными заголовками).

Иллюстрации, формулы, уравнения и сноски, встречающиеся в статье, должны быть пронумерованы в соответствии с порядком цитирования в тексте.

В разделе «Заключение» должны быть в сжатом виде сформулированы основные полученные результаты с указанием их новизны, преимуществ и возможностей применения.

Список цитированных источников располагается в конце текста, ссылки нумеруются согласно порядку цитирования в тексте. Порядковые номера ссылок должны быть написаны внутри квадратных скобок (например: [1], [2]).

В соответствии с рекомендациями ВАК Республики Беларусь от 29.12.2007г. №29/13/15 научные статьи аспирантов последнего года обучения публикуются вне очереди при условии их полного соответствия требованиям, предъявляемым к рецензируемым научным публикациям.

«ОБЛАЧНЫЕ» ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 004.75

С.В. Абламейко, Ю.И. Воротницкий, БГУ, г. Минск,
Н.И. Листопад, Главный информационно-аналитический центр
Министерства образования Республики Беларусь, г. Минск

Аннотация

В статье рассматриваются основные отличительные черты и функциональные возможности «облачных» технологий хранения и обработки информации. Анализируются возможности применения «облачных» технологий для реализации модели мобильного обучения, предусмотренной Концепцией информатизации системы образования Республики Беларусь (принятой в 2012 году) на период до 2020 года.

Введение

Современные «облачные» технологии хранения и обработки информации находят широкое применение в различных областях человеческой деятельности: в науке и образовании, на производстве, в социальной сфере и т.д. Фактически сегодня «облачные» системы стали неотъемлемой частью информационно-коммуникационной инфраструктуры (ИКИ) информационного общества.

В системе образования ИКИ является технической платформой образовательной парадигмы последних десятилетий – «образование на протяжении всей жизни». Эффективная реализация этой парадигмы предполагает возможность доступа обучаемых к образовательным ресурсам повсюду и в любое время. Такой доступ обеспечивается в рамках модели мобильного обучения [1], развитие и внедрение которой предусмотрено Концепцией информатизации системы образования Республики Беларусь на период до 2020 г., принятой в 2013 г. [2]. В Концепции, в частности, отмечается следующее.

Применение «облачных» технологий в системе образования позволит обеспечить мобильность и актуальность образовательных ресурсов. Для учебных заведений «облачная» образовательная среда обеспечит возможность без дополнительных затрат использовать современные и постоянно актуализируемые компьютерную инфраструктуру, программные средства и сервисы, предоставляемые ЦОД. Соответственно, будут снижены затраты учебных заведений на построение и сопровождение локальных информационных инфраструктур. «Облачные» технологии позволят вовлечь в образовательный процесс личные компьютерные устройства преподавателей, учащихся и их родителей.

Такой подход дает возможность создать удобную среду для доступа к ресурсам с разнообразных, в том числе мобильных, устройств и обеспечить синхронизацию деятельности пользователя, осуществляемой с разных устройств (компьютер в учебном классе, домашний компьютер, личный планшет или смартфон).

1 «Облачная» информационно-образовательная среда

Концепция «облачной» информационно-образовательной среды национальной системы образования предложена в [3] и развивается в работах [4, 5].

Мобильность обучения предполагает создание для каждого субъекта системы образования – учащегося, родителя, педагога, руководителя – персональной информационной среды, не привязанной к конкретному компьютерному устройству и инвариантной относительно места доступа. «Облачные» технологии позволяют создать удобную среду для доступа к ресурсам и сервисам с разнообразных, в том числе мобильных, устройств и обеспечить синхронизацию деятельности пользователя, осуществляемой с нескольких устройств (компьютер в учебном классе, домашний компьютер, смартфон и т.п.).

Внедрение «облачных» технологий предполагает, что хранение, сопровождение информационных ресурсов, организация доступа к ним, а также предоставление различных сервисов будут сосредоточены на платформе одного или нескольких центров обработки данных (ЦОД) системы образования. Доступ к ресурсам и сервисам осуществляется через национальные научно-образовательные сети (НИКС, UNIBEL, Bas-Net, BSUNet) и сеть Интернет. Отличительными особенностями «облачных» технологий являются следующие признаки [6]:

– сервисная модель обслуживания – представление сетевых ресурсов в виде пула настраиваемых сервисов, готовых к немедленному использованию на условиях online подписки без дополнительной установки и настройки программного обеспечения со стороны пользователя (здесь пока есть проблемы именно для планшетов и смартфонов);

– самообслуживание – возможность для потребителя самостоятельно изменять номенклатуру и конфигурацию сервисов в режиме online в соответствии с текущими потребностями;

– высокая автоматизация процесса управления пулом сервисов, учетными записями пользователей и потреблением ресурсов;

– эластичность – возможность динамического перераспределения имеющихся ресурсов между потребителями; при этом внутренняя техническая структура «облака» скрыта от потребителя и недоступна ему для модификации, а само расширение доступных ресурсов является прозрачным;

– использование распространенных сетевых технологий – «облачные» сервисы доступны для любого клиентского оборудования с использованием стандартных технологий и протоколов, поддерживающих стек протоколов TCP/IP.

С точки зрения пользователя отличием работы в «облачной» среде от использования традиционных сетевых ресурсов также является универсальный интерфейс, ориентированный на web-технологии и http-протокол в качестве базовых средств управления «облаком» и доступа к его сервисам.

Обычно выделяют следующие базовые классы «облачных» сервисов [20].

1. IaaS (Infrastructure as a Service) – «инфраструктура как услуга», клиенту предоставляется полный доступ к виртуальной машине с возможностью устанавливать и настраивать операционную систему (ОС) и любое про-

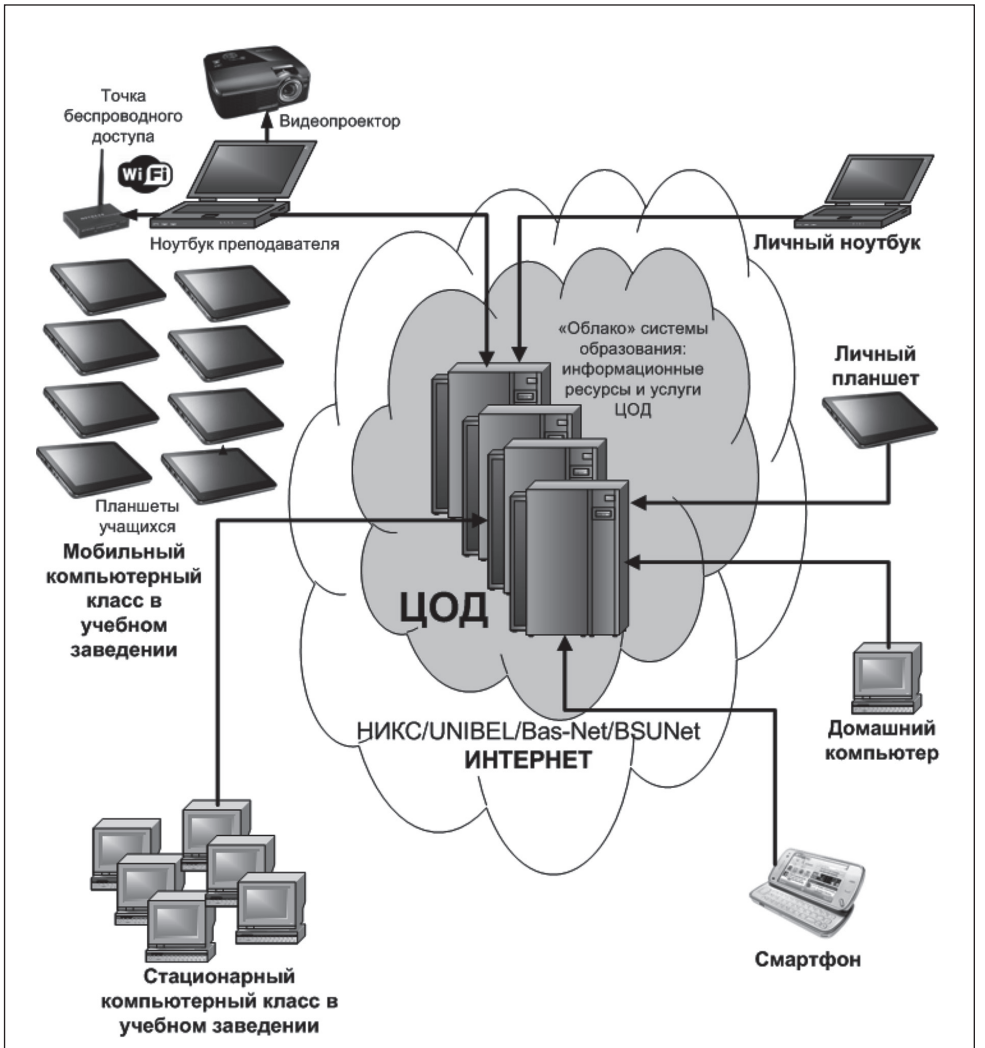


Рисунок 1 – Схема образовательного «облака»

граммное обеспечение (ПО). Модель IaaS предполагает, что клиент самостоятельно может управлять количеством процессоров, объемами оперативной памяти, дискового пространства и сетевых коммуникаций виртуальной машины. Сервис IaaS предполагает полную ответственность клиента за безопасность, работоспособность и законность использования ПО. На оператора «облака» возлагается лишь ответственность за безопасность и надежность функционирования аппаратной платформы.

2. PaaS (Platform as a Service) – «платформа как услуга», предоставляет клиенту ограниченный доступ к управлению ОС, удаленным рабочим столом (DaaS, desktop as a service), СУБД и т.д. В этом случае на оператора «облака» возлагается установка и настройка системного ПО, соблюдение соответствующих лицензионных соглашений и обеспечение мер безопасности. Клиент же имеет возможность устанавливать, настраивать и использовать прикладное ПО, несет ответственность за его безопасность и соблюдение лицензионных прав.

3. SaaS (Software as a Service) – «прикладное ПО как услуга», предоставляет online-доступ к использованию прикладного ПО. При этом, настройка ПО, обеспечение мер безопасности и соблюдение лицензионных соглашений возлагается на оператора «облака».

Совокупность сервисов, предоставляемых образовательным «облаком» конкретному пользователю, формируют его персональную информационно-образовательную среду. Доступ к этой среде осуществляется повсеместно (рисунок 1).

Программно-техническая инфраструктура «облака» строится на основе центров обработки данных (ЦОД). В зависимости от размещения и принадлежности ЦОД, порядка предоставления доступа к сервисам и способа организации работы клиента, выделяются корпоративные или специализированные «частные облака» (private cloud), универсальные «публичные облака» (public cloud), совместно используемые «общие облака» (common cloud) и смешанный тип обслуживания – «гибридные облака» (hybrid cloud).

С точки зрения пользователя отличием работы в «облачной» среде от использования традиционных сетевых ресурсов также является универсальный интерфейс, ориентированный на web-технологии и http-протокол в качестве базовых средств управления «облаком» и доступа к его сервисам.

2 Основные направления применения «облачных» технологий в образовании

Применение «облачных» технологий в системе образования позволяет решить две основные задачи. Во-первых, обеспечить для образовательных учреждений и отдельных учащихся возможность использовать современные и постоянно актуализируемые компьютерную инфраструктуру, программные средства, электронные образовательные ресурсы и сервисы. Во-вторых, снизить затраты отдельных учебных заведений и системы образования в целом на построение локальных информационных инфраструктур за счет эффективного использования вычислительных ресурсов, сосредоточенных в «облаке» и эластично выделяемых пользователям в соответствии с их запросами.

В работе [4] рассмотрены основные перспективные приложения «облачных» технологий в образовании.

Непрерывный доступ учащихся к образовательным ресурсам. Клиент – серверные архитектуры в совокупности с веб-технологиями позволяют предоставить

пользователям, работающим с различными компьютерными устройствами, гомогенную масштабируемую среду для доступа к вычислительным и информационным ресурсам. Современная молодежь, стремящаяся постоянно присутствовать в сети Интернет, должна получить адекватные механизмы использования имеющихся компьютерных устройств для образования. В настоящее время в Республике Беларусь подавляющее большинство учащихся и студентов имеют персональные компьютеры и обеспечены широкополосным доступом в сеть Интернет. Широкомасштабное вовлечение в образовательный процесс персональных устройств позволит существенно сократить издержки на оснащение компьютерами и лицензионными программными продуктами компьютерных классов в учебных заведениях, а также отвлечь молодежь от неэффективного использования персональных устройств (например, для общения в социальных сетях и т.п.).

Разработка и внедрение современных электронных образовательных ресурсов. В настоящее время в национальной системе образования накоплен значительный опыт разработки электронных средств обучения (ЭСО). Можно выделить следующие основные направления работ по созданию ЭСО за последние 5 лет.

Первое направление – разработка сетевых ЭСО для средней школы. В настоящее время работы по созданию сетевых ЭСО ведутся в рамках реализации проектов по подпрограмме «Электронное обучение и развитие человеческого капитала» Национальной программы ускоренного развития услуг в сфере информационно-коммуникационных технологий на 2011–2015 годы, а также в рамках программы по созданию электронного контента. Последняя программа реализуется Национальным институтом образования.

К основным мировым тенденциям развития ЭСО можно отнести широкое использование мультимедийных материалов, интерактивной графики, интеграция в ЭСО различных файлов (презентаций, электронных таблиц, графиков и др.), а также использование сетевых систем доступа к ЭСО, их актуализации и подписки на них. Актуальной является задача создания национального «облачного» репозитария электронных образовательных ресурсов на основе общих для учебных заведений и преподавателей соглашений, стандартов и технологий. Очевидно, что такая образовательная среда может быть эффективно построена на основе национальной «облачной» научно-образовательной инфраструктуры.

Второе направление – разработка автономных электронных учебно-методических комплексов (УМК) для высших учебных заведений, полностью охватывающих содержание дисциплин, включая теоретический курс, задачи и задания семинарских и практических занятий, задания лабораторного практикума, материалы для самостоятельной работы и т.п. Такие ЭСО просты в использовании и не требуют усилий для их инсталляции. Основные недостатки – отсутствие механизмов обновления и большой объем локально хранимых файлов при включении в состав ЭСО мультимедийных материалов. Кроме того, отсутствуют возможности анализа результатов тестирования учащихся, построения на основе этих результатов индивидуальных траекторий обучения.

Третье направление связано с организацией дистанционного обучения. Использование систем дистанционного обучения, как правило, реализованных в клиент-серверной архитектуре, позволяет организовать сетевой доступ к учебным материалам и реализовать основные функции управления учебным процессом. Основной проблемой внедрения таких систем в отдельных учебных заведениях, на наш взгляд, являются относительно высокие материальные и трудовые затраты на освоение и сопровождение современных систем дистанционного обучения.

Хостинг информационных ресурсов и информационное взаимодействие субъектов образовательного процесса. Можно утверждать, что объем информационных ресурсов, публикуемых белорусскими учебными заведениями в сети Интернет, подчиняется общим законам экспоненциального роста ресурсов этой глобальной сети. При этом растут затраты на серверное оборудование и широкополосные каналы для исходящего трафика в Интернет, которые могли бы обеспечить хранение больших объемов информации и доступ к ним. Создание в системе образования единого национального ЦОД позволит существенно снизить эти затраты, а также повысить безопасность хранимых ресурсов, снизить требования к квалификации ИТ-персонала учебных заведений. Кроме того, на базе такого «облачного» ЦОД могли быть реализованы сервисы, обеспечивающие информационное взаимодействие преподавателей и студентов.

Управление в системе образования. На протяжении последних лет в национальной системе образования сформирован целый ряд систем управления и интегрированных баз данных, на основе которых юридическим и физическим лицам могут предоставляться те или иные информационные сервисы. Одним из наиболее эффективных инструментов предоставления таких сервисов являются «облачные» системы. Так, в 2013 г. Министерством образования запланирована апробация «облачной» системы информационного обеспечения учебного процесса «Schools.By». В конечном итоге можно прогнозировать уже в ближайшем будущем переход от автономных систем управления учебным процессом, функционирующих в локальных сетях учебных заведений, к централизованным «облачным» системам.

3 Информационно-коммуникационная инфраструктура учебных заведений

«Облачные» технологии оказывают существенное влияние на информационно-коммуникационную инфраструктуру учебных заведений, в первую очередь, относительно небольших. При этом основу модернизированной инфраструктуры составит уже существующая типовая конфигурация компьютерной сети учебного заведения.

Существенно упрощается функционал компьютерной сети. С нее постепенно снимаются функции хранения информационных ресурсов и, соответственно, обеспечения их безопасности. Для хранения этих ресурсов каждому субъекту системы образования – ученику, педагогу, руководителю, учебному заведению в целом, районному, городскому, областному управлению образования и т.п. – в центральном «облаке» выделяется место для хранения. Отпадает необходимость в тира-

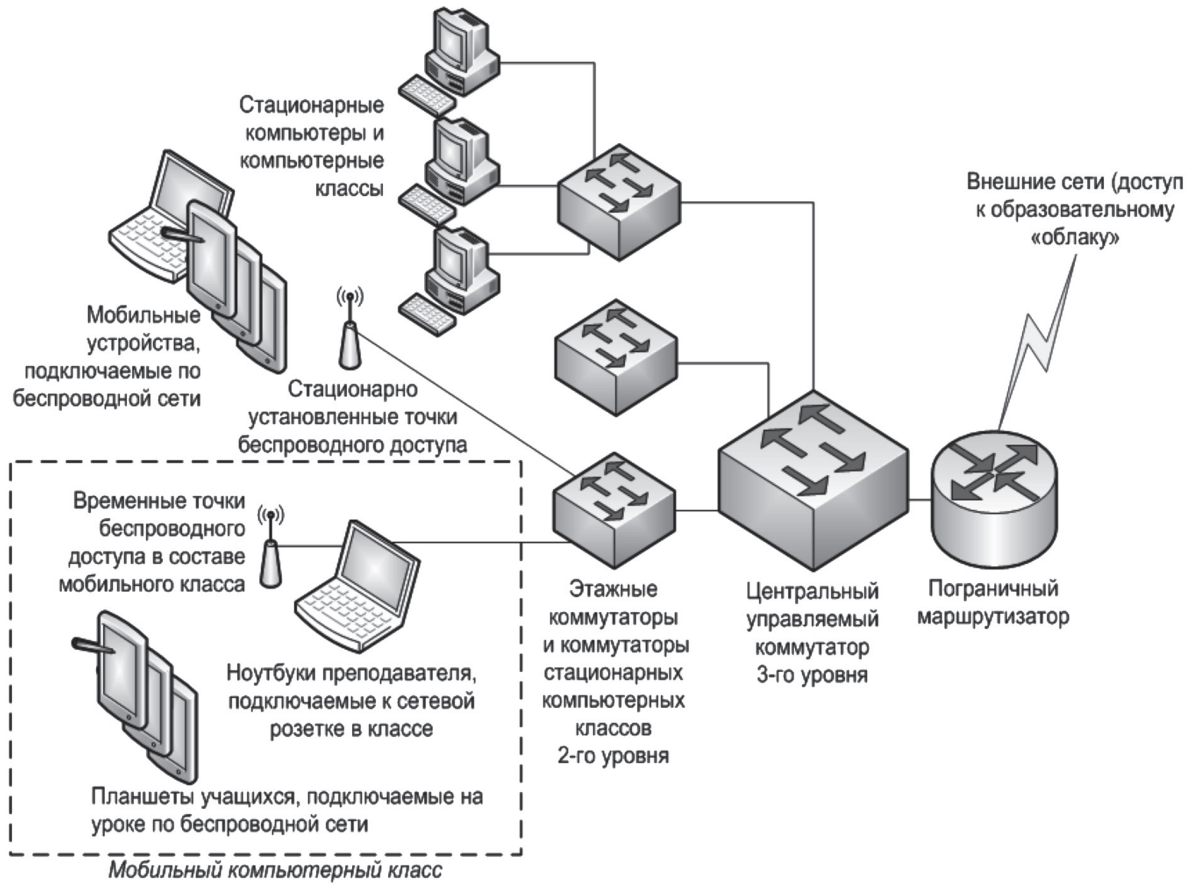


Рисунок 2 – Типовая компьютерная сеть школы, интегрируемая в «облачную» инфраструктуру

жировании и распространении электронных учебников и учебных пособий, обучающих программ и других образовательных электронных ресурсов, – все они хранятся в «облаке» ЦОД. Авторизованный доступ к этим ресурсам предоставляется через национальные научно-образовательные сети и Интернет. Соответственно, в учебных заведениях исчезает необходимость принимать специальные меры для обеспечения безопасности информационных ресурсов, выполнять их резервное копирование и т.п. В конечном итоге, в большинстве небольших учебных заведений отпадет необходимость в выделенных серверах, и типовая архитектура проводной сети будут выглядеть так, как показано на рисунке 2 на примере сети общеобразовательной школы.

4 Модели организации доступа учебных заведений во внешние сети

Стратегически правильным было бы развитие сети Министерства образования UNIBEL в регионах, организация собственных точек доступа и подключение к ним учебных заведений. Таким образом, в рамках сети UNIBEL учебные заведения могли бы получить доступ к «облаку» ЦОД и через него – в Интернет. Данная модель является наиболее привлекательной и с точки зрения информационной безопасности. Однако в силу большого числа школ, их территориальной разобщенности и высокой стоимости такого решения, оно может рассматриваться как перспективное. В ближайшее время могут быть реализованы две модели доступа учебных заведений к «облачным» ресурсам системы образования и в Интернет.

Первая предполагает, что операторы связи устанавливают для каждого учебного заведения VPN-соединение с центральными ЦОД. Доступ в Интернет обеспечивается для всех учреждений образования через единый шлюз и централизованную систему безопасности ЦОД. Таким образом, по сути, формируется корпоративная сеть системы образования с централизованным управляемым доступом во внешние сети (рисунок 3). Преимуществом данного решения является высокая степень управляемости и безопасности доступа учебных заведений в Интернет.

Альтернативой является непосредственный доступ учебных заведений в Интернет и, через Интернет, к ресурсам и сервисам «облака» ЦОД. В этом случае подключение к сети Интернет обеспечивается операторами связи на местах. Такое решение проще, унифицировано с доступом пользователей извне учебных заведений, однако, менее управляемо и безопасно.

Заключение

«Облачные» компьютерные системы представляют собой новый способ организации информационно-

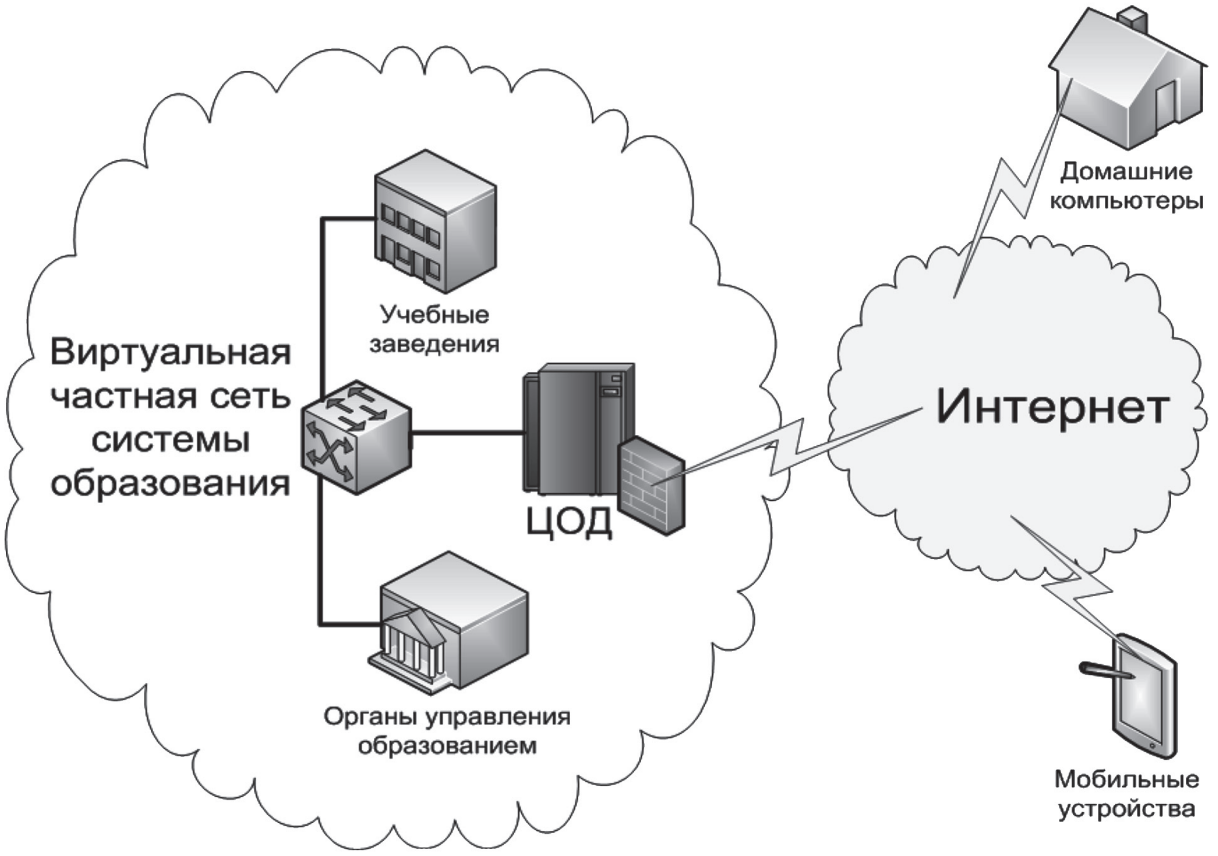


Рисунок 3 – Модель централизованного доступа в Интернет (через ЦОД) для учреждений и органов управления системы образования

коммуникационной инфраструктуры, характеризующийся упрощением и унификацией методов, средств и способов работы пользователя. Основными практическими преимуществами использования «облачных» систем в образовании являются: снижение требований к техническому оснащению и квалификации пользователей, создание условий для мобильности учащихся, студентов, учителей, оптимизация использования дорогостоящего высокопроизводительного оборудования и программного обеспечения, упрощение процессов управления лицензиями и обновлениями, стандартизация выполнения операций в рамках системы менеджмента качества.

Литература:

1. UNESCO policy guidelines for mobile learning // Paris: UNESCO, 2013.

2. Концепция информатизации системы образования Республики Беларусь на период до 2020 г. // Официальный интернет-портал Министерства образования Республики Беларусь [Электрон. ресурс] / Режим доступа: http://www.edu.gov.by/sm.aspx?guid=437693.

3. Абламейко, С.В. Перспективы применения «облачных» технологий в системе образования Республики Беларусь / С.В. Абламейко, Ю.И. Воротницкий, Н.И. Листопад // Четвертая Международная научная конференция «Суперкомпьютерные системы и их применение» (SSA'2012), 23-25 октября 2012 года, Минск: доклады. – Минск: ОИПИ НАН Беларуси, 2012. – С. 29–36.

4. Абламейко, С.В. «Облачная» концепция информатизации системы образования Республики Беларусь / С.В. Абламейко, Ю.И. Воротницкий, А.Н. Курбацкий, Н.И. Листопад // Информатизация образования. – 2012. – № 3. – С. 13–29.

5. Воротницкий, Ю.И. Мобильные компьютерные устройства в «облачной» информационно-образовательной среде общеобразовательной школы: монография / Ю.И. Воротницкий, М.Г. Зеков, А.Н. Курбацкий; под ред. проф. А.Н. Курбацкого. – Минск: РИВШ, 2012.

6. Buyya, R. Cloud Computing: Principles and Paradigm, / Rajkumar Buyya, James Broberg, Andrzej Gościński // John Wiley&Sons. – 2011.

Abstract

The paper examines the main characteristics and functionality of the «cloud» technologies for information's storage and processing. The possibilities of using «cloud» technology to implement the model of mobile learning provided by the «Blueprint for the Education System of the Republic of Belarus Informatization for the period up to 2020» are analyzed.

Поступила в редакцию 10.08.2013 г.

УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПРОЕКТАМИ НА ОСНОВЕ ОБЛАЧНОГО СЕРВИСА PAAS СУПЕРКОМПЬЮТЕРА СКИФ-БГУ

УДК 004.4: 004.9

А.В. Жерело, В.П. Кочин,
БГУ, г. Минск

Аннотация

В статье рассмотрены особенности реализации облачного сервиса «платформа как услуга», представлена схема взаимодействия основных сервисов суперкомпьютера СКИФ-БГУ с разработанным сервисом, а также определены пути дальнейшего развития облачных сервисов суперкомпьютера СКИФ-БГУ.

Введение

Основными направлениями применения суперкомпьютерных решений в БГУ являются использование вычислительных ресурсов для проведения сложных научных расчетов и обеспечение подготовки специалистов, обладающих знаниями в области параллельных вычислительных систем и навыками их практического использования в различных предметных областях [1]. В последнее время наблюдается рост числа пользователей суперкомпьютера, количества выполняемых задач и их вычислительной сложности, что, в свою очередь, приводит к росту конкуренции за ресурсы суперкомпьютера [2]. Проблема недостатка ресурсов усугубляется тем, что не все пользователи обладают необходимыми знаниями в области разработки и эксплуатации приложений, использующих суперкомпьютерные технологии. Совокупность указанных выше факторов приводит к нарушению функциональности суперкомпьютера, в частности, к проблеме неэффективного распределения нагрузки, которая, в свою очередь, приводит к проблеме недоступности ресурсов отдельных узлов.

Для решения данных проблем в БГУ был создан облачный сервис «Платформа как услуга» (PaaS), интегрированный с суперкомпьютером СКИФ-БГУ [3].

Общая архитектура решения

Сервис PaaS предназначен для ведения разработок программных продуктов, ориентированных на использование суперкомпьютерных и ГРИД-технологий, в частности, на использование систем распределенных вычислений на базе стандарта MPI.

Основным назначением сервиса является предоставление пользователю доступа к среде разработки и исполнения программных продуктов, идентичной функционирующей на СКИФ К1000-2, средствам совместного ведения разработки и запуска готовых решений, механизмам балансировки вычислительной нагрузки.

Разработанный сервис создан на выделенном кластере серверов обслуживания, интегрированных с суперкомпьютером посредством системы очередей. Элементами PaaS являются:

1. Сервер сопровождения разработки. На сервере размещаются следующие службы:
– система управления учетными записями пользователей;
– служба контроля версий;
– система непрерывной интеграции приложений;
– служба безопасного доступа;
– система мониторинга;
– служба электронной почты.

С целью обеспечения надежного хранения разрабатываемых приложений и безопасного устойчивого сетевого доступа к ресурсам PaaS был выбран сервер, построенный на двух четырехядерных процессорах Intel Xeon 5600, с 16 Гб оперативной памятью и 2,5 ТБ жестким диском.

2. Сервер разработки. Сервер предназначен для сборки тестируемых приложений и для запуска приложений в режиме отладки. В связи с тем, что данный элемент используется в качестве подготовительной площадки перед использованием приложения на кластере, на сервере разработки производится установка вычислительной среды полностью соответствующей среде вычислительных элементов СКИФ К1000-2. Так как сервер осуществляет сборку и предварительный запуск проектов, и, как следствие, должен обладать значительными вычислительными мощностями, архитектурно соответствующими параметрам СКИФ К1000-2, рекомендуется использовать сервер с двумя шестиядерными процессорами Intel 5520, 16 Гб оперативной памяти, 2,5 ТБ жесткий диском.

3. Вычислительные элементы суперкомпьютера СКИФ К1000-2 в количестве 30 элементов, используемые для планового запуска подготовленных приложений. Во избежание перегруженности элементов суперкопьютера для запуска готового ПО будет создана отдельная очередь службы OpenPBS.

Принципиальная схема взаимодействия основных подсистем показана на рисунке 1.

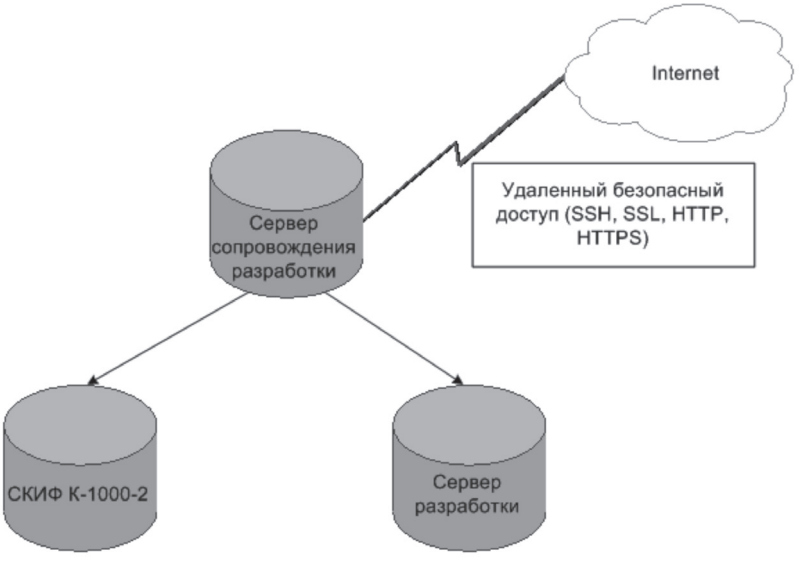


Рисунок 1 – Схема взаимодействия основных подсистем

В качества централизованной системы контроля версий была выбрана система Subversion (SVN). В настоящее время SVN является ведущей платформой, используемой группами разработчиков как открытого, так и коммерческого ПО, в частности, Apache, GCC, Assembla и т.д. SVN является продолжением линейки продуктов RCS, CVS и, как следствие, устраняет ряд проблем, возникающих при осуществлении совместной разработки [4].

В отличие от других систем, в частности, GIT, SVN является централизованной файлоориентированной системой (основными элементами хранения являются файлы и каталоги). Такой подход позволяет хранить внутри системы и текстовые, и двоичные файлы, что является преимуществом в случае использования в процессе разработки уже готовых библиотек.

Подсистемой запуска приложений на сервере разработки и на кластере является OpenPBS, интегрированный с СКИФ К1000-2. Основное назначение системы пакетной обработки заданий состоит в запуске программы на исполнение на тех узлах кластера, которые в данный момент не заняты обработкой других заданий, и в буферизации задания, если в данный момент отсутствуют свободные ресурсы для его выполнения. Большинство подобных систем предоставляют и множество других полезных услуг.

Важнейшим достоинством системы OpenPBS является достаточно удобная и эффективная поддержка вычислительных узлов разной конфигурации и архитектуры.

В качестве инструмента непрерывной интеграции проекта была выбрана система Jenkins-CI. Выбор системы был обусловлен тем, что данная система является системой с открытым исходным кодом, распространяется под свободной лицензией и предоставляет гибкую систему модификации процесса сборки за счет использования дополнительных модулей, что позволяет организовать сборку решений не только на базе основных для MPI решений языков (C, Fortran, C++). Еще одним достоинством системы является ее кроссплатформенность, так как основа системы реализована с использованием языка Java [5].

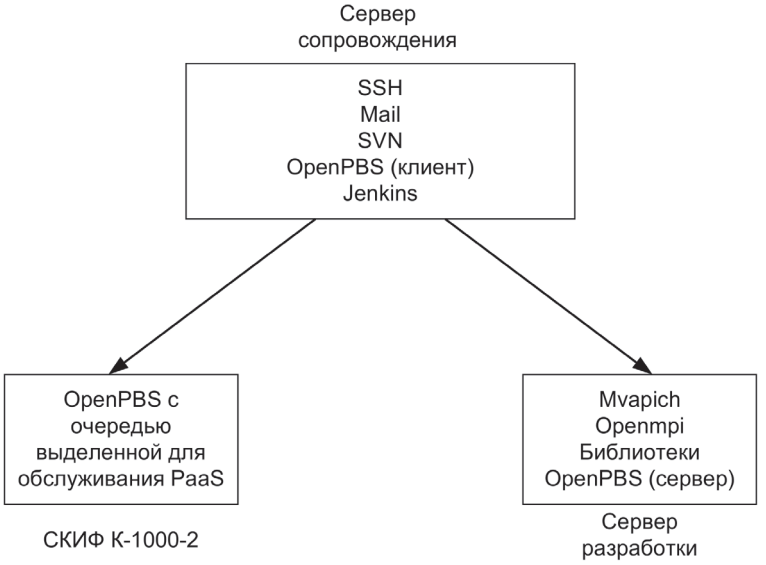


Рисунок 2 – Схема расположения и взаимодействия сервисов

Система обеспечивает требуемую периодическую и триггерную сборку проектов, осуществляет мониторинг, формирование отчетности по результатам сборки и обеспечивает своевременное оповещение о результатах участников проекта за счет возможности интеграции с системой электронной почты.

Схема расположения и взаимодействия сервисов показана на рисунке 2.

Заключение

Реализация облачного сервиса «Платформа как услуга» позволит упростить доступ для пользователей к ресурсам суперкомпьютера, а также интегрировать суперкомпьютерные ресурсы в единое «облако» БГУ. Создание сервиса создает предпосылки для реализации других облачных сервисов, что, в свою очередь, приведет к росту числа пользователей суперкомпьютера, а также упрощению работы с ним.

Литература:

1. Воротницкий, Ю.И. Суперкомпьютерные технологии в образовательном процессе и научных исследованиях в БГУ / Ю.И. Воротницкий, А.В. Жерело, В.П. Кочин, Г.Г. Крылов, Л.З. Утко // Третья международная научная конференция «Суперкомпьютерные системы и их применение» (SSA'2010), 25-27 октября 2010, Минск: доклады / Минск: ОИПИ НАН Беларуси, 2010.

2. Воротницкий, Ю.И. Суперкомпьютерные технологии в учебном процессе классического университета / Ю.И. Воротницкий, В.П. Кочин, А.В. Жерело, Г.Г. Крылов // Четвертая международная научная конференция «Суперкомпьютерные системы и их применение» (SSA'2012), 23-25 октября 2012, Минск: доклады. – Минск: ОИПИ НАН Беларуси, 2012. – С. 84–87.

3. Абламейко, С.В. Перспективы применения «облачных» технологий в системе образования Республики Беларусь / С.В. Абламейко, Ю.И. Воротницкий, Н.И. Листопад // Четвертая Международная научная конференция «Суперкомпьютерные системы и их применение» (SSA'2012), 23-25 октября 2012 года, Минск: доклады. – Минск: ОИПИ НАН Беларуси, 2012. – С. 29–36.

4. Open Source Software Engineering Tools, [Electronic resource] / Subversion, 2013. – Mode of access: http://subversion.tigris.org/. – Date of access: 01.05.2013.

5. Jenkins CI, [Electronic resource] / Jenkins, 2013. – Mode of access: http://jenkins-ci.org/. – Date of access: 01.05.2013.

Abstract

The article describes the features of the cloud service «platform as a service» is a diagram of the interaction of core services SKIF-BSU with the developed service, as well as the ways of further development of cloud services SKIF-BSU.

Поступила в редакцию 05.08.2013 г.

СИСТЕМА УЧЕТА И АНАЛИЗА ТРАНЗАКЦИОННЫХ ДАННЫХ ПРОЦЕССОВ УПРАВЛЕНИЯ ОСНОВНЫМИ СРЕДСТВАМИ ПРЕДПРИЯТИЯ НА ОСНОВЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ И OLAP-ТЕХНОЛОГИЙ

А. Козлов, г. Минск

Общая структура системы

В последнее время одними из самых актуальных проблем являются проблемы сбора и анализа информации. Классические аналитические системы являются дорогостоящими и, соответственно, слабо применимы к предприятиям малого и среднего бизнеса, так как требуют много затрат на покупку оборудования, установку, поддержку и модернизацию программного обеспечения. Также остро стоит вопрос сохранности данных.

При помощи OLAP (англ. – online analytical processing, аналитическая обработка в реальном времени) систем можно получить ряд данных, которые будут полезными при построении долгосрочных планов развития предприятия. В самом общем случае, примерами полученных данных может быть статистика выполнения планов различными сотрудниками предприятия за определенный период. На основании этих данных могут быть приняты такие решения, как премирование или увольнение определенных сотрудников. С использованием OLAP систем могут быть построены практически любые статистики, которые будут полезны при принятии стратегических решений в управлении предприятием.

В общем случае, система учета и анализа данных должна состоять из следующих модулей:

– система учета транзакционных данных процессов управления основными средствами на основе распределенной обработки данных;

– модуль анализа данных процессов управления основными средствами на основе системы интегрированных OLAP-серверов;

– система предварительной обработки и загрузки данных.

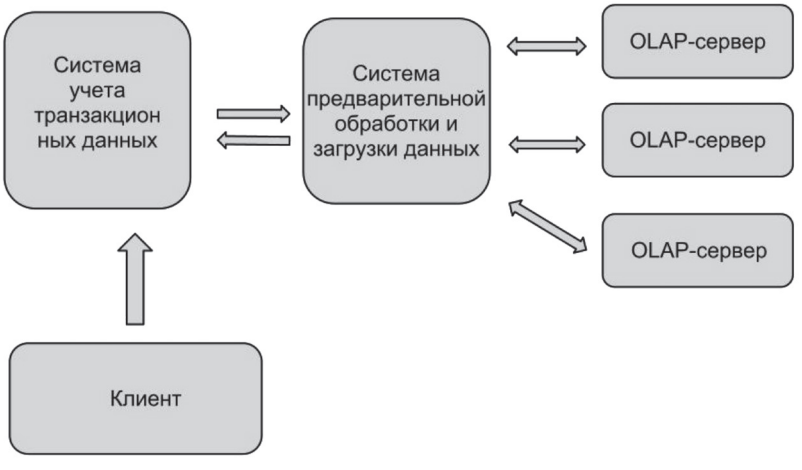


Рисунок 1 – Модель системы аналитической обработки данных

Структура системы представлена на рисунке 1. Так как потребителями будут являться предприятия малого и среднего бизнеса, очень важно, чтобы эти клиенты не несли затраты на покупку оборудования и поддержку системы. В таком случае оптимальным будет разработка системы в рамках концепции SaaS (SoftwareasaService – программное обеспечение как сервис, одна из моделей облачных вычислений), при которой все клиенты будут работать с одним ядром системы через веб интерфейс, а поддержкой системы будет заниматься провайдер услуги.

Система предварительной обработки и загрузки данных является центральной частью всей аналитической системы. Главной функцией данной системы является передача данных из хранилища системы учета в базы данных OLAP-серверов аналитического модуля. Также одной из основных задач системы предварительной обработки данных является контроль нагрузки на аналитический модуль. Система должна подключать новые OLAP-сервера в период пиковой загрузки и отключать неиспользуемые ресурсы во время простоя системы.

Введение в облачные вычисления

Облачные вычисления – это модель обеспечения сетевого доступа к общему пулу сетевых ресурсов, которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами. В настоящее время существует 3 основных модели предоставления услуг на базе облачных технологий (рисунок 2).

– IaaS (Infrastructure as a Service) – инфраструктура как сервис;

– PaaS (Platform as a Service) – платформа как сервис;

– SaaS (SoftwareasaService) – программное обеспечение как сервис.

Инфраструктура как сервис (IaaS) – это предоставление компьютерной инфраструктуры (как правило, в форме виртуализации) как услуги на основе облачных вычислений. В основе IaaS концепции находятся аппаратные средства (серверы, системы хранения данных, сетевое оборудование) и системное программное обеспечение (средства виртуализации, автоматизации и средства управления ресурсами).

Платформа как сервис (PaaS) – это предоставление интегрированной платформы для разработки, тестирования, развертывания и поддержки веб-приложений как услуги. Одним из главных плюсов является отсутствие затрат на приобретение, поддержку и модернизацию программного

Рисунок 2 – Модели облачных систем

обеспечения и оборудования. Приложения, построенные на основе PaaS услуг, масштабируемы, отказоустойчивы и безопасны, а интегрированная платформа для разработки, тестирования и разворачивания приложения упрощает процесс разработки.

Программное обеспечение как сервис (SaaS) – бизнес-модель продажи и использования программного обеспечения, при которой поставщик разрабатывает веб-приложение и самостоятельно управляет им, предоставляя заказчикам доступ к программному обеспечению через Интернет.

Основные типы OLAP-систем

OLAP – технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу. По типу используемой базы данных OLAP-системы можно разбить на 3 большие группы:

– многомерная OLAP (Multidimensional OLAP – MOLAP);

– реляционная OLAP (Relational OLAP – ROLAP);

– гибридная OLAP (Hybrid OLAP – HOLAP).

MOLAP – это классическая форма OLAP-системы. Данный тип системы так же называют просто OLAP. Эта форма использует в качестве хранилища многомерную базу данных, в которой данные денормализированы. При загрузке данных в хранилище MOLAP, они проходят предварительную обработку ETL-сервером (Extract, Transform, Load – извлечение, преобразование, загрузка). OLAP-структура, созданная из рабочих данных, называется OLAP-куб.

ROLAP работает напрямую с реляционным хранилищем, факты и таблицы с измерениями хранятся в реляционных таблицах, и для хранения агрегатов создаются дополнительные реляционные таблицы.

HOLAP использует реляционные таблицы для хранения базовых данных и многомерные таблицы для агрегатов. Особым случаем ROLAP является R-ROLAP реального времени (Real-time ROLAP – R-ROLAP). В отличие от ROLAP в R-ROLAP для хранения агрегатов не создаются дополнительные реляционные таблицы, а агрегаты рассчитываются в момент запроса. При этом многомерный запрос к OLAP-системе автоматически преобразуется в SQL-запрос к реляционным данным.

Каждый тип OLAP-систем имеет как свои сильные, так и слабые стороны. Если рассматривать классический MOLAP, то сильной стороной является быстродействие. Из минусов можно выделить большой объем хранилища данных, так как данные денормализованы и, следовательно, часть информации дублируется. Так же система требовательна к аппаратной части, так как OLAP-кубы строятся в оперативной памяти сервера. Данный тип систем отлично справляется с небольшими наборами данных. ROLAP-системы, в отличие от MOLAP, характеризуются низким объемом хранимых данных по причине их нормализации. Однако данный тип систем имеет гораздо более низкое быстродействие. Для повышения скорости обработки данных используется кэширование извлеченных агрегатов, но, если для выполнения операции требуются данные, которые ранее не использовались, то происходит стандартный SQL-запрос с объединениями в реляционную базу данных. Так как время выполнения запроса может быть относительно велико, то и само время выполнения аналитической операции также может быть достаточно большим. ROLAP-системы предпочтительней использовать с достаточно большими объемами данных, когда критичным параметром является объем хранилища. HOLAP находится между этими двумя подходами, он достаточно хорошо масштабируется и быстро обрабатывается. Архитектура R-ROLAP позволяет производить многомерный анализ OLTP-данных в режиме реального времени.

Модель системы учета данных

Основные требования к системе учета данных:

– низкие затраты на разработку и конкурентная стоимость для клиента;

– возможность использования различными клиентами одного ядра;

– масштабируемость системы;

– безопасность данных.

При построении классической OLTP (OnlineTransaction-Protocol) системы учета данных на выделенном сервере, мы сталкиваемся с рядом проблем: довольно высока стоимость разработки ПО. Система по умолчанию не является горизонтально масштабируемой. Наращивать вычислительные мощности представляется возможным только устанавливая более мощные аппаратные средства. Остро стоит вопрос безопасности данных. Для полной защиты от потери информации необходим дополнительный сервер, на который будут реплицироваться транзакционные данные, что также увеличивает стоимость конечной системы. К тому же, система должна использоваться различными клиентами с различными схемами базы данных. При этом, база данных должна быть масштабируемой. Решением проблемы

является использование NO-SQL (NotOnlySQL) документоориентированных хранилищ данных, где, как таковое, отсутствует понятие «схема».

Оптимальным выбором является разработка системы учета данных на основе PaaS сервисов. Изначально уменьшаются затраты на разработку, так как сама разработка ведется на уже готовой платформе. Система по умолчанию является масштабируемой. К тому же, PaaS системы в большинстве случаев строятся на основе нереляционных хранилищ данных. Это позволит в рамках одной базы данных работать нескольким клиентам с совершенно разной схемой данных. Решается вопрос с безопасностью данных, так как информация автоматически реплицируются на несколько серверов. В большинстве случаев фактор репликации равняется трем (это значит, что кроме главного сервера информация дублируется еще на два дополнительных). Безопасность облачных систем от вирусных атак также высока. При этом оплата ведется только за использованные ресурсы: дисковое пространство, количество обращений к серверам, количество процессов чтения и записи, процессорное время. Однако системы не лишены минусов. Одним из главных минусов является высокий порог вхождения для начала разработки. Еще одним существенным минусом является невысокая производительность на сложных запросах и отсутствие поддержки классических транзакций. Это связано с тем, что облачное хранилище данных, как правило, является распределенным и, при выполнении сложных запросов, задействуются для обработки сразу несколько нод, на которых хранится информация. При потере связи с какой-либо нодой запрос не будет выполнен, пока не получен ответ со всех нод. По этой причине, время выполнения запроса искусственно ограничивают, чтобы избежать подобных ситуаций. Однако проектируемая система учета является классической OLTP-системой (OnlineTransactionProtocol). Это значит, что данные будут идти небольшими пакетами, но постоянным потоком. Для этой цели облачное хранилище является подходящим выбором. Следует отметить, что при использовании нереляционного хранилища в рамках нашей задачи должны отдельно сохраняться файлы конфигураций баз данных различных клиентов. Эти файлы необходимы как для ядра системы учета, так и для системы предварительной обработки и загрузки данных. На основе этих файлов будет основываться сама схема данных для различных клиентов, и будет происходить визуализация модели данных хранилища (предоставления клиентам графического интерфейса для работы с базой данных). Также файлы будут использоваться системой предварительной обработки: на основе этих файлов конфигурации будут строиться OLAP-кубы данных в аналитическом модуле.

Оптимальным выбором будет хранение файлов конфигураций в xml-формате. Это ведет к следующему требованию: наличие файлового хранилища, интегрированного в PaaS платформу. Также данное хранилище должно быть доступно как внешний сервис, так как впоследствии на основе этих конфигураций будут организовываться OLAP-кубы в аналитическом модуле (рисунок 3).

Модель аналитического модуля

Основные требования к аналитическому модулю:

– низкие затраты на сопровождение системы;

– масштабируемость.

Главное целью OLAP-серверов в системе является анализ данных за фиксированный промежуток времени на основе агрегированных данных. В определенный момент времени наступает ситуация, при которой одного сервера с OLAP недостаточно для проведения анализа данных всех клиентов системы. При этом требуется подключение новых ресурсов (к примеру, поднять еще один OLAP-сервер). Также стоит отметить, что проведение анализа чаще всего проходит по окончанию определенного периода (месяц, квартал), следовательно, в течение всего периода аналитическая система является не востребованной. По этой причине более выгодным является подключение дополнительных OLAP-серверов именно в период максимальной нагрузки.

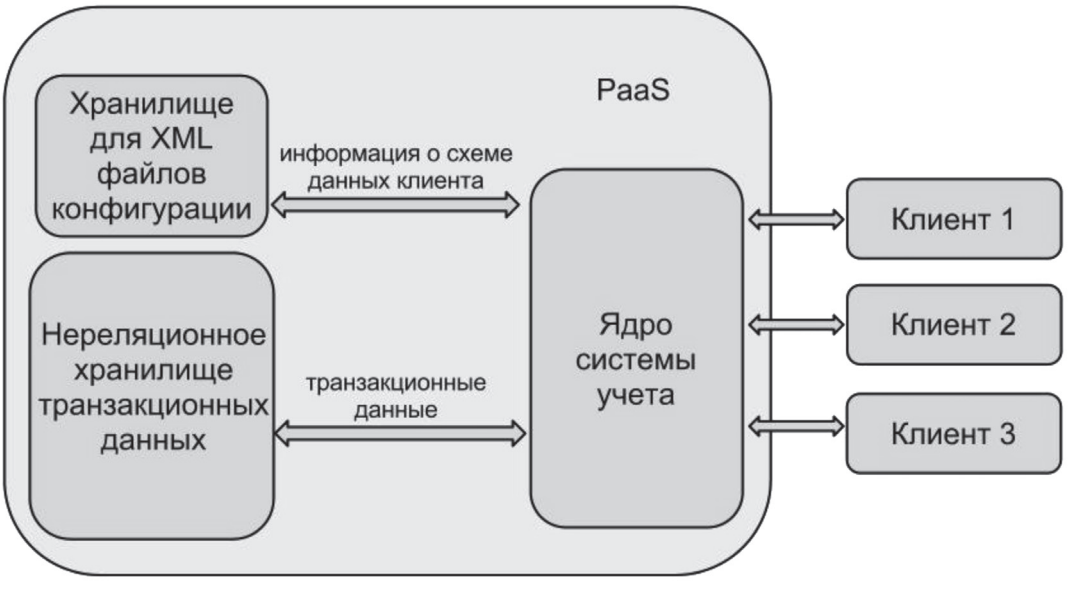


Рисунок 3 – Схема OLTP-системы учета данных

При использовании IaaS инфраструктуры данная операция займет наименьшее время. При этом может быть развернут уже готовый образ с установленной аналитической системой, который сразу же будет готов к использованию.

Что касается системы предварительной обработки и загрузки данных, то оптимальным решением также будет расположение этой системы в PaaS инфраструктуре, вместе с системой учета. Это объясняется несколькими факторами. Во-первых, это быстродействие. Иногда данные, прежде чем отправиться в аналитический модуль, должны быть предварительно обработаны. К тому же, как было описано выше, одним из фундаментальных ограничений облачных хранилищ является невозможность проводить сложные запросы. Следовательно, система предварительной обработки данных должна быть настроена так, чтобы получать данные из хранилища системы учета мелкими порциями, а затем производить их дальнейшую трансформацию и загрузку в аналитический модуль. То есть, при данной архитектуре системы будет производиться множество запросов к хранилищу и имеет смысл всю систему предварительной обработки располагать в PaaS среде. Также необходимо отметить, что с использованием PaaS система предварительной обработки будет масштабируемой и будет оставаться работоспособной при резких повышениях нагрузки на всю систему в целом.

Реальная модель системы учета и анализа данных представлена на рисунке 4.

Модель OLAP-сервера

Так как изначально приложение разрабатывается для предприятий малого и среднего бизнеса, то объем данных относительно невелик. Для данных задач прекрасно подойдет MOLAP система (рисунок 5). Отдельным плюсом будет тот факт, что данные в нереляционном хранилище изначально хранятся в виде агрегатов. Это позволит избежать дополнительной обработки информации при загрузке данных из системы учета в аналитический модуль и снизить нагрузку на сервер, расположенный в IaaS инфраструктуре, и, следовательно, повысить быстродействие и снизить стоимость.

При проектировании системы следует предвидеть ситуацию, когда может резко возрасти нагрузка со стороны какого-либо клиента. Причиной может быть быстрый экономический рост предприятия, слияние с другим предприятием и т.д. В итоге резко возрастает объем данных, которые должны быть подвергнуты анализу. MOLAP системы прекрасно справляются с небольшими базами данных, однако, при резком увеличении информации скорость запросов будет понижаться, а объем данных OLAP-куба резко возрастет. Решением данной проблемы является переход на использование ROLAP аналитической системы (рисунок 6), в которой данные нормализованы и не занимают большого объема. Однако для корректной работы и безопасного перехода с одного типа системы на другой необходимо наличие одинакового общего API (application programming interface) данных систем, чтобы управление происходило через общий интерфейс. Этот факт необходимо учесть при выборе реализаций OLAP-продуктов, а также при разработке системы предварительной обработки и загрузки.

Как было сказано выше, загрузку данных в аналитический модуль производит система предварительной обработки и загрузки данных. Причем, загрузка данных происходит небольшими частями. Это обусловлено невозможностью нереляционного PaaS-хранилища производить сложные запросы. Для

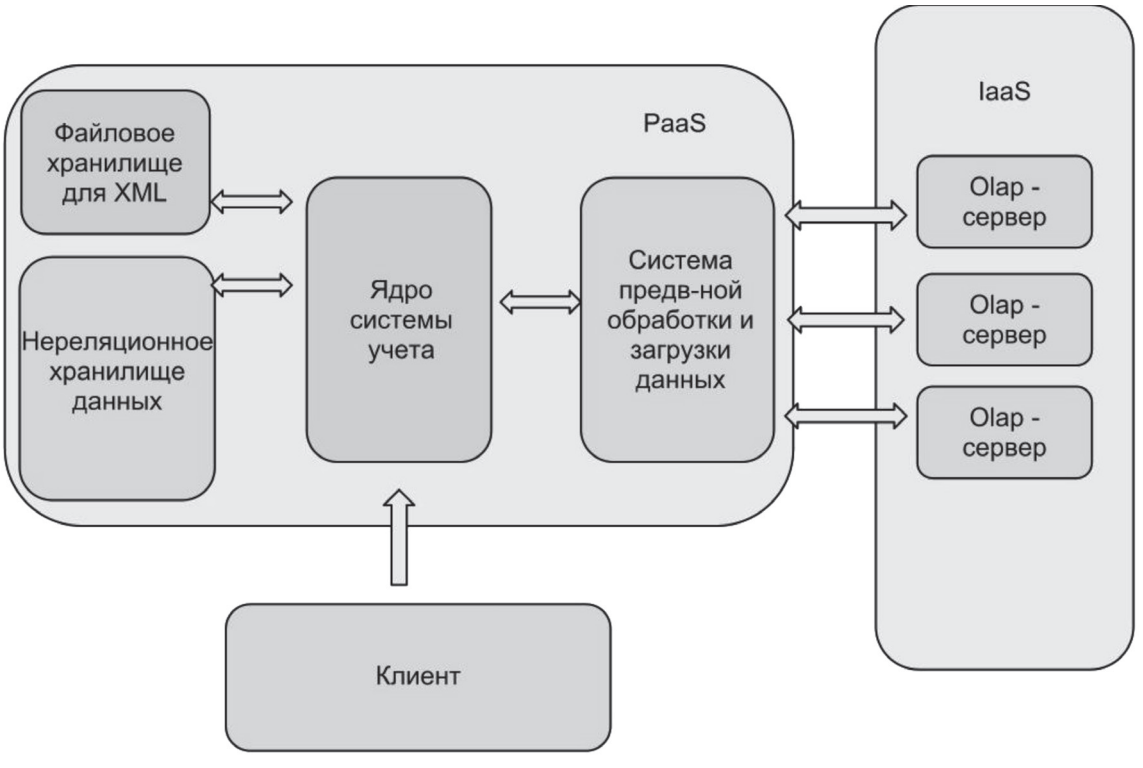


Рисунок 4 – Модель системы аналитической обработки в облачной инфраструктуре

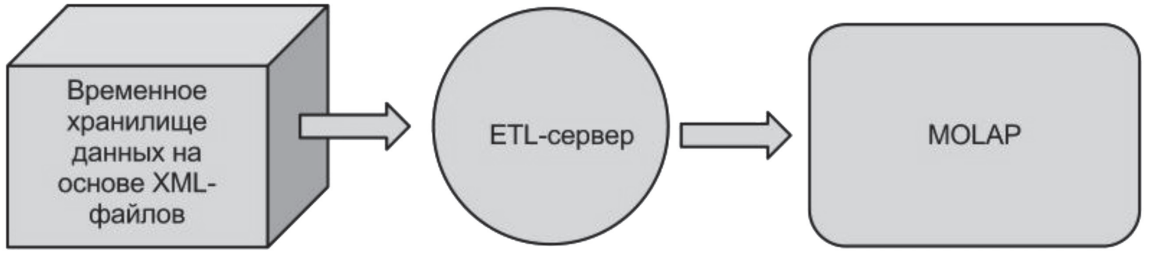


Рисунок 5 – Структура MOLAP-сервера

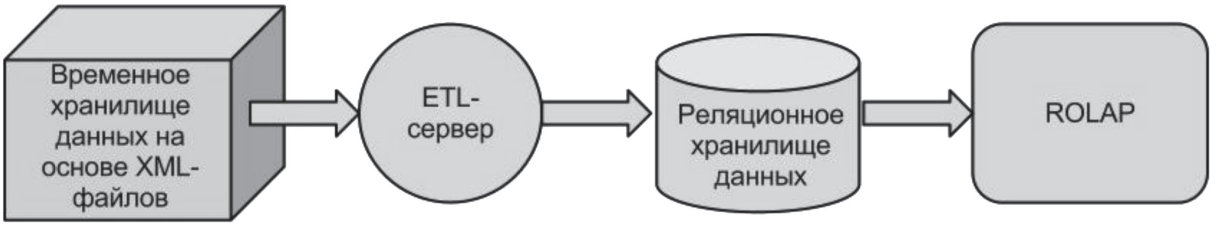


Рисунок 6 – Структура ROLAP-сервера

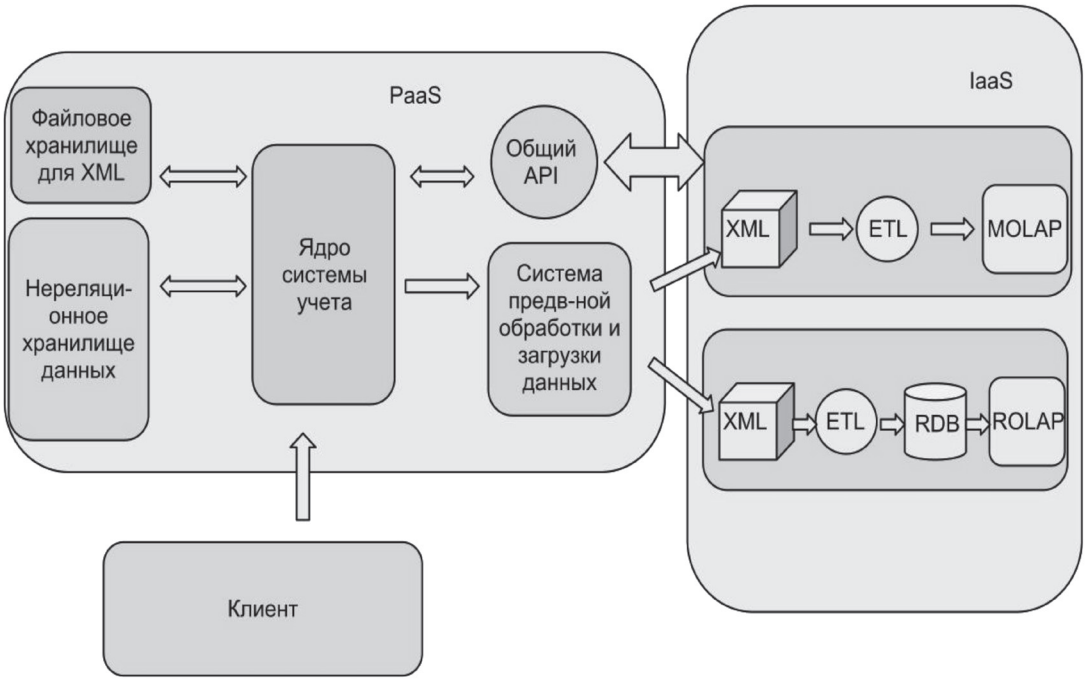


Рисунок 7 – Полная модель системы

корректного сбора данных на стороне OLAP-сервера необходимо дополнительное временное хранилище, из которого будет непосредственно производиться загрузка данных в OLAP-сервер. В качестве временного хранилища планируется использовать xml-файлы. Основные причины, по которым был выбран xml-формат:

1. Данные, поступающие из системы предварительной обработки и загрузки, являются денормализированными и представляют собой агрегаты (самодостаточные логические единицы), что легче всего отобразить в таких форматах, как xml либо json.

2. Данные в виде xml-файлов, в отличие от json, доступны как для последующей загрузки в MOLAP-сервер, так и для преобразования в реляционную модель для использования в ROLAP-системах.

Полная схема системы показана на рисунке 7. Для простоты в IaaS инфраструктуре на схеме указаны только два

Таблица 1 – Основные PaaS платформы и их вендоры

PaaS-сервис	Вендор	дата выхода на рынок	Состояние
Beanstalk	Amazon	2011	в производстве
GoogleAppEngine	Google	2008	в производстве
Force.com	Salesforce.com	2009	в производстве
Azure	Microsoft	2008	в производстве
OpenShift	RedHat	2012	режим тестирование
CloudFoundry	Vmware	2012	режим тестирование

OLAP-сервера. На самом деле их может быть больше и новые серверы должны подниматься по требованию в пиках нагрузки, как было описано выше. Управление серверами осуществляется системой предварительной обработки и загрузки данных через общий API OLAP-серверов.

Безопасность системы

Наиболее частые проблемы безопасности могут проявиться в двух самых выраженных направлениях:

1. Традиционные атаки на ПО. Они связаны с уязвимостью сетевых протоколов, операционных систем. Для защиты от данных типов угроз клиенту необходима установка антивируса и межсетевого экрана.

2. Атаки на клиента. Этот тип атак отработан в вебсреде, но он также актуален и для облака, поскольку клиенты подключаются к облаку, как правило, с помощью браузера. В него попадают такие атаки как Cross Site Scripting (XSS), перехваты веб-сессий, воровство паролей, «человек посредине» и другие. Защитой от этих атак традиционно является строгая аутентификация и использование шифрованного соединения.

Для безопасного взаимодействия системы учета данных с аналитическим модулем необходимо использование шифрованного соединения. Также плюсом будет строить систему на PaaS и IaaS сервисах от одного провайдера. В таком случае данные с большой вероятностью будут находиться в одних и тех же датацентрах (по географическому признаку) и будут передаваться по внутренним скоростным каналам, что минимизирует сбои системы из-за потерь информации при передаче.

Анализ облачных платформ

Тестирование PaaS сервисов является нетривиальной задачей. Каждый сервис представляет собой платформу для разработки, тестирования и разворачивания продукта. При этом, каждый сервис разрабатывался с учетом своих конкретных целей и задач, и различия между облачными платформами очень значительны, что делает их сравнение весьма субъективным, зависящим от конкретной задачи. К тому же, при использовании PaaS платформы, мы не платим за виртуальный сервер (как при использовании IaaS-инфраструктуры или же просто выделенных VPS-серверов), а оплата ведется за используемые ресурсы, такие как объем хранилища, количество операций чтения-записи, процессорное время. Оптимальной стратегией выбора облачной платформы будет проверка на соответствие основным требованиям разрабатываемой системы учета транзакционных данных и системы предварительной обработки и загрузки данных.

Среди основных требований, предъявляемых к PaaS платформе, можно выделить:

– наличие нереляционного хранилища данных;

– интеграция с файловым хранилищем;

– репликация данных;

– защищенный доступ по SSL.

Плюсом также будут являться следующие факторы:

– поддержка Map-Reduce парадигмы распределенных вычислений для нереляционного хранилища;

– наличие реляционного хранилища данных для хранения дополнительной информации о ресурсах, потребляемых клиентом;

– версионирование.

Особое внимание необходимо обратить на языки и технологии программирования, на основе которых будет вестись разработка системы и определить порог вхождения в технологию в целом. Следует отметить, что иногда технология может поддерживаться не полностью. Примером такой ситуации может являться случай, когда платформа поддерживает определенный язык программирования, но при этом не поддерживает весь набор библиотек. В таблице 1 показаны основные PaaS платформы, и их вендоры.

В таблице 2 описаны основные характеристики PaaS-платформ. На основе таблиц 1 и 2 можно сделать выводы, что платформы Force.com, OpenShift и CloudFoundary не подходят к нашей задаче. Первая платформа является несовместимой с задачей из-за отсутствия нереляционного хранилища данных, файлового хранилища, а также из-за языка программирования Apex, который не используется нигде кроме самой Force.com платформы. Что касается OpenShift и CloudFoundary, то, хоть данные платформы и могут работать с нереляционными хранилищами на основе СУБД MongoDB, они, на данный момент, находятся в режиме тестирования.

Оставшиеся платформы имеют сходный набор сервисов, они все поддерживают парадигму параллельных

вычислений MapRedu северсионирование и защищенный доступ. Поддержка Map-Reduce вычислений на данный момент не используется, однако, существует вероятность, что при внесении дополнительного функционала в систему, данный тип вычислений станет востребованным. Что касается стоимости услуг, на момент написания цены на услуги от трех вышеперечисленных провайдеров были примерно одинаковыми. Это значит, что цена не будет играть существенной роли при выборе платформы. После более детального рассмотрения платформы от Google, в ней обнаружилось одно фундаментальное ограничение: язык Java поддерживается не полностью, есть только ряд библиотек, которые могут быть использованы. Это существенно снижает гибкость разработки и делает данную платформу непригодной для использования в рамках нашей задачи.

Сравнивая платформы от Windows и Amazon, можем наблюдать примерно одинаковый функционал. Но, на данный момент, Amazon является мировым лидером в предоставлении IaaS инфраструктуры благодаря своей системе AmazonEC2. Это будет учитываться при выборе инфраструктуры, в рамках которой будет выбрано расположение аналитического модуля. Что касается хранилища, то AmazonDyanomoDB является классическим документоориентированным хранилищем данных, что также более приемлемо, чем решение от Microsoft. К тому же, пробный период при разработке на платформе Amazon составляет год. Это также является плюсом. Таким образом, можно сказать, что платформой для разработки будет являться AmazonBeanstalk.

Анализ OLAP-серверов

Основными параметрами при выборе OLAP-системы будут:

– лицензия;

– тип системы;

– API и язык запроса.

Важным критерием при разработке системы является тип лицензии. Так как основными потребителями продукта будут являться предприятия малого и среднего бизнеса, то стоимость системы должна быть минимальной. По этой причине будут рассматриваться только системы с открытой лицензией. Еще одним параметром является тип системы. Как было сказано ранее, основная обработка данных должна проводиться в MOLAP системах, однако, при повышениях нагрузки должна быть реализована система перехода на ROLAP системы. По этой же причине, для корректного перехода между системами важным параметром является общий API аналитических систем. Рассмотрим основные OLAP-системы, которые присутствуют на рынке в данный момент.

Исходя из данных таблицы 3, можно сделать вывод, что для аналитического модуля подходят такие OLAP-серверы, как Mondrian OLAP server от Pentaho и Palo от компании Jedox. Оба эти серверы распространяются по открытой лицензии и имеют общий APIolap4j. К тому же, эти сервера имеют разный тип: Palo – это классический MOLAP, а Mondrian представляет собой реляционный ROLAP. Использование библиотеки olap4j накладывает следующее ограничение: основным языком разработки должен быть Java.

Архитектура системы с учетом анализа решений на рынке

Конечная архитектура разрабатываемой системы представлена на рисунке 8.

В качестве PaaS платформы будет использоваться AmazonBeansTalk. Транзакционные данные хранятся в нереляционном документоориентированном хранилище dynamoDB. Файлы конфигурации схем клиентов хранятся в файловом хранилище AmazonS3. Загрузка данных из OLTP системы учета в аналитический модуль происходит через систему предварительной обработки и загрузки

Таблица 2 – Основные характеристики PaaS-платформ

PaaS-сервис	Нереляционное хранилище данных	Реляционное хранилище данных	Файловое хранилище	языки и технологии разработки
Beanstalk	DynamoDB	Relational Database Service (MySQL, ORACLE, SQL Server)	Amazon S3	PHP, Python, Ruby, .NET, Java
GoogleAppEngine	AppEngineDatastore	Cloud SQL (MySQL)	BlobStore	Java, Python
Force.com	отсутствует	Database.com (OracleDatabase)	отсутсвует	Apex (Java-подобный язык собственного производства)
Azure	AzureStorage	SQL Database	BLOB Storage	Python, Java, Node. JS, .NET
OpenShift	MongoDB	MySQL, PostreSQL	отсутсвует	PHP, Ruby, Node.JS, Java
CloudFoundry	MongoDB	MySQL, PostreSQL	отсутсвует	Ruby, Node.JS, Java, Scala

Таблица 3 – Основные параметры OLAP-серверов

OLAP сервер	Вендор	Лицензия ПО	Модель данных	API и язык запроса
Essbase	Oracle	Проприетарное ПО	MOLAP, ROLAP, HOLAP	XMLA, OLEDB, MDX
icCube OLAP Server	MISConsulting SA	Проприетарное ПО	MOLAP	XMLA, OLEDB, MDX
MicrosoftAnalysisServices	Microsoft	Проприетарное ПО	MOLAP, ROLAP, HOLAP	XMLA, OLEDB, MDX, olap4j
Microstrategy OLAP Services	Microstrategy	Проприетарное ПО	MOLAP, ROLAP	нет данных
Mondrian OLAP server	Pentaho	EPL	ROLAP	XMLA, OLEDB, MDX
Oracle OLAP Option	Oracle	Проприетарное ПО	MOLAP, ROLAP, HOLAP	OLEDB, MDX, olap4j
Palo	Jedox	GPL v2 or EULA	MOLAP	XMLA, OLEDB, MDX, olap4j
SAS OLAP Server	SAS Institute	Проприетарное ПО	MOLAP, ROLAP, HOLAP	XMLA, OLEDB, MDX
TM1	IBM	Проприетарное ПО	MOLAP	XMLA, OLEDB, MDX

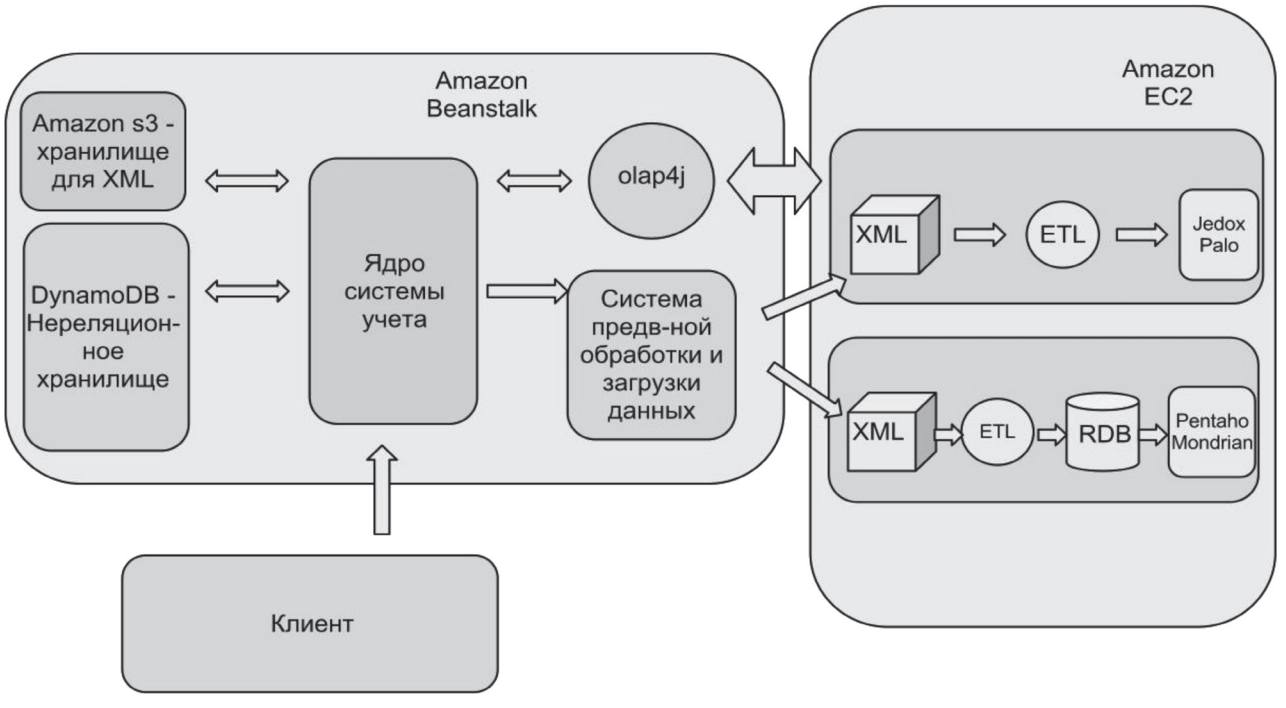


Рисунок 8 – Архитектура системы с учетом анализа решений на рынке

данных. В качестве аналитических серверов используются MOLAP решение Palo и ROLAPMondrian. OLAP-сервера располагаются в IaaS инфраструктуре AmazonEC2, а загрузка данных в хранилище самой OLAP-системы происходит из предварительного хранилища на основе xml-файлов. Связь между системой предварительной обработки и загрузки данных с OLAP-серверами происходит через общий интерфейс на основе olap4jAPI.

ОТОПЛЕНИЕ ПОМЕЩЕНИЙ ИНФРАКРАСНЫМИ ЭЛЕКТРООБОГРЕВАТЕЛЯМИ НОМАКОН™

А.Е. Рабко, И.Л. Козловский, В.А. Орсич

Инфракрасные (ИК) электрообогреватели серии НОМАКОН™ ЭИУС-100, ЭИУС-200 и ЭИУС-300 рекомендуются для использования в качестве источника направленного лучистого тепла при обогреве промышленных помещений и рабочих мест, в том числе, и на открытом воздухе [1]. ИК-электрообогреватели идеально подходят для обогрева не отапливаемых помещений типа мастерских, гаражей, станций СТО, небольших цехов и складов, где необходим постоянный или периодический экономичный обогрев, не требующий значительных капиталовложений. Особенно они эффективны для локального или местного обогрева рабочих мест в больших цехах, которые отапливать целиком не рентабельно.

Наряду с общеизвестными преимуществами ИК-обогревателей, определяющими их эффективность, такими, как минимальная инерционность разогрева и отсутствие выраженной конвекции воздуха с выносом тепла из зоны обогрева, электрообогреватели ЭИУС с керамическими излучателями работают в «темной» области спектра ИК-излучения, не генерируют коротковолнового светового излучения по сравнению с кварцевыми, карбоновыми и галогенными излучателями. При этом они абсолютно безвредны для здоровья человека: керамические излучатели НОМАКОН™ серии ИКН-100 и ИКН-200 сертифицированы для использования в медицинских учреждениях для обогрева новорожденных младенцев [2].

При включении ИК-электрообогревателей ЭИУС требуется всего несколько минут для достижения комфортной температуры, при которой работник может качественно выполнять свои функции. Когда человек покидает рабочее место на длительное время (более 30 мин) обогрев может быть выключен.

Конструкция электрообогревателей позволяет крепить их на стены и потолки (серии ЭИУС-100 и ЭИУС-200), а также устанавливать на пол (серия ЭИУС-300) (рисунок 1). Расстояние до нагреваемого объекта зависит от типа обогревателя, его мощности и способа установки. Ориентировочно, на 10 м² отапливаемой площади требуется ИК-электрообогреватель мощностью 0,65–1,2 кВт.

Целью авторов настоящей статьи является разработка метода расчета систем локального лучистого обогрева и методики применения ИК-электрообогревателей ЭИУС, соответствующих требованиям существующих нормативных документов, регламентирующих показатели микроклимата производственных помещений, оборудованных системами лучистого отопления (СЛО) [3–5].

Особенности микроклимата при воздействии СЛО заключаются в том, что человек чувствует себя хорошо, если выделение тепла его телом и потери тепла телом в окружающее пространство находятся в равновесии. Тепловой баланс нарушается, если при низких температурах окружающей среды количество отдаваемого тепла становится больше количества тепла, образующегося в человеческом теле. Возникающая при этом разность количеств тепла, которая, например, при температуре окружающего воздуха 10°С может достигать 150 Вт/м², должна быть сообщена человеку направленным лучистым отоплением.

В стандартных тестах [6] для оценки теплопотерь ИК-излучением принимают температуру поверхности тела (кожи) T_{ip1} = 32 °С и степень черноты кожи человека ε_{ip1} = 0,95. Если принять температуру окружающих стен и потолка (источники фонового ИК-излучения на человека) в пределах T_{ip2} = 0–16 °С при степени черноты ε_{ip2} = 0,92 (штукатурка, кирпичная кладка, обои), то расчетные тепло-

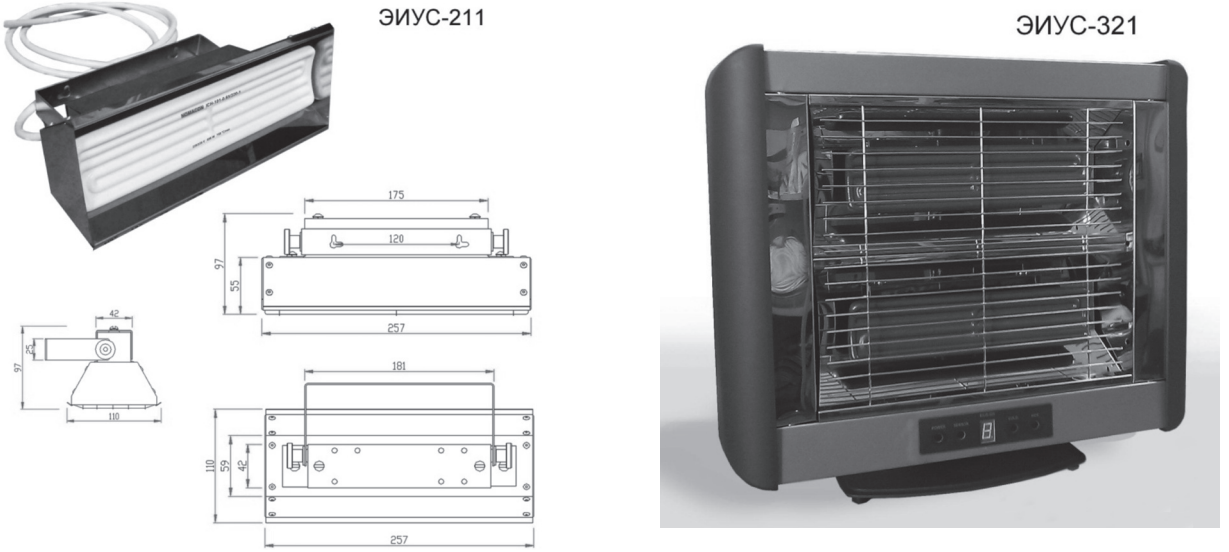


Рисунок 1

потери ИК-излучением голого тела (как результирующая истинного излучения тела и излучения фоновых источников) составит, Вт/м²:

$$E_{ip} = \varepsilon_{12} \cdot C_0 \cdot \left[\left(\frac{T_{ip1}}{100} \right)^4 - \left(\frac{T_{ip2}}{100} \right)^4 \right], \quad (1)$$

где $\varepsilon_{ip12} = \frac{1}{\frac{1}{\varepsilon_{ip1}} + \frac{1}{\varepsilon_{ip2}} - 1}$ – приведенная степень черноты кожа человека – фон, C_0 = 5,671 Вт/м²К⁴ – постоянная уравнения Стефана-Больцмана, T_{ip1} и T_{ip2} выражены в градусах Кельвина.

С учетом особенностей облучения одетого человека на основании сопоставления энергии излучения тела по формуле (1) и предельного теплового облучения из нормативных документов, приведенных в таблице 1, возможно принять граничные значения удельных мощностей теплового облучения для диапазона эффективного обогрева от E_{1max} до E_{1min} при различной температуре окружающей среды:

$$E_{1max} = 0{,}7418 \cdot E_{ip} - 16{,}1 \quad (2)$$
$$E_{1min} = 0{,}2124 \cdot E_{ip} - 2{,}96 \quad (3)$$

Таким образом, в зависимости от температуры окружающей среды, электрической мощности и конструкции обогревателя, а также от расстояния по нормали от поверхности излучения до объекта обогрева, возможно выделить три зоны лучистого обогрева в помещении, оборудованном СЛО: зону повышенной интенсивности теплового облучения при $E_1 > E_{1max}$, зону эффективного обогрева при $E_{1min} \le E_1 \le E_{1max}$, а также зону комфортного, но недостаточного обогрева при $E_1 \le E_{1min}$.

Следует отметить, что приведенное выше деление на зоны (рисунок 2) необходимо для последующих расчетов при первом включении ИК-электрообогревателя, а также при выходе СЛО на заданный режим поддержания температуры в помещении. С ростом температуры в помещении границы зон будут смещаться и важно расположить объекты обогрева (работников, оборудование и т.п.) в помещении таким образом, чтобы они постоянно находились в зоне эффективного обогрева, или в зоне комфортного обогрева недалеко от границы раздела зон.

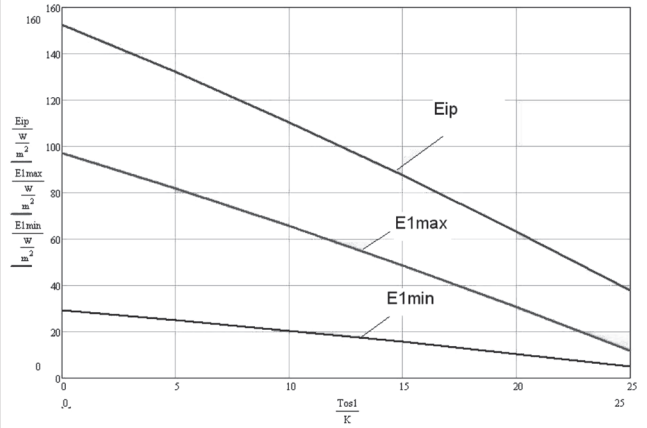


Рисунок 2 – Зависимость граничных значений интенсивности лучистого обогрева от температуры

Для расчета распределения зон обогрева при применении СЛО на базе ИК-электрообогревателей НОМАКОН™ ЭИУС-211 воспользуемся данными таблицы 2. Принимаем начальную температуру в помещении T_{ip2} равной 10 °С, температуру термостабилизации T_{ip2} равной 18 °С и по формуле (1) рассчитываем значения предельной интенсивности теплового облучения E_{ip1} = 111,4 Вт/м², E_{ip2} = 73,8 Вт/м². По формулам (2) и (3) рассчитываем граничные значения зон облучения: E_{1max} = 66,5 Вт/м², E_{1min} = 20,7 Вт/м² для периода начального разогрева, E_{2max} = 38,6 Вт/м², E_{2min} = 12,7 Вт/м² для периода термической стабилизации.

Таблица 1 – Показатели микроклимата производственных помещений, оборудованных СЛО

Температура воздуха, окружающей среды °С	СанПиН 9-80 РБ 98 [3] ГОСТ 12.1.005-88 [5] Интенсивность теплового облучения, Вт/м², не более, при облучаемой поверхности тела 50 % и более	СП 2.2.1.1312-03 [4]		ISO 7726:1998 [6], формула (1), Вт/м²
		Интенсивность теплового облучения головы, Вт/м², не более	Интенсивность теплового облучения туловища, Вт/м², не более	
0	-	-	-	153
5	-	-	-	132
10	-	-	-	111
11	-	60	150	106
12	35	60	125	101
13		60	100	97
14		45	75	92
15		30	50	87
16		15	25	83
20		-	-	63
25	-	-	-	38

Данные нормативных документов приведены на период 8-часовой рабочей смены применительно к человеку, одетому в комплект одежды с теплоизоляцией (термическим сопротивлением) R_0 =0,155 м²К/Вт (спецодежда от общих загрязнений) и выполняющему работу средней тяжести (категория IIа–IIб).

Таблица 2 – Технические характеристики ИК-электрообогревателей ЭИУС-211

Наименование показателя	**Электрическая мощность излучателя ИКН-101, N_{e1}, Вт**				
	250	**400**	**500**	**650**	**1000**
Размеры излучателя в плане, расчетная площадь поверхности излучения и степень черноты поверхности излучения	245х60 мм, F_1 = 175 см², ε_1 = 0,96				
Размеры рефлектора-отражателя в плане, приведенные размеры центральной элементарной поверхности излучения и приведенный радиус поверхности излучения	L_{ot1} = 250 мм, B_{ot1} = 100 мм L_1 = 100 мм, B_1 = 100 мм, $R_1 = \sqrt{\frac{L_1 \cdot B_1}{\pi}}$ = 56,42 мм				
Принятый объемный угол раскрытия ИК-лучей на выходе из отражателя	α_1 = 68 - 72°				
Температура излучающей поверхности, T_1, °С	400	490	550	610	720
Коэффициент эмиссии электрической энергии в энергию ИК-излучения η_1, % (лучевой к.п.д.) при степени черноты фона ε_2 = 0,92 и температуре окружающей среды T_2 = 10–25 °С (осредненный)	69,7	73,0	79,5	81,3	84,9

Расчет распределения удельной мощности облучения по нормали к излучающей поверхности электрообогревателя E_{n1}, Вт/м² производим по зависимости

$$E_{n1} = E_{ot1} \cdot \frac{R_1^2}{\left[R_1 + L_{n1} \cdot \tan\left(\frac{\alpha_1}{2}\right)\right]^2}, \qquad \textbf{(4)}$$

где $E_{ot1} = \frac{N_{e1} \cdot \frac{\eta_1}{100}}{L_{ot1} \cdot B_{ot1}}$ – удельная мощность излучения, приведенная к площади рефлектора-отражателя, Вт/м², L_{n1} – текущее значение расстояния от поверхности излучения, м. В вычислениях по формуле (4) принято значение объемного угла раскрытия ИК-лучей в вертикальной плоскости на выходе из отражателя α_1= 70°.

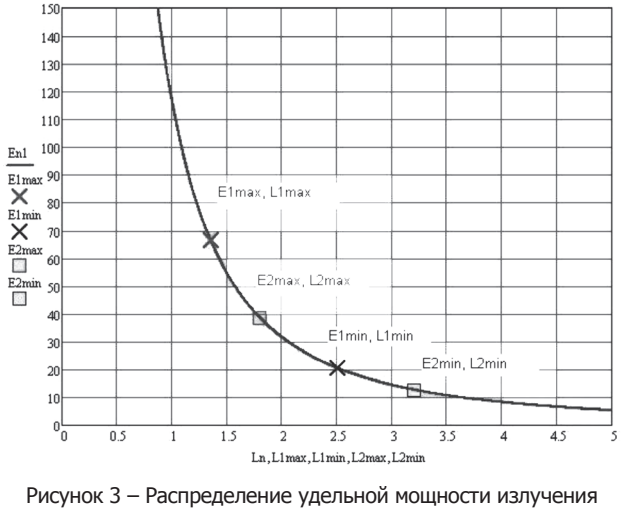


Рисунок 3 – Распределение удельной мощности излучения по длине

На рисунке 3 представлен полученный график распределения удельной мощности излучения по нормали к излучающей поверхности. Точками отмечены полученные граничные расстояния зон обогрева: L_{1max} = 1,35 м, L_{1min} = 2,5 м для периода начального разогрева, L_{2max} = 1,8 м, L_{2min} = 3,2 м для периода термической стабилизации.

На рисунке 4 с соблюдением масштаба представлены диаграммы обогрева помещения длиной 4 м с высотой потолков 2,5 м, которое оборудовано СЛО в виде ЭИУС-211, размещенного на стене на высоте 1,8 м с направлением рефлектора-отражателя по диагонали помещения под углом 30 ° вниз по отношению к горизонтали. Выделенное объемное расположение зон обогрева на диаграммах с возможностью изменения расположения электро-обогревателя и направления лучевого потока позволяет найти оптимальный вариант размещения СЛО и объектов обогрева.

Следует отметить, что приведенные выше расчеты и на их основе зонирование помещения по режимам обогрева справедливы при лучистом облучении 50 % и более тела человека. Оптимальное построение СЛО в данном случае определяет наличие двух и более обогревателей, расположенных диаметрально во всем помещении или в выделенной зоне обогрева. С целью снижения размеров зоны повышенной интенсивности облучения при достижении термической стабилизации электрообогреватель должен быть снабжен ступенчатым переключателем мощности обогрева.

Данная методика позволяет оценить интенсивность облучения, например, как на уровне пола, так и на уровне головы человека по мере приближения к обогревателю. Очевидно, что установка ИК-обогревателей на потолке с подачей излучения сверху в низ для помещений с потолками высотой 2,5–4 м будет создавать интенсивность облучения головы всегда выше, чем остального тела. При этом голова будет находиться в зоне повышенной интенсивности облучения, что не позволяет выдержать действующие нормативы облучения и резко повышает непереносимость лучистого обогрева.

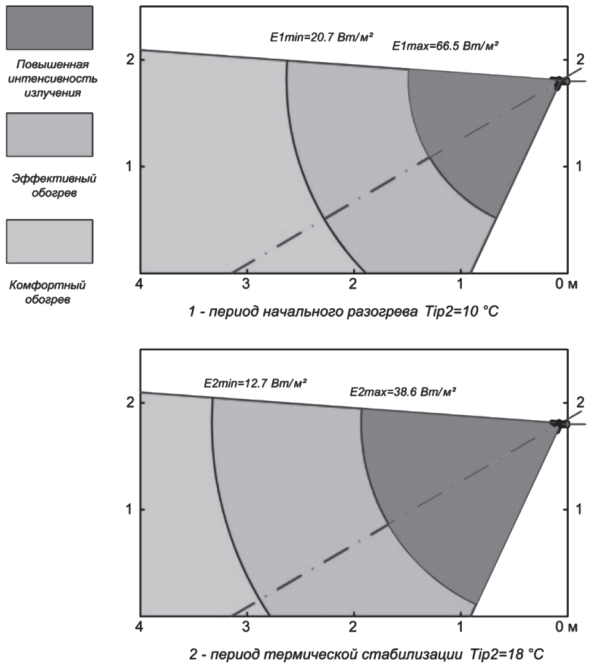


Рисунок 4 – Диаграммы инфракрасного обогрева помещения электрообогревателем ЭИУС-211 650 Вт

Литература:

1. Рабко, А.Е. Инфракрасные керамические излучатели и электрообогреватели НОМАКОН™/ А.Е. Рабко и др. // Электроника инфо. – 2011. – №5. – С. 26–29.

2. Демченко, А.И. Многофункциональная ИК-система обеспечения термонетрального окружения для новорожденных детей / Демченко А.И. и др. // Сб. трудов 22-й Международной конференции «СВЧ-техника и телекоммуникационные технологии». Севастополь. – 2012. – С. 989–990.

3. СанПиН 9-80 РБ 98. Гигиенические требования к микроклимату производственных помещений.

4. СП 2.2.1.1312-03. Гигиенические требования к проектированию вновь строящихся и реконструируемых промышленных предприятий. Приложение 2.

5. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

6. ISO 7726:1998. Ergonomics of the thermal environment. Instruments for measuring physical quantities.

MIKROTIK НАСТРОЙКА И ОБЗОР ТЕХНОЛОГИЙ PPTP, L2TP, IPSEC, PPPOE, IP2IP, EOIP, 802.1Q VLAN

А. Кузьмицкий

Вступление

С ростом потребностей компании в развитии возникает необходимость открывать удаленные филиалы или подразделения, которые зачастую могут находиться в другой части страны или мира. При этом потребность в едином информационном пространстве для них крайне важна и актуальна, о чем говорит значительно возросший интерес к оборудованию и программному обеспечению, реализующему такие возможности.

Так как, зачастую, связь между филиалами не подразумевает использования больших скоростей, то основными факторами при выборе оборудования являются надежность, стоимость владения, возможности.

Многие уже не раз успели убедиться, что программная платформа Mikrotik за невысокой ценой и крайне скромными размерами таит в себе все вышеописанные характеристики. Список поддерживаемых ею технологий не оставит равнодушным ни одного администратора, которому когда-либо придется столкнуться с описываемой нами проблемой.

Возможности

Ниже представлен список возможностей, которые предлагает RouterOS Mikrotik для построения корпоративных сетей:

– поддержка PPTP;
– поддержка L2TP;
– поддержка IPSec;
– поддержка PPPOE;
– поддержка IP2IP.
– поддержка EoIP;
– поддержка 802.1Q VLAN.

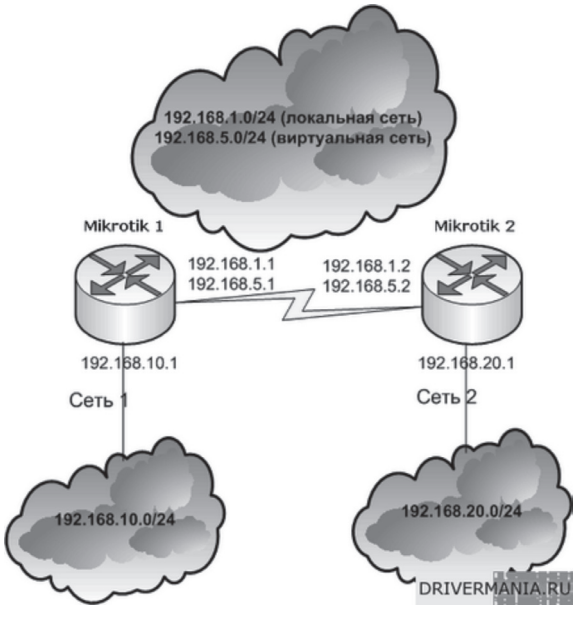


Рисунок 1

Использование в качестве сокрытия информации протоколов, инкапсулирующих пакеты верхних уровней с последующим шифрованием содержимого, позволяет добиться высокой криптоустойчивости и надежности. Даже если злоумышленники перехватят часть зашифрованного трафика, им придется потратить слишком много времени для его расшифровки и весьма вероятно, что им этого не удастся. Таким образом, использование вышеуказанных технологий и применение их комбинаций дает крайне высокий уровень шифрования и защиты передаваемой информации, о чем мы с вами поговорим ниже.

Тестовый стенд

Тренироваться мы с вами будем на следующей схеме (рисунок 1):

Mikrotik 1 и Mikrotik 2 расположены в разных сетях и являются граничными маршрутизаторами. Схема предусматривает, что они имеют внешние IP-адреса или имеют любой другой способ подключения друг к другу. Сеть 192.168.5.0, адреса 192.168.5.1 и 192.168.5.2 являются виртуальными, т.е. созданными в результате поднятия туннеля между маршрутизаторами.

Обращайтесь к этой схеме во время просмотра каждой главы, в этом случае вы легко поймете весь материал.

Описание технологии PPTP

PPTP (Point to Point Tunnel Protocol) переводится как «туннельный межточечный протокол». PPTP – достаточно распространенная технология, которая применяется для создания частных сетей поверх открытых. Высокая производительность, достаточные опции шифрования и аутентификации, реализация на большинстве сетевых программных платформ сделали его одним из самых популярных на рынке. Протокол PPTP обычно используется в следующих случаях:

– создание безопасных туннелей между маршрутизаторами через Интернет;
– объединение локальных сетей поверх открытых;
– создание корпоративных сетей связи с возможностью доступа в локальную сеть предприятия с удаленных компьютеров или мобильных устройств.

Реализация PPTP в Mikrotik позволяет выбрать следующие способы авторизации:

– mschap2;
– mschap1;
– chap;
– pap.

Стоит отметить, что на практике чаще всего используется mschap2, который более безопасен, чем существующие аналоги.

Рассказывая о PPTP стоит упомянуть о протоколе EoIP (EthernetOverIP), который очень часто встречается при создании корпоративных сетей и обычно используется поверх уже созданного виртуального туннеля. EoIP позволяет создать прозрачную сетевую среду, эмулирующую прямое

Ethernet-подключение между сетями. Использование сказанного средства лишает администраторов головной боли по поводу видимости или не видимости объединенных сетей в «Сетевом окружении» и проблем пробрасывания широковещательного трафика в удаленные подсети.

Рассмотрим пример создания шифрованного PPTP туннеля между двумя территориально удаленными офисами, которые используют RouterOS Mikrotik в качестве маршрутизаторов.

На одном из маршрутизаторов необходимо включить PPTP Server: /interface pptp-server server set enabled=yes

Сейчас создадим на этом сервере профиль для нового подключения и новый аккаунт:

/ppp profile add name=filial only-one=yes use-compression=yes use-encryption=yes use-vj-compression=yes

/ppp secret add name=newuser password=newpassword local-address=192.168.5.1 profile=filial remote-address=192.168.5.2 service=pptp

Для правильной идентификации подключившегося клиента целесообразно создать для него «собственный PPTP сервер»: /interface pptp-server add name=filial user=newuser

Сейчас на втором маршрутизаторе добавляем новый интерфейс для подключения к нашему второму маршрутизатору:

/interface pptp-client add name=filial_connection connect-to=192.168.1.1 user=newuser password=newpassword allow=mschap2 disabled=no

Далее, если вам необходим обычный туннель или вы хотите самостоятельно прописать нужные маршруты, в вашем распоряжении весь необходимый инструментарий. В самом минимальном случае вам необходимо прописать на клиентах в обеих сетях шлюзом поумолчанию внутренние интерфейсы маршрутизаторов, а на самих маршрутизаторах указать на каких интерфейсах находятся нужные сети.

К примеру, на первом маршрутизаторе выполним:

/ip route add dst-address=192.168.20.0/24 gateway=192.168.5.1 pref-src=192.168.5.2

А на втором:

/ip route add dst-address=192.168.10.0/24 gateway=192.168.5.2 pref-src=192.168.5.1

В результате мы получим возможность доступа из одной сети в другую, пользуясь маршрутизацией пакетов L3.

Сейчас вы увидите, как можно сделать то же самое, но на втором уровне OSI, применив EoIP и создав прозрачный мост между сетями.

Предположим, что у нас маршрутизаторы удачно создали PPTP туннель и мы хотим сэмулировать работу обычного моста, не трогая маршрутизацию третьего уровня. Для этого создадим EoIP туннели на обоих маршрутизаторах.

На первом:

/interface eoip add name=filial_EoIP remote-address=192.168.5.1 disabled=no

На втором:

/interface eoip add name=filial_EoIP remote-address=192.168.5.2 disabled=no

Теперь необходимо создать мост между внутренним интерфейсом и EoIP на каждом маршрутизаторе.

На первом выполним:

/interface bridge add

/interface bridge port add bridge=bridge1 interface=ether1

/interface bridge port add bridge=bridge1 interface=filial_EoIP

И на втором:

/interface bridge add

/interface bridge port add bridge=bridge1 interface=ether1

/interface bridge port add bridge=bridge1 interface=filial_EoIP

В описываемом случае мы создали прозрачный туннель между двумя сетями с разными диапазонами адресов, поэтому нужно или расширить сетевую маску у всех адресов, или настроить маршрутизацию пакетов.

Описание технологии L2TP

Протокол L2TP похож на PPTP, однако, обладает рядом важных преимуществ. В частности, L2TP туннели более устойчивы к сбоям и предлагают высокий уровень защищенности передаваемых данных в сочетании с IPSec. Обычно L2TP используется в следующих случаях:

– создание защищенных туннелей между маршрутизаторами через открытые сети;

– объединение локальных сетей поверх открытых;

– создание гибких схем аутентификации;

– доступ в корпоративную сеть с удаленных компьютеров.

Как и в случае с PPTP, L2TP подразумевает использование клиент-серверной схемы. Реализация протокола L2TP доступна в большинстве операционных систем, однако, его распространенность несколько ниже других подобных протоколов. В основном, это связано с некоторыми различиями в понимании принципов его работы производителями и не всегда качественному взаимодействию разных систем. В случае с одинаковыми системами таких проблем не возникнет.

Рассмотрим пример использования протокола L2TP на практике. На одном из маршрутизаторов необходимо включить L2TP Server:

/interface l2tp-server server set enabled=yes

Далее создадим на этом сервере профиль для нового подключения и новый аккаунт:

/ppp profile add name=filial only-one=yes use-compression=yes use-encryption=yes use-vj-compression=yes

/ppp secret add name=newuser password=newpassword local-address=192.168.5.1 remote-address=192.168.5.2 service=l2tp profile=filial

Для правильной идентификации подключившегося клиента целесообразно создать для него «собственный PPTP сервер»:

/interface l2tp-server add name=filial user=newuser

На втором маршрутизаторе добавляем новый интерфейс для подключения к нашему второму маршрутизатору:

/interface l2tp-client add name=filial_connection connect-to=192.168.1.1 user=newuser password=newpassword allow=mschap2 disabled=no

Описание IP2IP

Протокол IP2IP является самым простым из всех рассматриваемых нами. Принцип его работы основывается на инкапсуляции IP-пакетов в IP-пакеты. На практике это означает, что нужные нам данные в виде IP-пакетов будут передаваться по сети, упакованными в блоки данных DATA передающихся пакетов. Таким образом, достигается некоторое сокрытие информации без ее шифрования путем создания подключений точка-точка и инкапсуляции пакетов. На практике IP2IP чаще всего используется для создания туннелей между роутерами через сеть Интернет и в целях обмена информацией между маршрутизаторами.

Использование IP2IP в чистом виде не рекомендуется, если присутствует какая-либо возможность перехвата данных, поэтому чаще всего данный туннельный протокол работает как основа для IPSec.

Во многих системах наподобие Cisco IOS и некоторых других присутствуют средства для работы с данной технологией. Приведем пример из практики, позволяющий увидеть принципы создания подключений точка-точка с помощью протокола IPIP. Создание IPIP туннеля состоит из двух частей: создание самого подключения и назначения ему IP-адреса.

Первый маршрутизатор:

/interface ipip add name=tunnel1 local-address=192.168.1.1 remote-address=192.168.1.2 disabled=no

/ip address add address=192.168.5.1/24 interface=tunnel1 disabled=no

Второй маршрутизатор:

/interface ipip add name=tunnel1 local-address=192.168.1.2 remote-address=192.168.1.1 disabled=no

/ip address add address=192.168.5.2/24 interface=tunnel1 disabled=no

Поверх этого туннеля можно также поднять EoIP точно так же, как было сказано выше.

Описание PPPOE

Протокол PPPOE является частным случаем протокола PPP и является одной из самых распространенных туннельных технологий. Его распространению обязаны провайдеры, которые достаточно часто предлагают услуги широкополосного доступа в Интернет, применяя при этом PPPOE. Выбор этой технологии обусловлен ее высокой стабильностью, доступностью, масштабируемостью и отсутствием необходимости назначения IP-адресов конечным устройствам для создания PPPOE туннеля. Проще говоря, протокол PPPOE разрабатывался так, чтобы позволить связать удаленные точки через различные гетерогенные среды, сохраняя при этом доступность, производительность и стабильность.

Использование PPPOE вместо ресурсоемкого PPTP позволяет значительно снизить нагрузку на сервер, однако, если используемые скорости невелики и это не core маршрутизатор крупной организации, то особой разницы вы не заметите. Реализация PPPOE на Mikrotik позволяет воспользоваться следующими типами шифрования трафика:

– No encryption;
– MPPE 40bit RSA;
– MPPE 128bit RSA.

Перед нами стоит задача шифровать трафик, передаваемый через беспроводную среду. В интерфейс маршрутизатора включен беспроводной мост, который обеспечивает прозрачный канал связи с другим беспроводным мостом. Стоит отметить, что обычного шифрования беспроводного трафика в большинстве случаев недостаточно, в связи с появлением огромного количества утилит для его перехвата и расшифровки.

Приведем пример использования PPPOE для создания туннеля между маршрутизаторами.

На втором маршрутизаторе создадим PPPOE сервер:

/ppp profile add name=filial only-one=yes use-compression=yes use-encryption=yes

/interface pppoe-server server add interface=ether1 service-name=filial1one-session-per-host=yes default-profile=filial disabled=no use-vj-compression=yes

/ppp secret add name=newuser password=newpassword local-address=192.168.5.1 remote-address=192.168.5.2 service=pppoe

На первом:

/interface pppoe-client add name=filial_connection service-name=filial1 user=newuser password=newpassword allow=mschap2 disabled=no

Описание VLAN

Технология VLAN позволяет организовать виртуальные каналы между узлами связи на 2-ом уровне модели OSI. Стандарт 802.1Q описывает принципы построения виртуальных сетей на одном физическом Ethernet-интерфейсе. Большинство современных маршрутизаторов и коммутаторов умеют работать с этой технологией, которая достаточно часто встречается в современной практике.

Реализация VLAN в RouterOS Mikrotik разрешает использовать до 4095 виртуальных интерфейсов и предоставляет надежный транспорт для других протоколов высших уровней.

Использование VLAN дает ряд преимуществ перед туннельными протоколами третьего уровня, предоставляя логические высокоскоростные защищенные интерфейсы, которые ничем не отличаются в принципах работы от обычных физических.

Ниже представлен список сетевых адаптеров, которые корректно работают с 802.1Q:

– Realtek 8139;
– Intel PRO/100;
– Intel PRO1000 server adapter;
– National Semiconductor DP83816 based cards (RouterBOARD200 onboard Ethernet, RouterBOARD 24 card);
– National Semiconductor DP83815 (Soekris onboard Ethernet);
– VIA VT6105M based cards (RouterBOARD 44 card);
– VIA VT6105;
– VIA VT6102 (VIA EPIA onboard Ethernet).

Следующие сетевые адаптеры работают с 802.1Q в ограниченном режиме функциональности и не рекомендуются для использования:

– 3Com 3c59x PCI;
– DEC 21140 (tulip).

Приведем пример конфигурирования двух маршрутизаторов, использующих VLAN для связи друг с другом.

Первый маршрутизатор:

/interface vlan add name=vlan1 vlan-id=10 interface=ether1

/ip address add address=192.168.5.1/24 interface=vlan1

Второй маршрутизатор:

/interface vlan add name=vlan1 vlan-id=10 interface=ether1

/ip address add address=192.168.5.2/24 interface=vlan1

После того, как в статусной строке каждого VLAN интерфейса появилось слово running, попробуйте пропинговать созданные адреса:

ping 192.168.5.1
192.168.5.1 64 byte pong: ttl=255 time=1 ms
192.168.5.1 64 byte pong: ttl=255 time=3 ms
и
ping 192.168.5.2
192.168.5.2 64 byte pong: ttl=255 time=3 ms
192.168.5.2 64 byte pong: ttl=255 time=2 ms

Описание IPSec

Набор протоколов IPSec был разработан специально для сокрытия информации, передаваемой чрез открытые сети. Принципы их реализации значительно повлияли на подход к созданию IPv6 и развитие систем передачи данных промышленных стандартов.

Все протоколы IPSec делятся на два типа:

– протоколы шифрования и формирования шифрованного потока;

– протоколы обмена ключами.

К протоколам первого типа относятся ESP (Encapsulating Security Payload – инкапсуляция зашифрованных данных) и AH (Authentication Header – аутентифицирующий заголовок). Стоит отметить, что AH не подразумевает обеспечения конфиденциальности передаваемых данных и отвечает только за проверку их целостности.

К протоколам второго типа относится только один существующий на данный момент, IKE (Internet Key Exchange). Данный протокол обычно используется в двух случаях:

– передаваемый трафик попал под какое-либо правило, по которому он должен быть зашифрован и у клиента нет данных для его шифрования (Security Associations SA). В этом случае он отправляет запрос на получение ключа своему оппоненту;

– клиент получил запрос на получение ключа и должен ответить вызывающей стороне.

Протоколы IPSec, отвечающие за передачу зашифрованных данных, могут работать в двух режимах: транспортном (создание зашифрованного туннеля между маршрутизаторами) и туннельном (создание подключения между сетями и построение виртуальных частных сетей). Транспортный режим подразумевает шифрование только блока транспортных данных IP-пакета. Туннельный режим обязывает шифровать пакет полностью и инкапсулировать его в другой UDP-пакет, чем обеспечивается его беспроблемная маршрутизация. Также никак не влияет на маршрутизацию шифрование только поля данных IP-пакетов.

В ситуации, когда IPSec пакеты сгенерированы с использованием AH (Authentification Header), недостаточно применения технологии NAT. Структура IP-пакета, подверженного обработке IPSec протоколом, меняется, что делает невозможным его правильное распознавание. Для устранения этой проблемы прибегают к технологии NAT-Traversal, которая инкапсулирует IPSec трафик в UDP-пакеты и передает их по сети в виде привычного маршрутизируемого сетевого трафика. На принимающей стороне от пакета отбрасывается UDP-заголовок и концевик, и на стек протокола IPSec поступают полученные данные.

RouterOS Mikrotik имеет следующие средства для работы с IPSec: создание политик для шифрования правил, автоматическую генерацию ключей, ручное создание правил для шифрования трафика, работу как в транспортном режиме, так и в режиме туннелирования, средства мониторинга. Кроме того, в файерволе системы предусмотрен механизм NAT-T, о котором было рассказано выше.

Для создания простейшего транспортного IPSec подключения между двумя маршрутизаторами нужно:

– на первом маршрутизаторе выполнить:

/ip ipsec policy add sa-src-address=192.168.1.1 sa-dst-address=192.168.1.2 action=encrypt

/ip ipsec peer add address=192.168.1.2/24 secret=«drivermania.ru» generate-policy=yes

– на втором маршрутизаторе выполнить:

/ip ipsec policy add sa-src-address=192.168.1.2 sa-dst-address=192.168.1.1 action=encrypt

/ip ipsec peer add address=192.168.1.1 secret=«drivermania.ru»

Также на обоих маршрутизаторах необходимо разрешить используемые протоколами IPSec порты:

/ip firewall filter add chain=input protocol=udp dst-port=500 action=accept comment=«Allow IKE» disabled=no

/ip firewall filter add chain=input protocol=ipsec-esp action=accept comment=«Allow IPSec-esp» disabled=no

/ip firewall filter add chain=input protocol=ipsec-ah action=accept comment=«Allow IPSec-ah» disabled=no

Если у вас не возникло никаких трудностей с вышеописанным, откройте статистику и посмотрите шифруются ли пакеты:

```
ip ipsec> counters print
out-accept: 7
out-accept-isakmp: 0
out-drop: 0
out-encrypt: 8
in-accept: 16
in-accept-isakmp: 0
in-drop: 0
in-decrypted: 7
in-drop-encrypted-expected: 0
```

В случае использования IPSec в туннельном режиме для объединения сетей с адресами 192.168.10.0/24 и 192.168.20.0/24 правила будут выглядеть следующим образом:

/ip firewall nat add chain=srcnat src-address=192.168.10.0/24 dst-address=192.168.20.0/24 out-interface=public action=masquerade

/ip ipsec policy add src-address=192.168.10.0/24 dst-address=192.168.20.0/24 action=encrypt tunnel=yes sa-src-address=192.168.1.1 sa-dst-address=192.168.1.2

/ip ipsec peer add address=192.168.1.2 exchange-mode=aggressive secret=«drivermania.ru»

и

/ip firewall nat add chain=srcnat src-address=192.168.20.0/24 dst-address=192.168.10.0/24 out-interface=public action=masquerade

/ip ipsec policy add src-address=192.168.20.0/24 dst-address=192.168.10.0/24 action=encrypt tunnel=yes sa-src-address=192.168.1.2 sa-dst-address=192.168.1.1

/ip ipsec peer add address=192.168.1.1 exchange-mode=aggressive secret=«drivermania.ru»

В результате, маршрутизаторы 192.168.1.1 и 192.168.1.2 будут обмениваться ключами и создадут безопасный шифрованный туннель между сетями 192.168.20.0 и 192.168.10.0.

Как вы уже могли заметить, в показанных выше примерах мы оперировали двумя типами правил: правилами для указания политик шифрования (policy) и правилами для указания источников ключей (peer). На них стоит остановиться поподробнее и разъяснить некоторые параметры, однако, для начала рассмотрим еще два типа правил, касающихся IPSec, – это Proposals и ManualSAs. Создание Proposals можно сравнить с созданием профилей шифрования. Среди доступных опций предусмотрены:

– алгоритмы генерации данных для аутентификации, которые могут принимать значения: md5, sha1, null;

– алгоритмы генерации данных для шифрования со значениями: des, 3des, aes-128, aes-192, aes-256, null.

Также возможно указать время жизни профиля в секундах или байтах и способ генерации материала для шифрования из списка, предложенного ниже:

– modp768;
– modp1024;
– modp1536;
– none.

Профили Proposals используются в качестве опции при создании политик (Policy) ManualSAs.

Данный пункт предназначен для ручного создания Security Associations. Этот способ обычно используется для повышения сложности декодирования перехваченных данных и будет приемлем для ускорения работы протокола за счет ненадобности генерировать и создавать SA на обоих хостах.

Приведем пример создания шифрованного туннельного подключения двух роутеров при помощи ручного задания SA:

/ip ipsec manual-sa add name=ah-sa1 ah-spi=0x101/0x100 ah-key=drivermania.ru

/ip ipsec policy add src-address=192.168.10.0/24 dst-address=192.168.20.0/24 action=encrypt ipsec-protocols=ah tunnel=yes sa-src=192.168.1.1 sa-dst=192.168.1.2 manual-sa=ah-sa1

и

/ip ipsec manual-sa add name=ah-sa1 ah-spi=0x101/0x100 ah-key=drivermania.ru

/ip ipsec policy add src-address=192.168.20.0/24 dst-address=192.168.10.0/24 action=encrypt ipsec-protocols=ah tunnel=yes sa-src=192.168.1.2 sa-dst=192.168.1.1 manual-sa=ah-sa1

Как видите, необходимость задавать Peers отсутствует, так как данные для шифрования/дешифрования заданы вручную при помощи ManualSAs и алгоритм IKE не используется.

Протоколы IPSec являются самыми совершенными и криптозащищенными среди всех описанных нами, поэтому, если вам необходим крайне высокий уровень защиты передаваемых данных, то это ваш выбор.

Выводы

Итак, вы познакомились с самыми популярными способами создания туннелей между удаленными сетями. Если необходимость максимально засекретить передаваемую информацию – ваша главная прерогатива, то, несомненно, выбор должен пасть на IPSec, если вам необходим простой и быстрый туннель с шифрованием трафика, то стоит обратить внимание на PPTP/L2TP. Если вы не хотите оперировать IP-адресами и таким образом засекретить передаваемые данные, то стоит обратить внимание на PPPOE и VLAN. Если же вам нужен простой, но быстрый туннель между удаленными маршрутизаторами – ваш выбор IPIP. Ну и, несомненно, в случае необходимости прозрачного объединения сетей самым оптимальным будет использование протокола EoIP поверх уже созданных виртуальных каналов.

Каждый раз перед выбором одной из представленных технологий вы должны четко понимать, что требуется получить и что позволит решить задачу максимально надежно и безопасно, а для понимания всего этого нужен банальный опыт. Так что, побольше экспериментируйте и у вас все получится!

ОАО «АГАТ-СИСТЕМЫ УПРАВЛЕНИЯ» – ПРИЗНАННЫЙ ЛИДЕР НА РЫНКЕ БЕЛОРУССКОЙ ЭНЕРГЕТИКИ

Ирина Петрушко, начальник отдела маркетинга и продаж ОАО «АГАТ-системы управления» – управляющая компания холдинга «Геоинформационные системы управления»

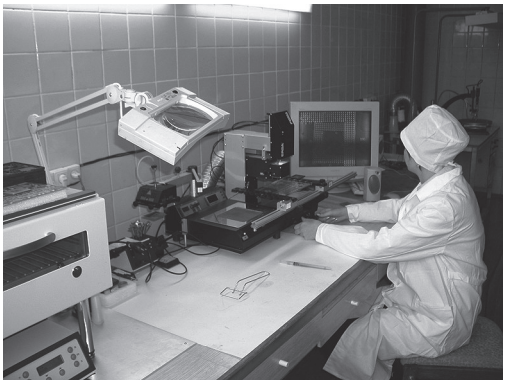
ОАО «АГАТ-системы управления» – управляющая компания холдинга «Геоинформационные системы управления» – это многопрофильная организация, одна из ведущих компаний Республики Беларусь по созданию автоматизированных систем управления общего и специального назначения, аппаратно-программных комплексов и технических средств, в том числе средств связи и передачи данных, вычислительной техники, контрольно-измерительных приборов, систем жизнеобеспечения и электропитания.

Организация имеет богатые традиции, ведет свою историю с 1969 года, когда был создан Научно-исследовательский институт средств автоматизации, и выполняет проекты по приоритетным научным направлениям. Разрабатываемые изделия во многом уникальны и не имеют аналогов в странах СНГ и за рубежом. Они конкурентоспособны на мировом рынке и приносят Республике Беларусь стабильный доход. Сегодня в числе заказчиков организации – страны СНГ, Азиатского региона, арабского мира и Европейского Союза.

Пожалуй, сложно назвать сегодня отрасль народного хозяйства Республики Беларусь, в которой так или иначе не применялись бы разработки ОАО «АГАТ-системы управления». В основе создаваемой продукции лежат инновационные системные и технические решения для управления объектами энергетики и нефтехимической отрасли, различными видами транспорта (воздушным движением, железнодорожным и автомобильным транспортом, движением на автомагистралях), а также десятками других направлений.

Сильной стороной научно-технической политики ОАО «АГАТ-системы управления» в развитии проектирования и производства продукции является масштабное техническое перевооружение. Созданы десятки моделирующих центров, оснащенных самой современной компьютерной техникой с использованием передовых информационно-коммуникационных технологий как своей, так и зарубежной разработки. Создана уникальная методология разработки больших интегрированных систем управления, освоены и применяются новейшие технологии, которые на Западе относят к «критическим» («ключевым»).

В организации внедрена автоматизированная система сквозного конструкторско-технологического проектирования, что, в конечном итоге, повысило научно-технический уровень создаваемых изделий и их качество. ОАО «АГАТ-системы управления» имеет собственную оснащенную производственную и испытательную базу и обеспечивает полный цикл разработки изделий – от эскизного проекта до серийного производства. Осуществляется планомерная модернизация производства с целью наращивания объемов выпускаемой продукции, освоения новых видов продукции, освоения и внедрения в производство новых перспективных технологий. Система качества Общества сертифицирована на соответствие требованиям СТБ ISO 9001-2009. Инновационный и комплексный подход к реализации проектов позволил ОАО «АГАТ-системы управления» – управляющая компания холдинга «Геоинформационные системы управления» стать одним из основных исполнителей работ по автоматизации объектов Белорусской энергосистемы, в том числе, на национальном уровне.

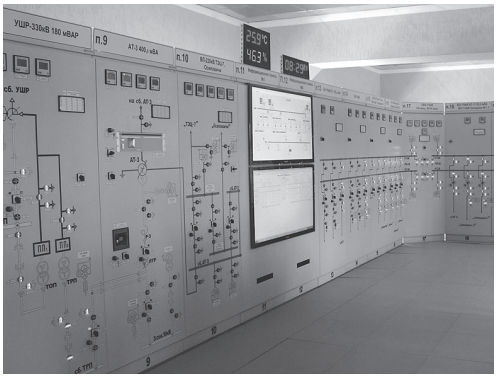


В настоящее время в нашей стране создается многоуровневая автоматизированная система контроля и учета электрической энергии (АСКУЭ) Республики Беларусь, включающая АСКУЭ межсистемных и межгосударственных перетоков и генерации электроэнергии (АСКУЭ ММПГ), АСКУЭ областных энергосистем, электросетей и их филиалов (региональные АСКУЭ), АСКУЭ промышленных потребителей, коммунально-бытового сектора, непромышленных и иных потребителей.

ОАО «АГАТ-системы управления» является Генпроектировщиком и Генподрядчиком по выполнению работ, связанных с проектированием и внедрением системы АСКУЭ ММПГ на энергообъектах энергосистемы Республики Беларусь (Указ Президента Республики Беларусь от 9 ноября 2006 г. №655), которая в настоящее время введена в промышленную эксплуатацию. Организацией выполняются работы по созданию и внедрению региональных АСКУЭ (решение Министерства энергетики Республики Беларусь от 12.11.2007 г.).

Организация проводит большую практическую работу по внедрению АСКУЭ и автоматизированных систем управления электроэнергией (АСУЭ) на промышленных предприятиях и иных объектах различной формы собственности, внедрение которых помогает хозяйственникам оптимизировать затраты на электроэнергию, что в конечном итоге отражается на себестоимости продукции.

В основе проектов, выполняемых организацией, лежит комплексный подход к построению систем, собственные системные и программные решения, широкий спектр интегрируемых в системы различных средств измерения ведущих фирм СНГ и Европы, собственное оборудование.



Разработанное специалистами ОАО «АГАТ-системы управления» программное обеспечение АСУ технологическими процессами на объектах энергетики и промышленных предприятиях «АГАТ-2000» (ПО АСУ ТП «АГАТ-2000») обеспечивает:

– высокую экономичность, надежность и готовность к управлению процессом передачи электроэнергии;

– снижение вероятности аварийных ситуаций и повреждения основного оборудования;

– оптимизацию планирования работы и вывода в ремонт электрооборудования;

– высокую надежность работы системы за счет улучшения качества и организации ремонтно-профилактических работ, анализа действий оперативного персонала, устройств релейной защиты и автоматики (РЗА) и противоаварийной автоматики (ПА) на основе регистрации аварийных ситуаций, контроля отклонений основных параметров электрооборудования и решения задач его диагностики;

– достоверность и информативность данных о ходе технологических процессов и состоянии оборудования и представление их в полном объеме оперативному персоналу;

– оптимизацию работы оборудования и энергообъекта в целом, снижение расхода электроэнергии на собственные нужды;

– высокую оперативность и обоснованность принимаемых решений;

– сокращение ошибок оперативного персонала;

– снижение затрат на ремонт и техническое обслуживание оборудования, устройств РЗА и технических средств системы;

– гибкость и открытость архитектуры системы;

– возможность расширения функциональных задач АСУ ТП энергообъектов;

– возможность автоматизации новых энергетических объектов;

– наличие удобного пользовательского интерфейса;

– высокую надежность системы за счет резервирования серверов;

– защиту от несанкционированного доступа.

Тесное сотрудничество пользователей системы и разработчиков организации обеспечивает индивидуальный подход к каждому заказчику, постоянное развитие и усовершенствование ПО, что позволяет получить не только «умный», но и удобный инструмент для энергосбережения, учитывающий как особенности конкретного объекта, так и все пожелания заказчика.

Сегодня ОАО «АГАТ-системы управления» предлагает следующие решения на основе ПО АСУ ТП «АГАТ-2000» для объектов энергетики, жилищно-коммунального хозяйства и промышленных предприятий:

– автоматизированные системы контроля и учета энергоресурсов (электроэнергии, тепла, воды) (АСКУЭ);

– информационно-аналитическое решение для построения систем учета энергоресурсов ЦСОИ «АГАТ Энерго»;

– система контроля, защиты и управления (СКЗУ) «АГАТ-2000».

Организация разработала и освоила серийное производство собственного оборудования для построения систем АСКУЭ, а также автономного применения:

– устройство контроля параметров качества электрической энергии УК1;

– устройства для сбора, временного хранения, обработки и передачи информации: модули сопряжения МС5, МС20;

– электротехническое оборудование: шкафы счетчиков, УСПД, телемеханики, релейной защиты и автоматики.

ОАО «АГАТ-системы управления» – управляющая компания холдинга «Геоинформационные системы управления» более 10 лет выпускает Устройство контроля параметров качества электрической энергии УК1, хорошо известное как инспекторам Энергосбыта и Энергонадзора, так и энергетикам крупных предприятий республики.

УК1 представляет собой высокоточный измерительный прибор, построенный на основе современных цифровых технологий. Устройство устанавливается на энергообъектах и осуществляет сбор, обработку и хранение информации о параметрах качества электрической энергии в соответствии с требованиями ГОСТ 13109-97. При необходимости удаленного сбора информации с УК1 его

можно использовать совместно с модулем сопряжения МС5. Данное оборудование может интегрироваться в систему АСКУЭ.

Контроль качества электроэнергии позволяет выявить отклонения параметров питающей сети от номинальных значений, которые значительно снижают экономичность работы потребителей электроэнергии за счет уменьшения производительности технологических установок, сокращения сроков службы электротехнического и электронного оборудования и могут наносить прямой материальный ущерб из-за нарушения технологических процессов изготовления брака продукции. Кроме того, это зачастую приводит к повышенному расходу электрической энергии.

Устройство рекомендовано концерном «БелЭнерго» для применения в службах «Энергонадзора», на объектах электроэнергетики, промышленных предприятиях и учреждениях для контроля параметров качества электроэнергии. УК1 успешно применяется в службах Энергонадзора РУП «Гомельэнерго», «Минскэнерго», «Витебскэнерго», «Гродноэнерго», «Брестэнерго», научно-исследовательских лабораториях, РУП «БелГИМ», БелГИСС, ЗАО «АТЛАНТ», РУП «МТЗ», ПО «Белоруснефть», ОАО «Брестгазоаппарат», ПРУП «БЭТЗ» БелЖД и других крупных предприятиях республики.

Устройство УК1 внесено в Государственные реестры средств измерений Республики Беларусь, Российской Федерации и имеет соответствующие сертификаты.

ОАО «АГАТ-системы управления» освоило производство линейки устройств для сбора, временного хранения, обработки и передачи информации: модули сопряжения МС5, МС20.

МС5 может применяться для построения автоматизированных систем телеметрии и управления различного назначения: автоматизированные системы контроля и учета энергоресурсов, автоматизированных систем управления электроосвещением, систем удаленной охранной сигнализации (с контролем через сотовый телефон).

Отличительной особенностью МС5 от существующих на рынке аналогичных устройств является его невысокая стоимость, компактный размер, унифицированность применения. К преимуществам МС5 следует отнести:

– возможность обмена данными одновременно по нескольким проводным интерфейсам: RS485, RS232, Ethernet 10Mbit, «токовая петля», USB 1.0 (в режиме ведущего или ведомого);

– наличие GSM/GPRS модема;

– наличие дискретных входов/выходов;

– наличие часов реального времени с автономным батарейным питанием;

– возможность работы в «прозрачном» режиме обмена данными;

– возможность перепрограммирования модуля для работы с устройствами, обладающими уникальными протоколами/алгоритмами работы;

– удаленное перепрограммирование модуля;

– многовариантность исполнения модуля с различными функциональными возможностями и стоимостью;

– возможность разработки модификации с функциональными возможностями по требованию заказчика.

Модуль сопряжения МС20 является самостоятельным устройством для сбора, временного хранения и обработки информации от оконечных устройств и передачи ее на верхний уровень к центрам сбора/обработки информации. Основная сфера применения МС20 – системы АСКУЭ. Применение модуля МС20 как основного связующего элемента АСКУЭ между верхним уровнем и счетчиками электрической энергии позволяет значительно упростить схему прохождения информации. Наличие опторазвязанных последовательных портов и порта Ethernet избавляет от необходимости применения внешних устройств защиты. Возможность независимой работы последовательных портов позволяет подключить достаточное количество счетчиков электрической энергии при условии частого их опроса. Снижается количество преобразователей интерфейсов и коммутирующих проводов. Крепление на DIN-рейку позволяет оперативно производить монтаж МС20 в электронном шкафу АСКУЭ.

ПО, установленное на МС20, позволяет решать следующие задачи:

– получать оперативную информацию для эффективной организации процесса распределения и потребления электрической энергии в основном и вспомогательном производстве;

– увеличивать скорость обработки информации;

– увеличивать достоверность получаемой информации;

– организовывать и упорядочивать учет и отчетность;

– улучшать организацию управления энергетическим хозяйством путем четкого разделения задач и функций;

– осуществлять оперативный контроль за энергохозяйством с возможностью обнаружения и предупреждения аварийных ситуаций, задания тарифных зон, графика праздничных и выходных дней, считывание текущей информации

(токов, напряжений, мощности) и архивной информации (активной и реактивной энергии, энергии по тарифным зонам, мощности), просмотр всех видов информации, полученных от счетчиков, формирование ведомостей, ведение отчетной документации, расчет всевозможных балансов подстанций (предприятий, сетей и др.).

Применение модуля МС20 позволяет вести как технический, так и коммерческий учет потребления энергоресурсов в зависимости от используемых в системе приборов учета электроэнергии (счетчиков технического или коммерческого учета). МС20 обеспечивает автоматизацию сбора, обработки и представления информации о ходе процесса потребления электрической энергии и состоянии оборудования электрических сетей.

ОАО «АГАТ-системы управления» – управляющая компания холдинга «Геоинформационные системы управления» в качестве Генерального подрядчика обеспечивает выполнение всего комплекса работ по созданию АСКУЭ и гарантирует сдачу системы в эксплуатацию с выполнением всех требований Заказчика в соответствии с Техническим заданием:

– предпроектное обследование;

– разработка технического задания;

– проектирование (архитектурные и строительные проекты по промышленным предприятиям, подстанциям, электростанциям, РП, ТП);

– изготовление, поставка оборудования и программного обеспечения СКЗУ, АСКУЭ;

– проведение строительно-монтажных и пусконаладочных работ;

– ввод в эксплуатацию (с проведением метрологической аттестации системы);

– обучение персонала Заказчика;

– сопровождение (гарантийное, послегарантийное, сервисное обслуживание).

Все технические и программные решения до реализации системы у Заказчика тестируются и апробируются в комплексно-моделирующем центре АСКУЭ на базе действующего оборудования в ОАО «АГАТ-системы управления».

Опыт организации в разработке АСКУЭ, наличие производственно-испытательной базы и моделирующего центра АСКУЭ позволяют создавать автоматизированные системы самого высокого качества. Тесное сотрудничество с заказчиком не только позволяет быстро и гибко реагировать на появляющиеся новые концепции и разработки в сфере энергосбережения, но и ведет к постоянному развитию и усовершенствованию как каждой системы в целом, так и отдельных ее элементов.

Уникальный опыт разработчика и системного интегратора, ориентация на комплексное решение задач автоматизации с использованием современного оборудования, высококвалифицированных специалистов, имеющих опыт участия в крупных проектах, обширная география присутствия в сочетании с высокой мобильностью и достаточной технической оснащенностью позволяют сегодня ОАО «АГАТ – системы управления» – управляющая компания холдинга «Геоинформационные системы управления» оставаться лидером в своей отрасли.

НОВОСТИ

Награда «iF Product Design Award 2013»

Продукция компании Advantech, одного из лидирующих мировых производителей на рынке промышленной автоматизации и вычислительных систем, была впервые отмечена наградой «iF Product Design Award 2013». Обойдя более 3000 конкурентов, две линейки продукции для систем человеко-машинного интерфейса (ЧМИ) – серии SPC и TPC – удостоились награды «iF Product Design Award 2013» и будут отмечены престижным отличительным знаком «iF label» в номинации промышленность/квалифицированная деятельность.

Начиная с 2010 года, компания Advantech выбрала в качестве корпоративной идеологии концепцию «Наделения Планеты Интеллектом» и положила начало активному организационному развитию, преобразованию своей маркетинговой программы и устойчивому расширению деятельности. Все эти мероприятия позволяют претворить в жизнь основную миссию компании по внедрению интеллектуальных решений в различные области работы и образа жизни человека.

Следующее поколение систем ЧМИ от компании Advantech – серии SPC и TPC – имеют ряд удобных для пользователя конструктивных особенностей, к которым относятся: емкостной сенсорный экран с поддержкой технологии Multitouch, широкий экран формата 16:9, интеллектуальные клавиши и адаптируемый пользовательский интерфейс, обеспечивающий интуитивно понятное и эргономичное управление для систем ЧМИ.

За более, чем 60-летнюю историю премия «iF Product Design Award» завоевала статус международной признанной награды за выдающийся дизайн, а бренд iF стал престижным символом особых успехов в области разработки продуктов.

advantech.com

ПРОИЗВОДСТВЕННАЯ ОТКРЫТОСТЬ

Светотехнический холдинг «Светлана-Оптоэлектроника» основан в 1998 году и сегодня является одним из крупнейших в России и Восточной Европе светодиодных производств полного цикла. Компания работает в сфере инноваций уже более 15 лет.

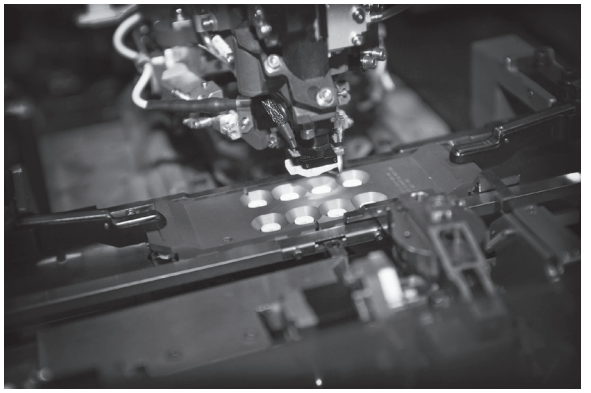
Инновационные предприятия, как в России, так и во всем мире, можно пересчитать по пальцам. Работать с «будущим» и получать от такого бизнеса прибыль – непростая, порой нерешаемая задача. Поэтому производства, успешно работающие в сфере нанотехнологий, это, прежде всего, предмет гордости для государств в целом и… главная тема для обсуждения скептиков. Большинство самих участников рынка уверены, что инновации в современной России и странах СНГ могут выжить только при серьезной господдержке, либо предприятия, работающие в этой сфере, могут существовать только в виде «потемкинской деревни». Лицевая сторона – бодрые пресс-релизы в СМИ с внушительными цифрами, серьезные темы докладов на различных конференциях и даже специальные экскурсии на производство, где посетителям издалека показывают некое загадочное оборудование, которое, со слов сотрудников компании, работает и даже что-то производит. Изнанка, в большинстве случаев, оказывается предприятием, работающим с вторичным сырьем и выдающим на рынок конечный продукт под видом российского ноу-хау по более высокой цене. Однако ярлык «инновационное предприятие» во многом дает карт-бланш на дополнительные субсидии и продвижение этого товара.

«Вычислить», какое предприятие настоящее, а какое «фейк» (от англ. fake – поддельный), с первого взгляда, практически невозможно. Убедиться, кто как работает, можно лишь воочию. «Производственная открытость» – явление для российского бизнеса новое, но за короткое время, ставшее практически обязательным для всех серьезных компаний, которые стремятся показать, что им скрывать нечего, да и не за чем.

Об основательности инновационного предприятия могут свидетельствовать и главные заказчики продукции, которую компания выпускает. Один из самых убедительных аргументов в бизнесе – это многолетнее сотрудничество с частными корпорации, которые должны быть уверены в успешном развитии своего будущего. На сегодня основными партнерами Группы компаний «Светлана-Оптоэлектроника» являются крупнейшие промышленные предприятия, транспортные компании, объекты энергетики федерального значения, лидеры дорожной инфраструктуры. Процентное соотношение заказов на светодиодную продукцию светотехнического холдинга «Светлана-Оптоэлектроника» в разных секторах экономики на 2013 год выглядит следующим образом:

– промышленные предприятия – 40 %;
– транспортная инфраструктура – 25 %;
– муниципальные и коммерческие учреждения – 20 %;
– прочее – 15 %.

Производственная территория группы компаний «Светлана-Оптоэлектроника» внушительная – три корпуса зданий, в которых последовательно осуществляются все технологические этапы производства светодиодов и последующего выпуска осветительных приборов на их основе, а также охранно-пожарной продукции и систем. На сегодня, это, пожалуй, единственная компания на территории нашей страны, где на одной производственной площадке располагается цех производства



светодиодов – так называемые «чистые помещения» и оборудование для эпитаксиального выращивания полупроводниковых гетероструктур. Все остальные стадии технологического процесса также выполняются силами предприятия: есть фотометрическая лаборатория, цеха литья, штамповки, сборки светотехники, упаковки. Таким образом, «Светлана-Оптоэлектроника» – единственное предприятие, где действительно организован полный технологический цикл: от выращивания наногетероструктур до создания светильников и автоматизированных систем интеллектуального управления освещением.

Максимальная мощность «Светлана-Оптоэлектроника» составляет, в пересчете на мощность светодиодов, 3 МВт в месяц. В 2003 году был получен сертификат ISO 9001, а чуть позже на одной площадке был освоен полный цикл производства светодиодов. В 2013 году ежедневный выпуск светодиодов составляет более 200 000, обеспечивая тем самым потребности группы компаний для выполнения заказов на светотехническую продукцию от ее партнеров.

Несмотря на новое технологическое оборудование, производственные помещения постоянно модернизиру-

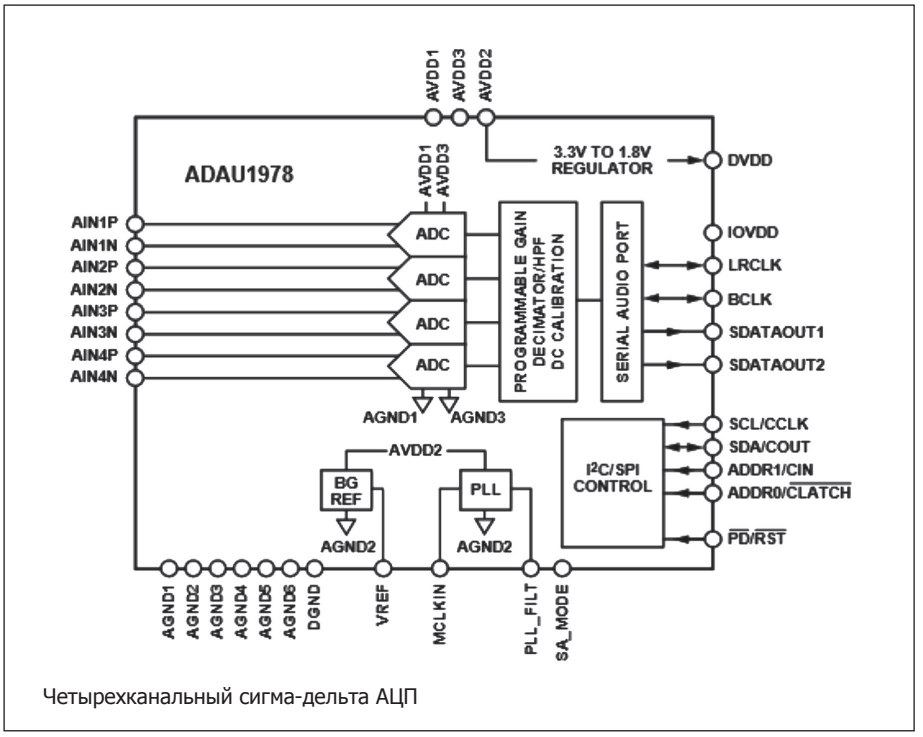
ются. Так, например, в дополнение к установке эпитаксиального выращивания, эксплуатируемой с середины 2000-х, недавно был приобретен новый, автоматический реактор, позволяющий производить светодиоды с более точными характеристиками. Другой пример: несколько месяцев назад на производстве компании был открыт экспериментальный цех по запуску нового процесса, исключающего свинцовую пайку светодиода! Такое ноу-хау, по словам экспертов холдинга, еще один шаг к экологичности выпускаемой продукции. Светодиодными платами, собранными по этой технологии, будут оснащены уличные светильники SvetaLED®, которые будут установлены в ближайшее время в Санкт-Петербурге, Вологде и Челябинске. Для последнего это решение особенно актуально, так как при падении метеорита в феврале 2013 года было разбито более 500 уличных ламп, содержащих ртуть. До сих пор экологи подсчитывают ущерб, нанесенный экологии города. Светодиодные светильники не содержат вредных для здоровья компонентов и поэтому на текущий момент, они считаются самыми безопасными источниками искусственного освещения.

На предприятии постоянно совершенствуется система обеспечения качества, что способствует повышению качества выпускаемых изделий и снижению процента брака. Сегодня вся продукция проходит функциональный контроль, часть проходит полный контроль в различных режимах. «Светлана-Оптоэлектроника» располагает полноценной испытательной лабораторией, позволяющей производить измерения параметров выпускаемых ламп, светильников и светодиодов. Инженерно-технические специалисты предприятия – в основном выходцы из Политехнического Университета, ЛЭТИ, ЛИТМО. Компания удерживает высокую планку в сфере научных достижений. В прошлом году ЗАО «Светлана-Оптоэлектроника» получила ряд значительных патентов на изобретения, а профессиональный уровень сотрудников холдинга был отмечен престижной премией «Эксперт года». В 2012 году лауреатом премии в номинации «Наука и инновации» стала – главный технолог «Светлана-Оптоэлектроника» – Елена Маслова. В этом году победу в той же номинации одержал заместитель генерального директора по научной и проектной работе – Дмитрий Бауман. Также нельзя не упомянуть и про награду «Компания года 2012», и премию по качеству правительства Санкт-Петербурга.

Снова вошедший в моду тренд «сделано в России» отчасти должен быть обязан своей востребованностью именно таким предприятиям, как «Светлана-Оптоэлектроника». Инновации, которые еще лет 10 назад казались фантастической выдумкой, сегодня становятся неотъемлемой частью нашей повседневной жизни. Отрадно, что в холдинге «Светлана-Оптоэлектроника» будущее обретает не только форму, но и свет.

Официальный дистрибьютор на территории РБ
ООО «Белпромэнергоэффект»
Belpee.by

АНАЛОГО-ЦИФРОВЫЕ ПРЕОБРАЗОВАТЕЛИ (АЦП)

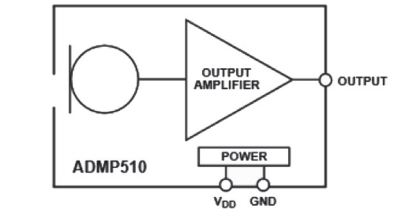


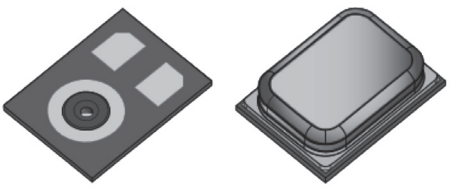
Четырехканальный сигма-дельта АЦП

ADAU1978 включает в себя четыре высококачественных АЦП, стабилизатор напряжения, источник опорного напряжения на запрещенной зоне и схему ФАПЧ, комбинация которых образует аналоговый входной интерфейс для автомобильных аудиосистем. Входной каскад АЦП работает с полностью дифференциальными сигналами со связью по переменному току и среднеквадратическим уровнем до 2 В. Архитектура АЦП на базе многоразрядного сигма-дельта модулятора с непрерывным интегрированием обеспечивает низкий уровень электромагнитных помех, динамический диапазон 106 дБ и полный уровень гармонических искажений и шума (THD+N) –95 дБ. Интегрированная схема ФАПЧ, способная формировать внутренний задающий тактовый сигнал из внешнего сигнала кадровой или тактовой синхронизации, избавляет от необходимости в отдельном генераторе высокочастотного задающего тактового сигнала. ADAU1978 работает с напряжениями питания 1,8 В и 3,3 В, рассеивая 56 мВт в нормальном режиме и 1 мВт в режиме пониженного энергопотребления. Компонент выпускается в 40-выводном корпусе LFCSP, рабочий температурный диапазон составляет от –40°С до +105°С.

АУДИО/ВИДЕО ПРОДУКТЫ

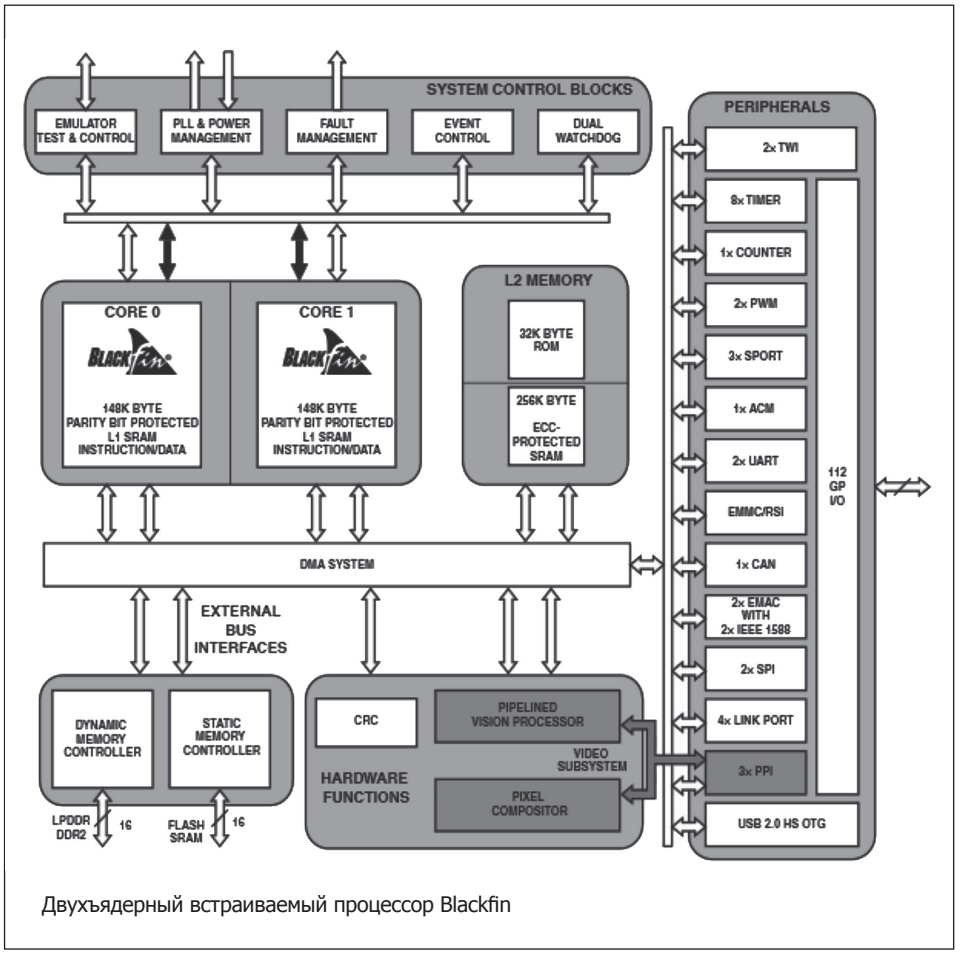
Высококачественный, малопотребляющий микрофон МЭМС ADMP510 с низким шумом и аналоговым выходом включает в себя ненаправленный, монтируемый звуковым отверстием к плате датчик, преобразователь импеданса и выходной усилитель. Благодаря чувствительности –38 дБВ он является идеальным выбором для приема сигнала как в ближней, так и в дальней зоне, а комбинация отношения сигнал-шум 65 дБА, рабочей полосы частот от 60 Гц до 20 кГц и устойчивости к радиочастотным помехам обеспечивает естественное звучание и высокую разборчивость в мобильных телефонах, планшетных ПК, цифровых камерах и гарнитурах Bluetooth. ADMP510 работает с одним напряжением питания в диапазоне от 1,5 В до 3,63 В, потребляя максимальный ток 220 мкА при 1,8 В, что позволяет продлить срок службы батарей в портативных устройствах. Компонент выпускается в миниатюрном корпусе для поверхностного монтажа с габаритами 3,35 мм × 2,5 мм × 0,98 мм, который совместим с пайкой методом оплавления припоя и не содержит галоидных соединений. Встроенный фильтр частиц повышает надежность. Компонент имеет рабочий температурный диапазон от –40°С до +85°С.





Монтируемый звуковым отверстием к плате микрофон МЭМС с повышенной устойчивостью к радиопомехам, крайне низким шумом и аналоговым выходом

ВСТРАИВАЕМЫЕ ПРОЦЕССОРЫ И DSP



Двухъядерный встраиваемый процессор Blackfin

Процессоры Blackfin® ADSP-BF606, ADSP-BF607, ADSP-BF608 и ADSP-BF609 совмещают в себе высокую эффективность обработки мультимедийных сигналов, обеспечиваемую ядром DSP с архитектурой SIMD (single-instruction, multiple-data; одна команда, много данных), и функции управления, характерные для микроконтроллера с архитектурой RISC. Они имеют два 16-разрядных умножителя-накопителя (MAC), два 40-разрядных АЛУ, четыре 8-разрядных видео-АЛУ, 40-разрядное устройство сдвига, 148 кбайт памяти SRAM L1, до 256 кбайт памяти SRAM L2 и способны работать с частотой до 500 МГц. Набор периферийных модулей включает в себя: один счетчик с инкрементом/декрементом, восемь таймеров/счетчиков с ШИМ, два трехфазных модуля ШИМ, три дуплексных синхронных последовательных порта (SPORT), которые поддерживают восемь каналов цифрового стереозвука в формате I2S, два порта SPI, один контроллер шины USB on-the-go (USB OTG), три параллельных периферийных интерфейса (PPI) с поддержкой видеоформатов ITU-R 656, один интерфейс съемных носителей информации, один интерфейс CAN (controller area network), два контроллера двухпроводного интерфейса (TWI), два универсальных асинхронных приемника/передатчика (UART) с поддержкой IrDA, один модуль управления АЦП, четыре линк-порта, два контроллера Ethernet MAC (IEEE 1588), один видеоускоритель Pixel Compositor, один конвейерный видеопроцессор и 112 линий GPIO. Эти процессоры, производимые по низковольтной технологии с низкой потребляемой мощностью, обеспечивают передовые возможности управления энергопотреблением и производительностью, что делает их идеальным выбором для приложений следующего поколения, от автомобильных систем до промышленных измерительных систем и систем управления питанием/двигателями, в которых требуется характерная для RISC процессоров модель программирования, поддержка мультимедийной информации и передовая обработка сигналов в одном интегрированном корпусе. ADSP-BF60x выпускаются в корпусе CSP-BGA с 349 шариковыми контактами, рабочий температурный диапазон составляет от 0°C до +70°C (градация K) или от –40°C до +85°C (градация B).

Оценить возможности процессора ADSP-BF609 можно используя разработанный компанией AXONIM модуль AX-SOM-BF609.

К основным преимуществам модуля следует отнести:

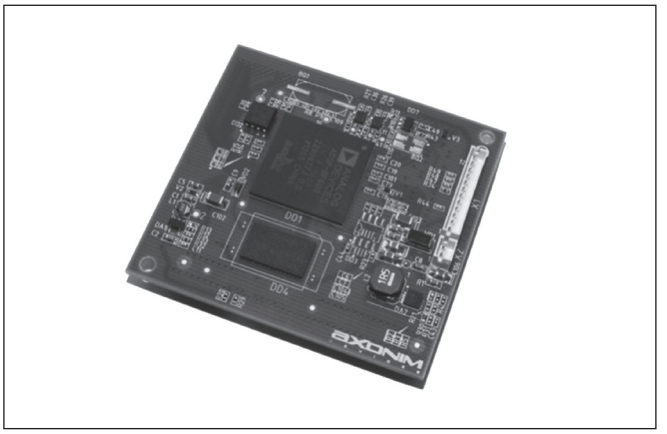
– минималистический набор компонентов, достаточный для загрузки и функционирования SoC (DDR2, SPI ПЗУ, монитор температуры кристалла и DC/DC);

– доступность всех I/O через разъемы позволяют настраивать и применять все доступные конфигурации SoC;

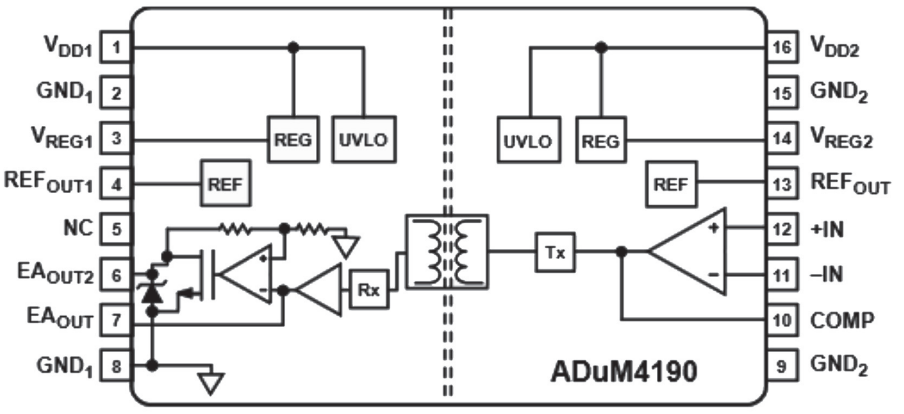
– возможность замены DDR2 ОЗУ, SPI ПЗУ на различные объемы без внесения изменений в трассировку платы и схему;

– габариты – 65x55 мм.

Более подробное описание модуля можно найти на сайте http://axonim.by или http://alfa-chip.com



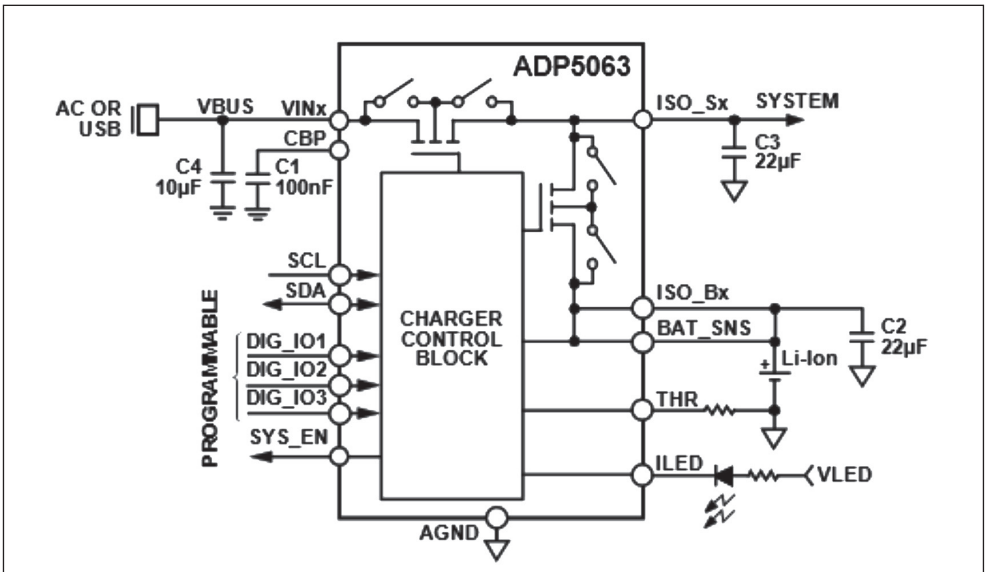
ИНТЕРФЕЙСНЫЕ КОМПОНЕНТЫ



Высокостабильный усилитель сигнала ошибки с гальванической развязкой

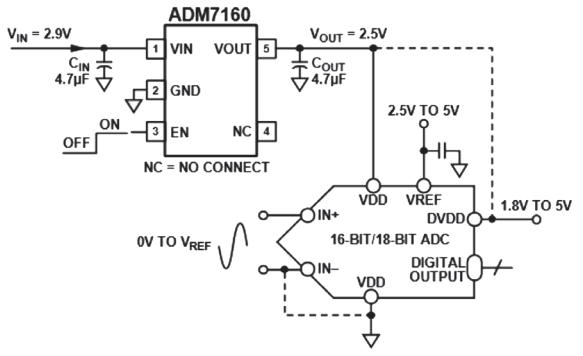
Усилитель сигнала ошибки с гальванической развязкой ADuM4190 идеально подходит для линейных источников питания с обратной связью, работающими с контроллерами на стороне первичной обмотки. Комбинация ширины полосы 400 кГц, типичной начальной погрешности 0,5 % при 25°C и полной погрешности 1 % в диапазоне рабочих температур позволяет производителям преобразователей постоянного напряжения и преобразователей переменного напряжения в постоянное, включая преобразователи стандарта DOSA (Distributed-power Open Standards Alliance), значительно повысить быстродействие и расширить рабочий температурный диапазон своих устройств, а также сократить время переходного отклика до пяти раз. Этот компонент, построенный на базе технологии цифровой изоляции iCoupler® компании Analog Devices, выдерживает напряжение до 5 кВ и не имеет характерных для оптопар проблем, связанных с ухудшением коэффициента усиления по току с течением времени. В ADuM4190 интегрированы прецизионный источник опорного напряжения 1,225 В и широкополосный операционный усилитель, который может быть использован в разнообразных типовых схемах компенсации контуров источников питания. Компонент выпускается в 16-выводном корпусе SOIC, рабочий температурный диапазон составляет от –40°C до +125°C.

УПРАВЛЕНИЕ ПИТАНИЕМ И ТЕПЛОВЫМ РЕЖИМОМ



Совместимая с USB линейная схема управления зарядкой LiFePO4 батарей с интегрированным преобразователем напряжения

ИМС управления зарядкой батареи ADP5063 позволяет заряжать мобильные телефоны, цифровые фотоаппараты и другие портативные устройства с портом mini-USB от адаптера питания, автомобильного зарядного устройства или порта USB хост-компьютера. Компонент полностью совместим со спецификацией USB3.0 и отвечает требованиям к зарядке батарей спецификации USB1.2. Он работает от входного напряжения в диапазоне от 4 В до 6,7 В и способен выдерживать напряжения от -0,5 В до +20 В, что уменьшает проблемы, связанные с пиковыми выбросами на шине при отключении или подключении устройства. Интегрированный полевой транзистор между выходом преобразователя зарядного устройства и батареей обеспечивает развязку, позволяя питать систему от порта USB, даже когда батарея отсутствует или полностью выработала ресурс. Для согласования с характеристиками конкретного источника USB-микросхема имеет возможность задания предельного тока для оптимальной зарядки и установки уровней совместимости USB. ADP5063 выпускается в 20-выводном корпусе LFCSP, рабочий температурный диапазон составляет от -40°C до +125°C.

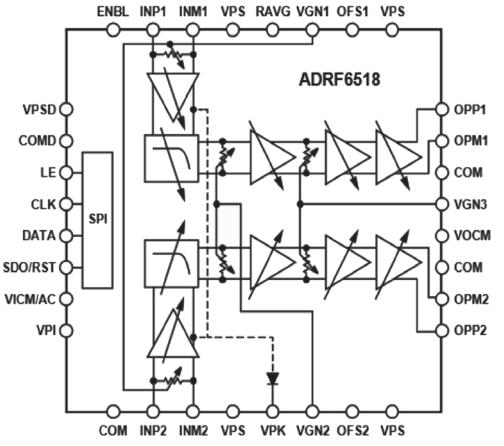


Линейный стабилизатор с крайне низким шумом и выходным током 200 мА

Линейный стабилизатор с малым падением напряжения (low-dropout, LDO) ADM7160 работает с входными напряжениями в диапазоне от 2,2 В до 5,5 В, поддерживая падение напряжения 10 мВ при выходном токе 10 мА и 150 мВ при выходном токе 200 мА, что делает его идеальным выбором высококачественных аналоговых и аналого-цифровых схем, которые требуют напряжения питания в диапазоне от 1,1 В до 3,3 В. Компонент обладает коэффициентом подавления пульсаций питания 54 дБ на частоте 100 кГц, среднеквадратическим уровнем шума 9 мкВ в полосе от 10 Гц до 100 кГц, а также обеспечивает превосходный переходный отклик на изменения тока нагрузки и входного напряжения при работе с миниатюрными керамическими конденсаторами номиналом 1 мкФ по входу и выходу. Стабилизатор имеет дрейф 29 ppm/°C, что позволяет применять его в термостабильных системах, например, при работе с термостатированными кварцевыми генераторами. Вход сигнала разрешения содержит внутренний подтягивающий к земле резистор. Схемы защиты от перегрева и перегрузки по току предотвращают повреждение в жестких рабочих условиях. ADM7160 потребляет 265 мкА при IOUT = 200 мА, 60 мкА при IOUT = 10 мА и 10 мкА в отсутствие нагрузки. Компонент выпускается в 5-выводном корпусе TSOT и 6-выводном корпусе LFCSP, рабочий температурный диапазон составляет от –40°C до +125°C.

ВЧ И ПЧ-КОМПОНЕНТЫ, КОМПОНЕНТЫ ДЛЯ СИСТЕМ БЕСПРОВОДНОЙ И ШИРОКОПОЛОСНОЙ СВЯЗИ

ADRF6518 представляет собой согласованную пару полностью дифференциальных, обладающих малым шумом и низкими искажениями усилителей с переменным коэффициентом усиления и программируемых фильтров. Они способны подавлять сильные внеполосные помехи, надежно усиливая полезный сигнал, что позволяет смягчить требования к ширине полосы и разрешению аналого-цифровых преобразователей (АЦП). Превосходное межканальное согласование и широкий свободный от побочных составляющих динамический диапазон при любых настройках коэффициента усиления и ширины полосы делают компонент идеальным выбором для систем связи с квадратурными сигналами, в которых используются сигнальные созвездия с большим количеством точек и передача сигналов на нескольких поднесущих, а также возможно присутствие близкорасположенных мешающих сигналов. Коэффициент усиления усилителей, граничная частота фильтров и другие параметры программируются через совместимый с SPI последовательный порт. Усилители с переменным коэффициентом усиления, предшествующие фильтрам, обеспечивают непрерывную регулировку в диапазоне 24 дБ с возможностью установки фиксированного коэффициента усиления 9 дБ, 12 дБ или 15 дБ и определяют дифференциальный входной импеданс компонента, который составляет 400 Ом. Фильтры имеют шестиполюсную характеристику Баттерворта с частотой среза по уровню 0,5 дБ, изменяемой в диапазоне от 1 МГц до 63 МГц с шагом 1 МГц. При необходимости работы в полосе более 63 МГц фильтры можно полностью исключить из сигнальной цепочки; при этом ширина полосы по уровню -3 дБ расширяется до 1100 МГц. Уровень пикового сигнала на входах фильтров может контролироваться при помощи широкополосного пикового детектора. Усилители с переменным коэффициентом усиления, которые следуют за фильтрами, обеспечивают непрерывную регулировку в диапазоне 24 дБ с возможностью установки фиксированного коэффициента усиления 12 дБ, 15 дБ, 18 дБ или 21 дБ. Выходные буферы дают дополнительный коэффициент



Усилители с переменным коэффициентом усиления с полосой 1100 МГц и программируемые фильтры основной полосы частот

усиления 3 дБ или 9 дБ, обеспечивают дифференциальный выходной импеданс менее 10 Ом и способны поддерживать уровень гармонических искажений третьего порядка (HD3) более 65 дБн при выдаче сигнала с размахом 1,5 В в нагрузку 400 Ом. Выходное синфазное напряжение по умолчанию равно VPS/2 и может быть понижено до 900 мВ при помощи вывода VOCM. Независимые контуры коррекции смещения постоянной составляющей можно отключить, если необходима работа со связью по постоянному току. ADRF6518 работает с одним напряжением питания в диапазоне от 3,15 В до 3,45 В, потребляя максимальный ток 400 мА в активном состоянии и 1 мА в неактивном состоянии. Компонент выпускается в 32-выводном корпусе LFCSP с открытой теплоотводящей площадкой, рабочий температурный диапазон составляет от –40°C до +85°C.

Альфасофт

НАИМЕНОВАНИЕ ТОВАРА	ЦЕНА	НАЗВАНИЕ ОРГАНИЗАЦИИ	АДРЕС, ТЕЛЕФОН
ЭЛЕКТРОТЕХНИЧЕСКАЯ ПРОДУКЦИЯ			
Индукционные лампы Saturn 40, 80, 120, 150, 290, 300W. В комплекте электронный балласт	**80-380 у.е.**	**ООО «ФЭК»**	**г. Минск. Тел./ф.: 200-34-23, тел.: 200-04-96. E-mail: lighting@fek.by**
Индукционные лампы Smart Dragon 40, 80, 120, 150, 200, 300W.	**80-380 у.е.**		
Дроссели, ЭПРА, ИЗУ, пусковые конденсаторы, патроны и ламподержатели для люминесцентных ламп	**Договор**	**ООО «Альфалидер групп»**	**г. Минск. Тел./ф.: 391-02-22, тел.: 391-03-33. www.alider.by**
Мощные светодиоды (EMITTER, STAR), сборки и модули мощных светодиодов, линзы ARLIGHT	**Договор**	**ООО «СветЛед решения»**	**г. Минск. Тел./ф.: 214-73-27, 214-73-55. E-mail: info@belaist.by www.belaist.by**
Управление светом: RGB-контроллеры, усилители, диммеры и декодеры			
Источники тока AC/DC для мощных светодиодов (350/700/100-1400 мА) мощностью от 1W до 100W ARLIGHT			
Источники тока DC/DC для мощных светодиодов (вход 12-24V) ARLIGHT			
Источники напряжения AC/DC (5-12-24-48V/ от 5 до 300W) в металлическом кожухе, пластиковом, герметичном корпусе ARLIGHT, HAITAIK			
Светодиодные ленты, линейки открытые и герметичные, ленты бокового свечения, светодиоды выводные ARLIGHT			
Светодиодные лампы E27, E14, GU 5.3, GU 10 и др.			
Светодиодные светильники, прожектора, алюминиевый профиль для светодиодных изделий			
КВАРЦЕВЫЕ РЕЗОНАТОРЫ, ГЕНЕРАТОРЫ, ФИЛЬТРЫ, ПЬЕЗОКЕРАМИЧЕСКИЕ ИЗДЕЛИЯ			
Любые кварцевые резонаторы, генераторы, фильтры (отечественные и импортные)	**от 0,10 у.е.**	**УП «Алнар»**	**г. Минск. Тел./ф.: 209-69-97, тел. (029) 644-44-09. E-mail: alnar@alnar.net www.alnar.net**
Кварцевые резонаторы Jauch под установку в отверстия и SMD-монтаж	**от 0,10 у.е.**		
Кварцевые генераторы Jauch под установку в отверстия и SMD-монтаж	**от 0,50 у.е.**		
Термокомпенсированные кварцевые генераторы	**от 2,20 у.е.**		
Резонаторы и фильтры на ПАВ			
Пьезокерамические резонаторы, фильтры, звонки, сирены	**от 0,04 у.е.**		
СПЕЦПРЕДЛОЖЕНИЕ			
Большой выбор электронных компонентов со склада и под заказ	**Договор**	**ЧТУП «Чип электроникс»**	**г. Минск. Тел./ф.: 269-92-36. E-mail: chipelectronics@mail.ru www.chipelectronics.by**
Широчайший выбор электронных компонентов (микросхемы, диоды, тиристоры, конденсаторы, резисторы, разъемы в ассортименте и др.)	**Договор**	**ООО «Альфалидер групп»**	**г. Минск. Тел./ф.: 391-02-22, тел.: 391-03-33. www.alider.by**
Мультиметры, осциллографы, вольтметры, клещи, частотомеры, генераторы отечественные и АКИП, APPA, GW, LeCroy, Tektronix, Agillent	**1-й поставщик**	**ООО «Приборо-строительная компания»**	**г. Минск. Тел./ф.: 284-11-18, тел.: 284-11-16. E-mail: 4805@tut.by**

АСТ
эксперт
автоматизация сервис трейдинг

SIEMENS

- Контроллеры: Logo!, Simatic: S7-200, S7-1200, S7-300, S7-400, S7-1500;
- Электродвигатели;
- Преобразователи частоты (Micromaster, Sinamics);
- Устройства плавного пуска Sicostart;
- Низковольтная коммутационная техника;
- Микропроцессорная релейная защита;
- Контрольно измерительные приборы (расходомеры, уровнемеры, датчики давления, датчики проводимости, PH – метры);
- Контроллеры для систем отопления: Synco 100, Sigmagyr, Synco 700;
- Устройства для измерения, ограничения и управления: датчики, температуры, влажности, конденсата, давления, протока, качества воздуха;
- Контроллеры для комнатной автоматизации: RXA, RXL, RXB, RXC;
- Комнатные термостаты;
- Клапаны и приводы;
- Приводы воздушных заслонок – OPENAIR.

220114, Республика Беларусь,
г.Минск, ул.Филимонова, д.25б
e-mail: ast@ast-expert.by
web: www.ast-expert.by

т.: +375(17)389 70 70
389 70 71
389 70 72
ф.:+375(17)267 42 29