АНО «Институт логики, когнитологии и развития личности»
ALT Linux

Одиннадцатая конференция разработчиков свободных программ

Калуга, 26–28 сентября 2014 года

Тезисы докладов

Москва,
Альт Линукс,
2014

УДК 004.91
ББК 32.97

Десятая конференция разработчиков свободных программ: Тезисы докладов / Калуга, 26–28 сентября 2014 года. М.: Альт Линукс, 2014. — 60 с. : ил.

В книге собраны тезисы докладов, одобренных Программным комитетом одиннадцатой конференции разработчиков свободных программ.

ISBN 978-5-905167-17-1

Программа конференции

26 сентября

14.00-15.00 Регистрация участников

Дневное заседание
15.00–18.00

15.00–15.45 А. Е. Новодворский

Импортозамещение и роль в нем свободного программного обеспечения 6

15.45-16.30 Дискуссия

16.30–17.00 Кофе-пауза

17.00–18.00 А. Г. Боковой

Жизнь в пузыре (в двух частях) 9

27 сентября

Утреннее заседание
10.00–13.30

10.00–10.30 Александр Рябиков, Сергей Середа

Коммерциализация СПО под GPL лицензией 12

10.30–11.00 С. М. Выгинный, В. И. Гуржиев

Ограничения в проекте национальной ОС и практический опыт преодоления таких ограничений 17

11.00–11.30 Д. Д. Державин

Робототехнические приложения на платформе ALT Linux .. 19

11.30–12.00 Кофе-пауза

Дневное заседание
15.00–18.00

28 сентября

Утреннее заседание
10.00–12.00

10.00–10.30 Вадим Кузнецов, Николай Крючков

10.30–11.00 А. В. Ненахов

11.00–11.30 Резерв

11.30–12.00 Кофе-пауза

Алексей Новодворский
Москва, Альт Линукс

Импортозамещение и роль в нем свободного программного обеспечения

Аннотация

Мы рассматриваем далее импортозамещение программного обеспечения как средство развития отрасли в условиях необходимости обеспечения технологической независимости.

Положение дел

До сих пор приоритет отдавался поддержке экспорта разработанного в России ПО и здесь достигнуты впечатляющие результаты. Однако, объем экспорта готовых продуктов оценивается в 30% от общего экспорта ПО, причем права на эти продукты зачастую принадлежат зарубежным юридическим лицам. (`http://www.tadviser.ru/index.php/`Статья:Экспорт_ПО_из_России)

Поддержка отечественных производителей готовых программных продуктов в госзакупках практически отсутствует, в отличие от США: `http://en.wikipedia.org/wiki/Buy_American_Act` и других стран: `http://kapital-rus.ru/articles/article/180344#glava2`

Риски реализации программы импортозамещения

1. Главным риском является изоляция от мировых процессов разработки программного обеспечения и, как следствие, рост технологического отставания.
2. Рост использования нелегальных копий зарубежного ПО, который ведет к отсутствию поддержки производителя.

Определение отечественного ПО

Это определение, закрепленное в нормативно-правовых актах, является основой программы имопортозамещения. Основанное на принадлежности исключительных прав российским юридическим и фи-

зическим лицам, оно должно быть легко проверяемым и комактным, например http://www.cnews.ru/news/top/index.shtml?2014/07/15/579606 , согласованное IT-ассоциациями. Включение в него ПО от производителей стран таможенного союза важно, но до создания единого реестра физлиц ТС приведет к проблемам при реализации программы импортозамещения.

Преференции отечественным производителям ПО при госзакупках

Как было сказано выше, преференции отечественным производителям ПО являются обычной практикой в развитых странах и должны быть изучены. Равенство условий для импортных продуктов возможно только на условиях взаимности.

В госзакупках, не затрагивающих непосредственно вопросы национальной безопасности, следует избегать запретительных мер для иностранных продуктов.

Свободное программное обеспечение в импортозамещении

Использование свободного программного обеспечения (ГОСТ Р 54593–2011) позволяет в значительной степени избежать обозначенных выше рисков импортозамещения. Свободное ПО позволяет работать в русле передовых технологий и избежать многих проблем с легальностью распространения и использования. Вместе с тем, сведение импортозамещения к преимущественному использованию СПО и создание особых преференций для его производителей было бы ошибкой. Конвергенция двух ранее несмешивающихся направлением стала очевидной:

1. В последние годы СПО перестало быть альтернативным направлением разработки. Сейчас практически нет сколько-нибудь крупных проектов, которые бы использовали только СПО или только проприетарное ПО, — это позволяет снизить стоимость разработки. Неправильно говорить про отдельную «индустрию СПО».
2. Распространение некопилефтных (т. е. не обязывающих открывать код производных продуктов) лицензий, особенно Apache v.2, позволяет создавать проприетарные продукты на основе

свободных, избегая проблем с патентным правом. Обычно при этом производителю выгодно часть кода отдавать обратно в свободный проект для влияния на его развитие.

Использование СПО может способствовать решению проблем технологической независимости, но само по себе не решает их, а иногда и порождает новые риски:

1. Простота и дешевизна копирования и ребрендинга продуктов СПО ведет к появлению многочисленных клонов, которые зачастую не обеспечиваются процессами развития разработки и поддержки. Новые версии продуктов-клонов не разрабатываются отечественными разработчиками, а копируются. В этих условиях не приходится говорить о технологической независимости вообще. Продукты-клоны весьма дешевы, а потому вытесняют с рынка отечественные разработки со своими технологиями и значительной долей оригинального кода, свободные или проприетарные.

2. Успешность разработок СПО прямо зависит от интеграции в мировое сообщество разработчиков СПО. К сожалению, число активных российских участников международных СПО-проектов в последние годы по крайней мере не растет, что не дает возможности говорить об их существенном влиянии их на направление развития таких проектов.

Решение двух вышеперечисленных проблем необходимо предусмотреть в программе импортозамещения.

Быстрое начало

Успех любого значительного начинания, затрагивающего большую отрасль, пользователями продуктов которой являются практически все граждане страны, зависят во многом от достаточно быстрых и видимых результатов. Планирование лишь на 3-7 лет вперед вызывает серьезные сомнения в успешности всего начинания.

1. В России имеются значительный задел реально используемых продуктов отечественных фирм, позволяющий начинать с их продвижения и использования почти во многих сферах применения. Отметим проекты:

- Диасофт в банковской сфере: http://banks.cnews.ru/top/2014/08/20/100procentno_rossiyskaya_programmnaya_platforma_nacelilas_na_banki_583303
- масштабные внедрения в медицинских учреждениях России: http://www.cnews.ru/reviews/index.shtml?2013/10/10/545843
- перевод СМЭВ на стек Apache: http://corp.cnews.ru/news/top/index.shtml?2014/08/21/583565

Это далеко не полный список.

2. Оперативное создание нормативно-правовой базы для преференций отечественным программным продуктам в госзакупках, позволит его производителям получить быструю отдачу от продаж, рефинансировать разработку и создать хороший фон для инвесторов.

Александр Боковой
Эспоо, Финляндия, Red Hat Ltd.
Проект: Samba Team http://samba.org, http://cwrap.org

Жизнь в пузыре (в двух частях)

Тестирование сложных сетевых программных комплексов требует построения воспроизводимых тестовых стендов. Тесты также представляют собой относительно простой способ знакомства с кодом проекта, вариантами его внедрения, корректностью документации и т.д. Многие свободные проекты выиграли бы от снижения барьера для входа новых разработчиков, если бы могли предоставить тестовую среду, которая легко собирается, желательно в автоматическом режиме. К сожалению, в большинстве случаев у свободных проектов, реализующих сложные сетевые решения, если и присутствует инфраструктура для тестирования, то она недоступна новичкам или не может быть развернута локально.

cwrap.org – это проект, выросший из десятилетнего опыта создания средств автоматического тестирования в Samba Team. Он позволяет развернуть в рамках одного пользователя сложную сетевую

инфраструктуру, включающую в себя несколько контроллеров доменов, клиентских «машин», сервер DNS, сервер Kerberos KDC, и другие компоненты, необходимые для исполнения нескольких тысяч тестируемых конфигураций Samba. Для разработчика запуск тестовой среды выглядит как простой `make test` и не требует ни средств виртуализации, ни каких-либо дополнительных привилегий. Более того, средства `cwrap.org` позволяют проводить тестирование на разных платформах, а не только в рамках современных дистрибутивов GNU/Linux.

Для изолированного тестирования клиент-серверных приложений необходимо моделировать сетевую инфраструктуру, разрешение имен и числовых идентификаторов пользователей и групп, преобразование имен узлов в IP-адреса, сохранение прав доступа к файлам с учетом привилегий исполняемой среды и многое другое. cwrap обеспечивает эти возможности, представляя динамические библиотеки, которые загружаются в адресное пространство приложения средствами системы, при этом корректно поддерживая и сложные многопоточные приложения.

На текущий момент в `cwrap.org` реализованы следующие «обертки»:

- `socket_wrapper` выполняет все сетевые операции, используя коммуникационные возможности доменных сокетов UNIX. Поддерживаются сокеты и режимы адресации семейств IPv4 и IPv6, а передаваемые по имитируемой сети сообщения автоматически записываются в формате PCAP, используемом распространенными средствами анализа сетевого трафика, например, Wireshark.

- `nss_wrapper` позволяет переопределить информацию о пользователях и группах, которые будут видны приложению, а также имена сетевых узлов. Помимо статических файлов пользователей и групп, `nss_wrapper` позволяет загружать уже имеющийся модуль интерфейса `NSS` в `glibc` и комбинировать результаты использования модуля с предопределенными данными из файлов.

- `uid_wrapper` переопределяет поведение системы при запросе приложением привилегированных операций. Приложение может переключать контекст исполнения, меняя пользователя, под которым выполняются действия, а `uid_wrapper` обеспечит в том числе и корректную обработку прямых системных вызо-

вов функций ядра (`syscalls`) в контексте конкретного потока исполнения.

- `resolv_wrapper` позволяет переопределить поведение сетевого стека при разрешении имен сетевых узлов, включая перенаправление запросов на другие сервера DNS, в том числе и исполняемые под контролем `socket_wrapper`.

В рамках проекта Samba «обертки» `cwrap.org` используются для полномасштабного тестирования приложений — от традиционного файл-сервера до реализации контроллера домена Active Directory, включающего в себя одновременную работу сервера samba, центра раздачи ключей Kerberos, сервера DNS, сервера LDAP и разнообразных клиентских компонент. Каждый тестируемый сценарий получает собственную виртуальную среду исполнения и не зависит от других сценариев, что позволяет параллельное исполнение. В рамках существующей программы тестирования выполняется около 4500 разных сценариев, а общее время выполнения `make test` достигает двух часов. Каждое исправление, попадающее в git-репозитарий проекта Samba, обязательно проходит полный цикл тестирования.

Помимо поддержания тестовой среды, cwrap.org применяется и для решения более сложной задачи. Архитектура Active Directory подразумевает, что различные компоненты контроллера домена (сервер LDAP, сервер Kerberos KDC, сервер DNS, сервер SMB и LSA) имеют единое состояние клиентского подключения. Клиент может инициировать соединение по протоколу LSA, а затем переключиться на протокол LDAP и запросить результат исполнения одной из операций протокола LSA. Некоторые из критических для домена операций намеренно реализуются таким образом, например, установление доверительных отношений между доменами в лесу Active Directory или доверительных отношений между двумя разными лесами Active Directory.

Необходимость общего доступа к состоянию клиентского соединения вынудила в свое время Samba Team интегрировать Kerberos KDC, сервер DNS и сервер LDAP. К сожалению, одна из наиболее активно развиваемых реализаций Kerberos, от MIT Kerberos Consortium, не поддерживает режим встраивания сервера в другие приложения и использовать Samba AD DC с MIT Kerberos долгое время было невозможно. Это приводило к разного рода проблемам при интеграции Samba AD DC в дистрибутивы GNU/Linux, поскольку в рам-

ках дистрибутива нельзя обеспечить взаимозаменяемость библиотек Kerberos — они совместимы на уровне сетевых протоколов, но не ABI, а также имеют различия в формате представления данных на диске.

В течение 2013-2014 годов компанией Red Hat в рамках интеграции Samba AD DC в дистрибутивы Fedora, Red Hat Enterprise Linux и CentOS, была выполнена фундаментальная работа над переводом Samba AD DC на использование MIT Kerberos. Ключевым компонентом этой системы является использование cwrap.org для обмена состоянием клиентских соединений между сервером KDC и остальными компонентами Samba AD DC. Принципиальная работоспособность этой архитектуры была продемонстрирована на конференции SambaXP в мае 2014 года, а в сентябре 2014 Samba Team выполнит тестирование получившегося решения в рамках Interoperability Lab, организуемого Microsoft.

Александр Рябиков, Сергей Середа, к.э.н.
Москва, ОАО ИнфоТеКС, Фонд «Адемпиере»

Проект: Фонд поддержки и развития делового свободного программного обеспечения «Адемпиере» http://www.adempiere.ru

Коммерциализация СПО под GPL лицензией

Аннотация

Отсутствие нормального способа коммерциализации свободного программного обеспечения значительно ограничивает привлечение инвестиций в развитие свободных программных проектов с вирусными лицензиями, т. к. нормальные способы коммерциализации имеются только у владельцев исключительных прав на исходный код. Все остальные разработчики вынуждены или заниматься не профильным бизнесом или использовать приемы обхода GPL лицензии, что бы работала бизнес модель проприетарного программного обеспечения, основанная на монопольном владении объектом авторского права. Но в последнем случае не получает развития свободный проект. Предлагается новый способ коммерциализации доработок свободного программного обеспечения, разрабатываемых под «вирусными» GPL лицензиями, который позволит инвесторам вернут вложенные средства, а сообществу получить ощутимый вклад в развитии свободных проектов.

Как известно, «свободные» лицензии дают конечным пользователям полную свободу, в том числе и свободу распространять конечный программный продукт наравне с разработчиком. Принято считать, что невозможно зарабатывать деньги непосредственно на доработке и распространении свободного программного обеспечения, распространяемого на условиях GPL или аналогичных «вирусных» лицензий. Описываемые в литературе и применяемые на практике бизнес-модели практически безусловно предполагают бесплатное (или за символическую плату) распространение таких программ. В противоположность этому, наличие «традиционной» (на жаргоне называемой «проприетарной») лицензии позволяет зарабатывать непосредственно на продаже программного обеспечения.

Обладателю исключительных прав ещё доступна модель двойного лицензирования, подразумевающая «несвободную» коммерческую лицензию на ПО для бизнес-заказчиков и СПО лицензию для представителей сообщества. Для всех же вторичных проектов (так называемых «форков») подобная возможность исключена. Любые доработки кода, выпущенного под GPL, могут быть только под этой же или совместимой «вирусной» лицензией.

С учетом этого, считается, что зарабатывать на Free Software, не являясь обладателем прав на оригинальный продукт, можно либо оказывая сервисные услуги (сопровождение, документирование, обучение и т.п.), либо компонуя его с собственными разработками с закрытым исходным кодом, способом, который допускается используемой свободной лицензией.

Поэтому, главная проблема с коммерциализацией GPL программ заключается в том, что первый же покупатель, который получил исходники купленной программы, вправе сам стать распространителем данного ПО, в том числе, выложив полученное ПО в свободный бесплатный доступ, что безусловно создаст проблемы с возвратом финансов, вложенных в разработку такого программного продукта.

Описываемый ниже способ коммерциализации доработок свободного ПО позволит в определенной мере изменить сложившуюся ситуацию. Предлагаемая бизнес-модель предназначена, в основном для B2B разработчиков, развивающих уже существующие свободные программные продукты, но не владеющих исключительным правом на сам продукт (т.е. наиболее часто встречающаяся ситуация).

Суть бизнес-модели сводится к созданию условий, аналогичных «традиционномой» модели распространения, т.е. к «продаже» дора-

ботанного продукта сразу нескольким покупателям, что даст возможность окупить понесенные затраты на доработку СПО за счет их многократной продажи. Такой эффект достигается за счет создания временного лага между началом продаж доработанного программного продукта и моментом его размещения в свободном доступе для бесплатной загрузки, что позволяет разработчику стать, на какое-то время, единственным продавцом, у кого эти доработки можно будет приобрести.

Необходимо отметить, что в предлагаемой бизнес-модели не происходит ни изменения, ни нарушения условий «вирусного» лицензионного соглашения — эффективное ограничение распространения продукта определяется договором между хозяйствующими субъектами и вытекает из процесса выполнения заказных программных разработок.

Коммерциализация GPL на основе условий лицензии

Временной лаг между передачей программного продукта заказчику и появлением у него права на дальнейшее распространение купленного ПО создаётся за счёт превращения покупателя в (со)разработчика и подписания с ним соглашения о нераспространении.

Схема взаимодействия к компанией-пользователем:

1. Компания разработчик не выкладывает продукт в свободный доступ, т.к. GPL этого не требует `http://www.gnu.org/licenses/gpl-faq.en.html#DoesTheGPLRequireAvailabilityToPublic`

2. Разработчик «продает» компании-пользователю исходный продукт без своих доработок на основании того, что GPL допускает платное распространение программ `http://www.gnu.org/licenses/gpl-faq.en.html#DoesTheGPLAllowMoney`

3. Одновременно, разработчик нанимает компанию-пользователя для выполнения работ, связанных с новым кодом, например, для тестирования. В этом случае возникает законная возможность ограничить распространение тестируемого кода на время действия соответствующего договора, т. к. GPL позволяет запретить наемному работнику распространять результаты за-

казной разработки `http://www.gnu.org/licenses/gpl-faq.en.html#DevelopChangesUnderNDA`

4. Время действия такого договора и будет определять временной лаг между началом распространения GPL доработок и моментом их их возможного появления в свободном доступе.

5. Разработчик может одновременно работать по такой схеме сразу с несколькими заинтересованными компаниями.

Ричард Столлман не в восторге от описанного выше способа, но нарушение лицензии GPL будет только в том случае, если работа пользователя будет фиктивной. Другими словами, пользователь должен по настоящему работать в соответствии с контрактными обязательствами и эта работа должна реально оплачиваться.

Коммерциализация GPL на основе договорного права

Второй способ создать временной лаг между моментом передачи программного продукта заказчику и появлением у него права на его дальнейшее распространение возможен за счёт того, что права на созданную по заказу программу для ЭВМ и, соответственно, возможность её дальнейшего распространения возникают только после полного исполнения своих обязательств сторонами договора на доработку ПО.

Схема взаимодействия к компанией-пользователем:

1. Разработчик заключает с компанией-пользователем договор на доработку исходного продукта с GPL лицензией.

2. В соответствии с действующим законодательством (статья *712. Право подрядчика на удержание; статья 1296. Программы для ЭВМ и базы данных, созданные по заказу*), в этом случае существует законная возможность ограничить распространение дорабатываемого кода на время действия соответствующего подрядного договора.

3. Договор подразумевает обязательное наличие этапа опытной или (и) опытно-промышленной эксплуатации, что и будет временным лагом между началом распространения доработок СПО и моментом их появления в свободном доступе. т.к. это возможно только после перехода права собственности на ПО.

4. Разработчик может одновременно работать по такой схеме сразу с несколькими заинтересованными компаниями.

Описанная конструкция не является каким-то искусственным построением. Наоборот, она является описанием вполне реального процесса заказной разработки корпоративного ПО, который завершается только после проведения опытной или опытно-промышленной эксплуатации разработок (с их полным сопровождением все это время) и передачи их в промышленную эксплуатацию.

Мы обсуждали с Ричардом Столлманом и этот способ коммерциализации. По его мнению, предложенная схема нарушает GPLv3, где в явном виде прописано, что лицензия имеет приоритет над контрактными обязательствами. С этим мнением не согласны юристы, но кто из них прав может решить только судебная практика.

В заключении

Наложение любых ограничений на распространение СПО всегда воспринимается «в штыки» сообществом, даже в том случае, если ограничения полностью соответствуют условиям лицензии и не противоречат законодательству.

С другой стороны, отсутствия нормального способа коммерциализации СПО значительно ограничивает привлечение инвестиций в развитие свободных программных проектов, т. к. нормальные способы коммерциализации имеются только у владельцев исключительных прав на исходный код. Все остальные разработчики вынуждены или заниматься непрофильным бизнесом или использовать приемы обхода GPL лицензии, что бы работала традиционная бизнес модель, основанная на монопольном владении объектом авторского права. Но в этом случае не развивается дорабатываемый свободный проект.

Предложенный способ коммерциализации доработок СПО с «вирусными» лицензиями устраняет описанные выше недостатки. Конечно, первое время код будет отсутствовать в свободном доступе, но через какое то время он все равно станет доступен для всех. И в результате, и инвесторы вернут вложенные средства и сообщество получит ощутимый вклад в развитие свободных проектов.

Фонд поддержки и развития делового свободного программного обеспечения «Адемпиере» оказывает финансовую, юридическую и организационную помощь разработчикам СПО, в том числе в приме-

нении данной методики коммерциализации доработок проектов под GPL лицензией, а так же ищет спонсоров, готовых к инвестициям в свободное программное обеспечение.

Выгинный С.М., Гуржиев В.И.
Москва, ООО «НПП» Гарант-Сервис-Университет
Проект: Система ГАРАНТ www.garant.ru

Ограничения в проекте национальной ОС и практический опыт преодоления таких ограничений

Аннотация

Компания Гарант, как разработчик проприетарного ПО, со многими органами государственными власти работает по freeware модели. Наш опыт работы с ОГВ показывает, что тип лицензии, будь то свободное ПО, freeware или коммерческая лицензия, имеет значение только на начальных этапах внедрения. О типе лицензии знают только лица, принимающие решения и системные администраторы, обслуживающие информационные системы организации. Но с ПО работают конечные пользователи и их обучение — это задача, которая ложится или на системных администраторов, или на компанию, внедряющую ПО. Выступление можно рассматривать как практический опыт поддержки конечных пользователей в незнакомом для них ПО в рамках централизованного внедрения этого ПО в ОГВ и бюджетных организациях по всей стране. Опыт такого масштаба с одной стороны отсутствует в области свободного програмного обеспечения, с другой стороны, может быть использована разработчиками свободного ПО. При этом технология поддержки выступает способом монетизации ПО, распространяемым по моделям свободного ПО и freeware.

На сегодняшний день в Российской Федерации складывается благоприятная обстановка к развитию внутреннего рынка программного обеспечения. Уже не первый год руководство страны задумывается над созданием национальной ОС и программного обеспечения для неё. В последнее время, на фоне санкционной борьбы, тема становится очень актуальной. Естественно, что в сложившейся внешнеполитической обстановке взоры обращены к рынку свободного ПО. Разработка операционных систем для ПК, серверных и мобильных платформ, пакет офисных приложений, СУБД, систем проектирования и систем управления предприятий — это государственные приоритеты

для замены привычного, читай проприетарного ПО [2]. При наличии политической воли реализовать такой масштабный проект, даже при многомиллиардной поддержке государства, можно только на основе существующего свободного ПО.

Однако, создание свободного ПО и общенациональное внедрение — это не одно и то же. *«На любом этапе большого ИТ-проекта должны быть компании, готовые квалифицировано решить любые проблемы с внедрением. В идеале у них есть план внедрения с первого до последнего шага, предусматривающий все возможные проблемы. Иногда подобные услуги стоят очень дорого, но «экосистема» построенная вокруг Microsoft и другого коммерческого ПО — это и есть ИТ-бизнес... Именно поэтому, Microsoft, Oracle, IBM, SAP и т.д. — с удовольствием рассказывают о своих завершенных проектах, в том числе для российских госучреждений и госпредприятий. А в области национального ИТ мы слышим исключительно PR начальных проектов».* [1]

При успешной реализации проекта по созданию национальной ОС и пакетов прикладного программного обеспечения, в полный рост встанет проблема внедрения этого ПО. Обычный пользователь консервативен. Изменение привычный рабочей среды — это большая проблема. При работе на высококонкурентном рынке наша компания очень часто сталкивается с необходимостью переобучения пользователей, когда внедрение нового ПО происходит по указанию сверху, вопреки желанию конечного пользователя и/или системного администратора.

Мы используем следующие способы поддержки и обучения пользователей:

Онлайн-обучение — обучение с помощью спутниковых и интернет-семинаров.

Оффлайн-поддержка — взаимодействие с конечным пользователем на его рабочем месте с помощью специалистов региональной партнерской сети. Сейчас личной поддержкой пользователей занимается 1500 специалистов в 230 партнерских организациях.

На некотором этапе своего развития мы решили попробовать таким способом обучать пользователей не только своему программному обеспечению, но и чужому. Эксперимент удался и мы обучили более 2000 пользователей работе на площадке государственных закупок Сбербанк-АСТ.

С другой стороны, очень часто пользователь не заинтересован в своем обучении навязанному ему ПО. Для того чтобы его заинтересовать мы разрабатываем программы нематериальной мотивации:

- сертификация пользователей — это положительно сказывается на самооценке пользователя и его резюме;
- программы повышения квалификации — очень востребованы, например, в бухгалтерской среде.

Внедрение даже очень хорошего программного обеспечения всё равно требует взаимодействия разработчика и конечного пользователя. На территории нашей огромной страны такая поддержка должна опираться на распределенную региональную сеть. Мы предлагаем разработчикам свободного ПО объединить усилия в импортозамещении программного обеспечения с целью достижения рыночных результатов.

Литература

[1] Бычков Валерий, Почему в России нет национального дистрибутива Linux, `http://smartsourcing.ru/blogs/poleznye_tehnologii_i_produkty/2654`

[2] Павел Кантышев, Елизавета Серьгина, Анастасия Голицына, Минкомсвязи планирует обложить производителей софта 10%-ным налогом с продаж, `http://www.vedomosti.ru/tech/news/33391241/softovyj-sbor`

Дмитрий Державин
Санкт-Петербург, СПбФ ОАО «НИЦЭВТ»

Робототехнические приложения на платформе ALT Linux

Аннотация

В настоящее время на платформе ALT Linux доступен набор ПО, позволяющий использовать дистрибутивы на базе репозитория Сизиф в качестве полноценной рабочей среды для решения ряда задач в области робототехники.

В частности: доступен полный набор инструментов для печати на 3D-принтерах с открытой аппаратной частью, среда разработки Arduino, инструментарий проекта УМКИ.

Коллекция робототехнического ПО в репозитории Сизиф далеко не полна: в ближайшее время планируется, в частности, добавить наземные станции управления беспилотными летательными аппаратами и расширить список приложений для трёхмерной печати на принтерах RepRap.

Чтобы сделать знакомство с робототехническим ПО для Linux более удобным, подготовлен специализированный образ загрузочного диска, предназначенный для развёртывания на компакт-диски и портативные накопители данных.

На платформе ALT Linux поддерживался и поддерживается набор приложений разной степени специализации, пригодных для решения задач в области робототехники. Примерами таких приложений могут служить: Blender — программа для моделирования трёхмерных объектов — от мультипликационных персонажей до деталей механизмов; Arduino — среда разработки для открытой вычислительной платформы, широко применяемой в робототехнике; УМКИ — станция управления учебными роботами.

При этом целью данного доклада является скорее не составление подробного списка робототехнических приложений, а привлечение более пристального внимания пользователей ОСПО к робототехнике, а также очерчивание круга робототехнических задач, для решения которых платформа ALT Linux сейчас уже пригодна, и задач, для решения которых хотелось бы иметь возможность её применить в ближайшем будущем.

Дело в том, что, к сожалению, при современном фантастическом уровне доступности программного и аппаратного обеспечения школьники всё ещё не печатают на уроках труда собственноручно сконструированные и смоделированные детали.

Одним из шагов на пути к решению этой проблемы мог бы стать дистрибутив Linux, предоставляющий возможность «из коробки», например, залить в трёхмерный принтер, построенный на открытой аппаратной

платформе, свежую прошивку; смоделировать деталь для печати; перевести модель в траектории печатающей головки, и тут же распечатать.

Современный трёхмерный принтер — типичный станок с ЧПУ — не единственный пример робота, которого относительно просто построить в условиях домашней или школьной мастерской. Другой типичный пример — беспилотные самолёты, вертолёты и автомобили, базирующиеся на открытых программных и аппаратных платформах.

Их программная часть также включает в себя прошивку, интерфейс непосредственного управления, программный интерфейс и среду разработки. При этом открытых аппаратных платформ, и, соответственно, сред разработки не так уж и много. Таким образом, имея поддержку Arduino, мы уже обеспечиваем возможность построения широкого спектра роботов — от шестиногих шагающих машин до систем управления «умным домом».

В данный момент набор робототехнических приложений, находящихся в репозитории Сизиф, вполне позволяет использовать дистрибутивы ALT Linux для знакомства с предметной областью. В ознакомительных целях подготовлен специализированный минидистрибутив, включающий в себя все приложения репозитория Сизиф, так или иначе относящиеся к робототехнике.

Михеев Андрей Геннадьевич, Николаев Александр Викторович, Панкин Владислав Вячеславович, Пятецкий Валерий Ефимович

Москва, Консалтинговая группа РУНА, Казань, компания «Центр», Тольятти, компания «Лаборатория Свободных Решений», Москва, НИТУ «МИСиС»

Проект: АИС «ГОСУСЛУГИ». http://gosuslugi.tp-npp.ru/

Свободная платформа автоматизации государственных и муниципальных услуг с открытым исходным кодом (типовое решение для муниципалитетов)

Аннотация

Платформа предназначена для поддержки деятельности органов власти в части предоставления государственных и муниципальных

услуг на территории Российской Федерации, дает возможность автоматизировать административные регламенты предоставления государственных и муниципальных услуг, включая автоматизацию межведомственного взаимодействия с использованием Системы межведомственного электронного взаимодействия. Платформа реализована на базе свободно распространяемого программного обеспечения, имеет web-интерфейс и работает под управлением различных операционных систем (Linux, Windows и пр.) и баз данных (PostgreSQL, Oracle и др.)

Концепция создания платформы

АИС «ГОСУСЛУГИ» представляет собой автоматизированную информационную систему поддержки деятельности органов власти в части предоставления государственных и муниципальных услуг (функций) на территории Российской Федерации. АИС «ГОСУСЛУГИ» предназначена для автоматизации процессов предоставления государственных и муниципальных услуг, включая автоматизацию межведомственного взаимодействия с использованием Системы межведомственного электронного взаимодействия.

АИС «ГОСУСЛУГИ» реализована на базе технологических платформ свободно распространяемого программного обеспечения, имеет web-интерфейс и работает под управлением различных операционных систем (Linux, Windows и пр.) и баз данных (PostgreSQL, Oracle и др.)

Состав платформы(запланированный)

- Система управления бизнес-процессами и административными регламентами RunaWFE
- Прикладные подсистемы
- Коннекторы к сервисам ведомств
- Коннекторы к сервисам СМЭВ
- Электронная подпись (ЭЦП)
- Шаблоны административных регламентов

Краткое описание существующих функциональных модулей

- Портал взаимодействия с заявителем и Личным кабинетом — WEB-интерфейс заявителя, непосредственно контактирующего

с системой. Позволяет проводить аутентификацию и авторизацию заявителя, в личном кабинете создавать новые обращения и отслеживать их состояние.

- FrontOffice с АРМ универсального специалиста — WEB-интерфейс оператора, осуществляющего непосредственный контакт с заявителем. Позволяет автоматизировать работу оператора по приему заявителей.
- BackOffice — WEB-интерфейс оператора, осуществляющего исполнение процессов. Позволяет автоматизировать работу оператора по исполнению процессов оказания услуг.
- Панель администрирования — WEB-интерфейс администратора. Позволяет автоматизировать работу администратора по настройке системы.
- Мобильная платформа — набор интерфейсов (WEB-интерфейс, мобильное приложение и API), реализующих взаимодействие заявителей и операторов с системой. Позволяет в качестве аппаратного средства взаимодействия с системой использовать мобильную платформу.
- Экспертная подсистема — подсистема поддержки принятия решений. Комплекс методов и инструментов позволяющих упро-

стить пользователю и заявителю процесс принятия решений при работе с системой.

- Подсистема поддержки пользователей — подсистема обеспечения комплексной поддержки пользователей. Позволяет реализовывать механизмы технической и консультационной поддержки пользователей, производить обучение и информировать об изменениях, новостях и событиях.
- Подсистема исполнения услуг — подсистема автоматизированного исполнения основных административно-управленческих процессов. Позволяет управлять обращениями заявителей, а также исполнять регламенты услуг.
- Адаптерная платформа — подсистема универсальных адаптеров. Позволяет унифицировать состав адаптеров и атрибутивный состав данных, необходимый для взаимодействия подсистем.
- Подсистема формирования интерактивных форм — подсистема, обеспечивающая автоматизированное создание интерактивных форм. Позволяет реализовать механизмы визуального конструирования форм, а также их подключение и использование в других модулях или подсистемах.
- Электронный архив — подсистема хранения электронных образов документов. Позволяет сохранять электронные образы документов и их описания. Поддерживает версионность документов, а также ведет историю изменений. Позволяет реализовывать функции поиска документов и разграничения прав доступа.
- Аналитическая подсистема. Включает в себя: модуль консолидированная отчетность и аналитика и построитель отчетных форм. Обеспечивает сбор и хранение показателей работы системы, а также проведение аналитических исследований и подготовку отчетных материалов.
- Подсистема администрирования Включает в себя: модуль управления правами доступа, модуль журналирования, обеспечение юридической значимости (ЭЦП) и др. осуществление гибкой настройки и администрирования системы.
- Подсистема ведения НСИ и информационных реестров Включает в себя: НСИ, модуль ведения информационных реестров,

реестр пользователей, реестр сервисов, реестр заявителей, реестр услуг, реестр обращений. Служит для создания, ведения и хранения информационных и справочных материалов и реестров.

- Сервисная шина — шина, реализующая взаимодействие с внешними системами. Позволяет настраивать и публиковать сервисы, а также реализовывать механизмы информационного обмена с внешними системами.

Варианты внедрения муниципалитетами

1. Единый облачный сервис для муниципалитетов, развернутый в ЦОДе региона.
2. Отдельная (локальная) инсталляция на сервере муниципалитета Муниципалитет самостоятельно управляет и обслуживает систему

Последовательность действий по внедрению в муниципалитете

1. Установка платформы на отдельный физический или виртуальный сервер
2. Настройка интеграции с Active Directory (LDAP) для обеспечения аутентификации пользователей по доменным паролям
3. Настройка системы связи со СМЭВ
4. В случае применения одной установки на несколько юридических лиц: настройка системы удаленного подписывания запросов электронной подписью органа власти на оборудовании отдельных юридических лиц, пользующихся системой
5. Создание процессов оказания услуг на основе примеров (этих примеров тоже еще нет в платформе)
6. Настройка прав доступа пользователей
7. Регистрация системы в СМЭВ через оператора регионального сегмента СМЭВ (здесь в дальнейшем будут размещены шаблоны типовых заявок)

8. Настройка интеграции с региональным порталом государственных и муниципальных услуг (это индивидуально, в каждом регионе может быть по-разному)

Примеры прикладных подсистем

1. Информационная система «Реестр СМСП». Предназначена для создания единого Реестра получателей государственной поддержки среди субъектов малого и среднего предпринимательства в рамках субъекта РФ, для автоматизации получения услуги по подаче заявки на оказание поддержки субъекту малого и среднего предпринимательства, а также для аналитической обработки данных, хранящихся в Реестре. Промышленная эксплуатация Реестра начата в мае 2013 г. в Республике Татарстан. Услуга по подаче заявки на оказание поддержки выведена на Портал Государственных и муниципальных услуг РТ `uslugi.tatar.ru`.
2. Компонент «АРМ универсального специалиста», входящий в состав FrontOffice Свободной платформой. Компонент предназначен для автоматизации работ оператора в части работы с заявителем: регистрации обращений, обработки документов, управления процессом оказания услуг, а также обеспечивает оператору информационную поддержку. Представляет собой оптимизированный для оператора WEB-интерфейс. Компонент был использован в системах АИС МФЦ Московской области и АИС ОПС Почты России, предназначеных для автоматизации основных административно-управленческих процессов, протекающих в рамках предоставления государственных и муниципальных услуг.

Ссылки

1. Ссылка на сайт проекта: `http://gosuslugi.tp-npp.ru`
2. Национальная Программная Платформа: `http://tp-npp.ru/`
3. Проект RunaWFE: `http://runawfe.org/rus`
4. Сайт компании Лаборатория Свободных Решений: `http://osslabs.ru/`
5. Сайт компании Центр: `http://cg.ru/`

Игорь Власенко
Киев, ALT Linux

Проект: Облачно-дружественная распределенная инфраструктура для сервисов автоматизации ALT Linux Team
http://www.altlinux.org/Категория:Packaging_Automation

Кластер автоматизации сопровождения пакетов год спустя

Аннотация

По итогам года использование облачного кластера автоматизированного импорта, сборки и тестирования программных пакетов позволило повысить продуктивность работы майнтайнера более чем в 100 раз и внести весомый вклад в развитие дистрибутивной платформы ALT Linux Sisyphus.

Появившиеся в последнее время для проприетарных операционных систем «магазины приложений» позволили сделать в этих системах установку приложений такой же, если не более легкой, как в дистрибутивах Linux. При этом «магазины приложений» выигрывают по охвату пакетной базы. «Простой пользователь» практически не сталкивается с необходимостью устанавливать приложения из сторонних источников.

У современных дистрибутивов Linux ситуация с охватом пакетной базы хуже. В популярных дистрибутивах упакованы наиболее популярные приложения, «10% приложений, которые нужны в 90% случаев». Однако в случае, если пользователю нужно что-то, выходящее за пределы стандартного набора приложений, ему потребуется собирать их самостоятельно, что требует хотя бы начальных навыков в программировании и автоматически вычеркивает девять десятых пользователей.

Создание дистрибутива с полной пакетной базой стало бы важным шагом в продвижении Свободного Программного Обеспечения в направлении простых пользователей. Однако полная упаковка имеющейся пакетной базы традиционными методами кооперации волонтеров недостижима. Нереалистично высокое число волонтеров, требуемое для такой задачи, не соответствует имеющейся социальной базе. Кроме того, явления вроде «закона Брукса» в больших коллективах

разрушают эффективность совместной работы, даже если бы социальная база для добровольцев была бы достаточно большой.

Разработка средств автоматизации сопровождения пакетов стала попыткой воплотить мечту о создании дистрибутивной платформы с полной пакетной базой программными средствами. Венцом этой деятельности стало создание облачного кластера автоматизации сопровождения пакетов. Развернутый в августе 2013 года, кластер автоматизации сопровождения пакетов уже проработал год. Пришло время подведения итогов.

Основным назначением кластера является сборка новых и обновление старых пакетов. Рассмотрим производительность кластера на примере дистрибутивной платформы ALT Linux Sisyphus. Состоянием на 10 сентября за это время для платформы ALT Linux Sisyphus всего было создано 59904 пакета. Из них 46803 пакета было создано с применением средств автоматизации, силами одного разработчика, а 13101 пакет по традиционной технологии, вручную, силами команды из более чем 100 человек.

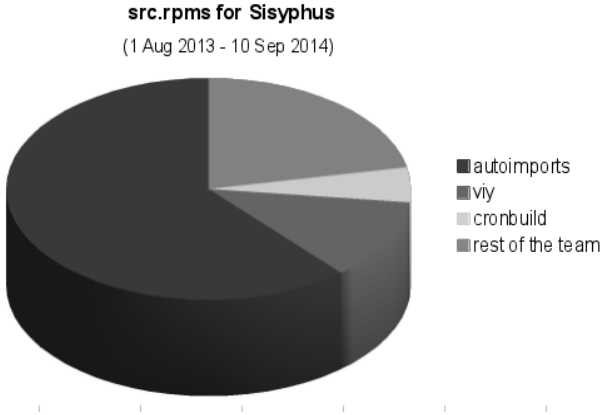

Рис. 1: Вклад кластера автоматизации.

Следующая таблица показывает число сборок пакетов.

autoimports	viy	cronbuild	ALT Linux Team
36803	7017	3034	13101

Из приведенных цифр можно заключить, что использование кластера автоматизации сопровождения пакетов позволило повысить продуктивность работы майнтайнера более чем в 100 раз и внести весомый вклад в развитие дистрибутивной платформы ALT Linux Sisyphus.

Основной проблемой проекта является отсутствие персонала и недостаточная вовлеченность сообщества в его работу. Проект дорос до той стадии, когда усилий одного человекка явно недостаточно.

Облачный кластер автоматизации рассчитан на сопровождение сотен тысяч пакетов. Однако, чтобы раскрыть этот потенциал, необходима команда хотя бы из нескольких человек.

Владимир Коваленко, Дмитрий Костюк
Брест, Брестский государственный технический университет

О применимости мобильных платформ в системах добровольных вычислений

Аннотация

Анализируются современные тенденции в сфере добровольных вычислений (volunteer computing), направленные на включение в GRID мобильных устройств (смартфонов и планшетов). Анализируется влияние типичного сценария использования мобильных устройств и результаты численного сравнения их вычислительных ресурсов с традиционными рабочими станциями, обсуждается круг задач, подходящих для специфики аппаратных ресурсов и режима их использования. Рассмотрены примеры GRID-проектов, ориентированных не на вычислительные задачи. Обсуждаются результаты собственных экспериментов по использованию мобильных устройств для автоматического распределенного тестирования графических приложений.

Термин «добровольные вычисления» (англ. volunteer computing) предполагает использование личных ресурсов пользователей Internet в GRID-сети, ориентированной на решение какой-либо задачи. Появление таких систем относится к 90-м годам (`SETI@home`, проекты `distributed.net` и др.). Сегодня вычислительная мощность пользовательских GRID если не превосходит топовые суперкомпьютеры, то по крайней мере сравнима с ними. Вычисления охватывают задачи математики, криптографии, биологии, медицины, физики и др. Проекты завлекают участников важностью решаемой задачи, статусными

знаками и рейтингами, привлекательными клиентскими приложениями (например, в виде экранной заставки, повышающей социальный статус владельца).

В последнее время клиенты volunteer computing появляются для мобильных платформ, в первую очередь Android — как благодаря основе на Linux, так и в силу особенностей самой платформы, вынуждающих концентрировать в мобильном устройстве вычислительные ресурсы, распараллеленные в виде 2–8 ядер и большую часть времени не задействованные пользователем. Клиенты создаются либо целевым портированием (Android-клиент BOINC), либо пересборкой Linux-версии (одно из ранних упоминаний об экспериментах такого типа, известных авторам — первоапрельская шутка о вычислительном кластере Google на базе Nexus One, приведшая к появлению нескольких реальных аналогов [1]). Однако на включение в GRID мобильных устройств, при всей их многочисленности и многоядерности, действуют ограничения в виде особенностей архитектуры и специфики использования.

Архитектура ARM, возникшая в 80-е годы для рабочих станций под управлением RiscOS, в 90-е годы переориентировалась на наладонные компьютеры благодаря простоте, эффективному использованию блоков процессора и потому низкому энергопотреблению. Дальше оптимизация ARM проходила с упором на этот критерий, и сегодня конечные пользователи вполне осведомлены о том, что «у ARM и x86 разные мегагерцы». Количественно проиллюстрировать разницу можно, сравнив в качестве отправной точки длительность выполнения типичных команд в циклах процессора (рис. 1-а). Показатели семейства ARM на рисунке снизятся еще больше, если учесть меньшие тактовые частоты и разрядность некоторых инструкций; ситуацию не исправляет даже учет большего числа ядер. Если дополнительно пронормировать производительность по рассеиваемой мощности (рис. 1-б), ARM окажется безусловным лидером, но это имело бы значение, будь сегодня востребован низковаттный GRID.

Заметим, что реальная производительность процессоров значительно отличается от такого расчета в пользу суперскалярных x86: использованные технологии оптимизации выполнения потоков команд (спекулятивное выполнение, предсказание переходов, слияние и разделение инструкций и т. д.) делают процессоры более производительными. Реальный разрыв между ARM и x86 заведомо больше, но он плохо поддается расчетам, и его приходится оценивать на конкрет-

	Data transfer	Logical	Integer +/	Integer *	SSE/NEON(VFP)	Special math
ARM Cortex A9	2,66	1	4,16	5,66	10,5	
ARM Cortex A7	3,66	1	5,16	6,66	16,5	
x86 Atom 330	2,66	1	4	19	7	
x86 AMD A10 (Trinity)	2,33	1	3,5	5	5,5	113,33
x86 I7 3xxx	2	1	3,5	3	3	73,33

а)

	Data transfer	Logical	Integer +/	Integer *	SSE/NEON(VFP)	Special math
ARM Cortex A9	98 932	263 158	63 259	46 494	25 063	
ARM Cortex A7	136 612	500 000	96 899	75 075	30 303	
x86 Atom 330	46 992	125 000	31 250	6 579	17 857	
x86 AMD A10 (Trinity)	4 292	10 000	2 857	2 000	1 818	88
x86 I7 3xxx	3 846	7 692	2 198	2 564	2 564	105

б)

Рис. 1: Длительность выполнения инструкций в циклах процессора (а) и результат нормирования по рассеиваемой мощности (б); светлый фон соответствует большей производительности

ных задачах, учитывающих, правда, еще и разницу в быстродействии памяти и дисковой подсистемы, как, например, бенчмарки ресурса `phoronix.com` (рис. 2).

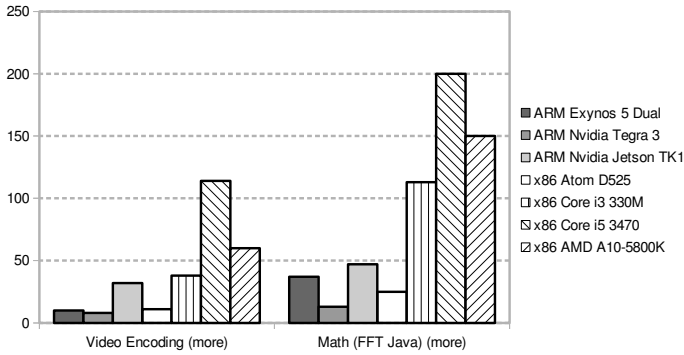

Рис. 2: Сравнение архитектур на реальных задачах; большие значения соответствуют большей производительности

Рабочий процесс мобильного устройства, с другой стороны, предполагает как можно более длительное функционирование от батареи, и как следствие — минимальное использование имеющихся мощностей. Разработчикам приходится это учитывать: так, Android-клиент BOINC работает во время зарядки и передает данные через WiFi. Т. о., помимо меньших вычислительных мощностей, мобильный узел

скорее всего будет оставаться в волонтерской GRID-сети лишь до тех пор, пока сеть практически не нагружает его вычислениями. В результате, применение мобильных устройств в GRID пока носит эпизодический характер.

Итак, задачи, наиболее подходящие для мобильных вычислений, должны либо использовать вычислительную мощность в виде кратковременных пиковых нагрузок (1), либо извлекать пользу не столько из вычислительного ресурса, сколько из территориального разброса (2) и гетерогенности узлов сети (3).

К задачам категории 1 в первую очередь относится самодиагностика с целью измерения достижимых пиковых нагрузок; однако это превращает построение системы скорее в спортивное мероприятие. Задачи типа 2 периодически обсуждаются в прессе (применение смартфонов для создания распределенной сети датчиков). Однако обсуждение остается теоретическим, т. к. массовые устройства не содержат датчиков, полезных в реальных исследованиях или мониторинге. Очевидно, выгодна ориентация на задачи 3-й группы, связанные с гетерогенностью узлов.

Примеры «нематематических» волонтерских вычислений известны. Так, проект Majestic-12 основан на использовании интернет-канала пользователя для построения распределенной системы сетевого поиска, проект Surveill@Home нацелен на оценку производительности и отказоустойчивости веб-сайтов, а один из четырех методов взаимодействия, заложенных в проекте SmartLab, связан с удаленной отладкой приложений на мобильном устройстве [1].

В русле последнего направления нами был проведен эксперимент по оценке возможности построения GRID-сети для тестирования графических интерфейсов мобильных приложений. Фрагментация платформы и многообразие условий, в которых должен работать мобильный продукт, может быть сколько-нибудь ощутимо покрыто тестами только в условиях крупной компании. Сложность для конечного разработчика еще выше с учетом того, что типовые мобильные приложения — графические. Последнее делает заманчивым параллельное тестирование приложений с массовым делегированием их выполнения сторонним устройствам.

Задача анализа результатов работы тестовых приложений решалась нами «в лоб», путем видео-протоколирования и последующей автоматической систематизации скринкастов средствами библиотеки

OpenCV (в отличие от более традиционной связки Android и JUnit, это позволяет отследить ошибки отрисовки элементов [2]).

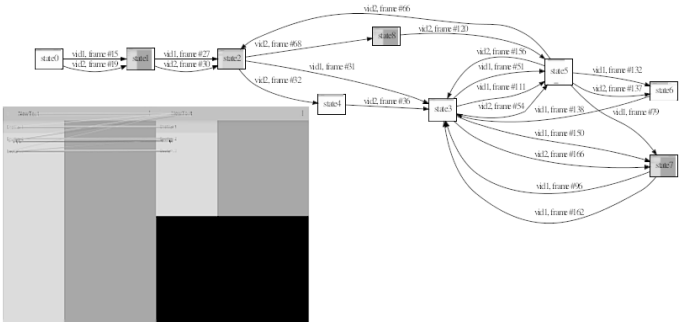

Рис. 3: Выделение контрольных точек в кадре и построение графа переходов

Анализ видео-потоков выполнялся покадровым сравнением. Для нивелирования различий в разрешениях экранов, временной шкале потоков и неравномерности, сравнение производилось по ключевым точкам — моментам, когда на разных машинах отображается определенное состояние интерфейса (использован реализованный в библиотеке OpenCV метод SURF). На рис. 3 можно видеть, помимо переноса контрольных точек, результирующий граф с разветвлениями в точках расхождения скринкастов, нормированных к условной единой временной шкале. С помощью структуры графа и отметок времени на нем отображаются время реакции, ошибки и др. дополнительная информация.

Разумеется, для практического применения на принципах добровольных вычислений требуется решение ряда вопросов построения сети и изоляции личных данных. Однако взамен автоматически решается вопрос привлечения участников, т. к. потенциальные потребители вычислений заинтересованы выступать в роли волонтеров, и сеть может «жить» за счет мобильных устройств самих разработчиков. Более того, хотелось бы отметить идеальное соответствие задачи озвученному паттерну (большая GRID-сеть, которая большую часть времени ничего не делает): ресурсоемкость подхода, использованного нами для тестирования приложений, гарантирует невысокую загрузку мобильных устройств, т. к. при действительно массовом тестирова-

нии для обработки результатов разработчику понадобится отдельный вычислительный кластер.

Литература

[1] *Larkou G. et al.* Managing Smartphone Testbeds with SmartLab // Proceedings of the 27th USENIX Large Installation System Administration Conference (LISA 13), Washington D.C., USA, 2013, pp. 115–132. https://www.usenix.org/conference/lisa13/technical-sessions/presentation/larkou

[2] *Hu C., Neamtiu I.* Automating GUI Testing for Android Applications // 3rd Int. conf. on Advanced Science and Technology, September 27–29, 2011, Seoul, Korea, pp. 77–83. http://www.cs.ucr.edu/~neamtiu/pubs/ast11hu.pdf

Костарев А.Ф.
Пермь, ООО НЕВОД, ГК ИВС

Проект: Репозиторий C2R

Apache Hadoop 2.0 (YARN). Последние тенденции в обработке больших данных (BigData)

Аннотация

Вплоть до 2012 года основным моделью обработки больших объемов данных в *Apache Hadoop* был фреймворк *MapReduce*, обеспечивающий координацию процессов обработки данных (*Map*), распределенных по серверам кластера и процессов формирования итоговых результатов обработки (*Reduce*). Все разрабатываемые в рамках *Apache Hadoop* продукты укладывались в прокрустово ложе этой технологии. В апреле 2012 года разработчики компании *Hortonworks* предложили новую модель координации процессов обработки больших данных — *YARN* (*Yet Another Resource Negotiator*) которая в настоящее находит все большее распространение при создании программного обеспечения в области *BigData*. В докладе рассматриваются основные принципы программной среды *YARN*, средства координации процессов и перспективы использования данной технологии в разрабатываемом в рамках *ГК «ИВС»* репозитория *C2R*.

Модель MapReduce

Модель *MapReduce* распределённых вычислений была предложена компанией *Google* для параллельных вычислений над очень большими (до несколько петабайт) наборами данных в компьютерных кластерах. Суть модели состояла в распределённом хранении и обработке однородных данных. В отличие от модели централизованной обработки данных (передача данных к вычислителю) модель *MapReduce* предоставляла механизм распределенной обработки (передача вычислений к данным).

На каждом сервере, где хранятся обрабатываемые данные, запускается процесс Map, выполняющий обработку части данных, хранящейся на этом сервере. Результатом обработки является список пар ключ→значение которые передаются и процессам *Reducer*, формирующим итоговый результат. Координацию задач (*Task*) в рамках каждой работы (*Job*) производит центральный процесс JobTracker (см. рис.), отвечающий за управление ресурсами кластера, запуском задач *Map*, *Reduce* на узлах кластера и их перезапуском в случае сбоя. Процессы *TaskTracker*, функционирующие на всех обрабатывающих узлах кластера отвечают лишь за запуск/останов задачи по запросу *JobTracker* и предоставления ему информации о статусе выполняемых задач.

Модель MapReduce является удобной средой для выполнения широкого класса задач обработки больших данных (*BigData*). Но существуют алгоритмы в области обработки графов (*Google Pregel*/*Apache Giraph*) и итеративного моделирования (*Message Passing Interface - MPI*) реализация которых в модели MapReduce затруднена.

Более того, так как модель *MapReduce* ориентируется на пакетную обработку данных, то реализация задач, требующих обработки в режиме реального (или близкого к реальному) времени (таких как потоковая обработка) в ее рамках невозможна.

В связи с этим В апреле 2012 года разработчики компании *Hortonworks* предложили новую модель координации процессов обработки больших данных — *YARN* (*Yet Another Resource Neogitator*).

Модель YARN

Фундаментальная идея модели *YARN* лежит в разделении 2-х основных функций *JobTracker'a* управление ресурсами и управления

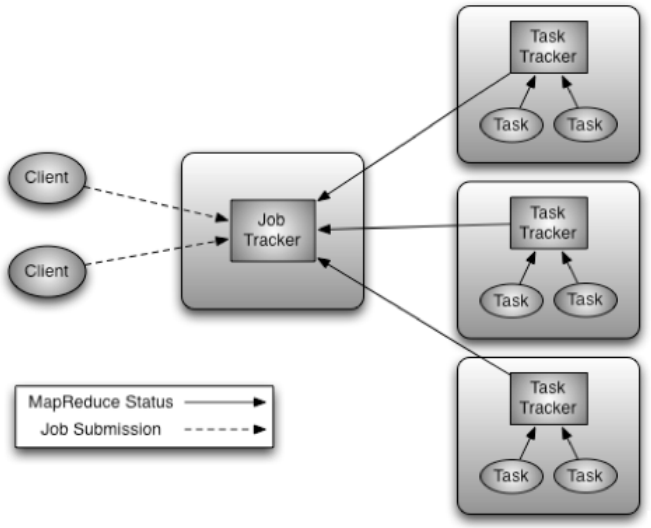

Рис. 1: Координация процессов обработки данных (Map, Combiner, Partitioner, Reduce) в модели MapReduce

задачами. За управление ресурсами отвечает единый на весь кластер ResourceManager (см. рис. 2). Выполнение же каждой запускаемой в *рамках* кластера приложения отвечает отдельный демон — *ApplicationMaster*.

Таким образом каждый *ApplicationMaster* отвечает за запуск, выполнение и завершение своего приложения. В своей работе он:

- запрашивает наличие ресурсов у *ResourceManager*, определяет список серверов на которых необходимо запустить требуемые процессы;
- по полученному списку серверов обращается к *NodeManager'ам* данных серверов для запуска необходимого количества процессов (*Container's*) для обработки данных;
- контролирует выполнение и завершение всех процессов работающих в рамках данного приложения.

Данная модель позволяет запускать и контролировать выполнение широкого класса задач по обработке данных включая задачи по об-

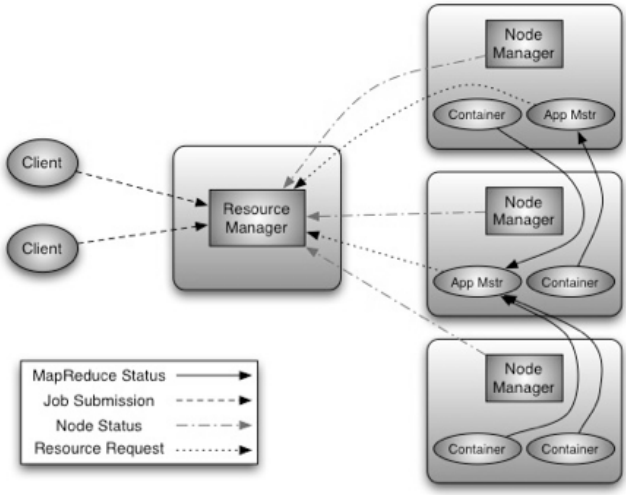

Рис. 2: Координация процессов обработки данных (Map, Combiner, Partitioner, Reduce) в модели YARN

работке графов, постоянно функционирующие задачи по обработке потоковых данных, сервера распределенных баз данных и так далее. Стандартные задачи *MapReduce* в этой модели являются лишь одним из типов поддерживаемых приложений.

Текущее состояние платформы YARN (Hortonworks)

На настоящий момент (сентябрь 2014) наиболее активное участие в развитии модели *YARN* принимает компания *Hortonworks*.

На рисунке 3 отображены основные приложения, поддерживание в рамках платформы *YARN*.

Наиболее существенным нововведением нацеленным на адаптацию существующего программного обеспечения к модели *YARN* является предложенная в апреле 2014 года оболочка *Slider*, позволяющая без изменения кода интегрировать приложения в платформу *YARN*. К этим приложениям, приведенным на рисунке 3 относятся: *NoSQL* база данных HBase, потоковая система обработки данных Storm, поис-

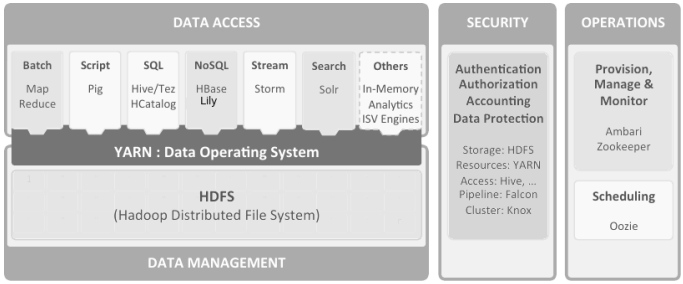

Рис. 3: Текущее состояние платформы YARN

ковая система *Solr* и другие приложения. *Slider* позволяет запускать в рамках одного кластера несколько версий одно приложения, что обеспечивает весь цикл разработки приложений: создание, отладка, эксплуатация.

Разрабатываемый в рамках группы компаний ИВС *облачный контент-репозиторий C2R* использует часть приложений, поддерживаемых моделью *YARN*: *NoSQL* базу данных *HBase*, поисковую система *Solr*, framework *MapReduce*. В ближайшее время планируется перевод репозитория C2R на данную платформу и использование всего комплекса приложений платформы.

Литература

[1] Костарев А.Ф., Полещук А.Н., Контент-репозиорий C2R. Свидетельство о государственной регистрации программы для ЭВМ №2011617248/, 2011

[2] Introducing Apache Hadoop YARN — `http://hortonworks.com/blog/introducing-apache-hadoop-yarn/`

[3] Hortonworks Data Platform — `http://hortonworks.com/hdp/`

Денис Силаков
Москва, ООО «НТЦ ИТ РОСА»

Проект: ROSA, urpmi, ROSA Freeze http://www.rosalab.ru

Средства восстановления системы в ROSA Linux

Аннотация

Дистрибутив Linux является динамично развивающейся средой, постоянно получающей большие пласты обновлений. Обновления призваны привнести в систему улучшения и новый функционал, однако иногда они имеют обратный эффект, ломая что-то работающее и вызывая недовольство пользователей. Кроме того, многие члены сообщества – это энтузиасты, которых постоянно тянет попробовать что-то новое. Например, всевозможные программы из различных источников, которые не имеют прямого отношения к дистрибутиву. Рано или поздно программы могут вступить в конфликт с приложениями и политиками ОС и привести систему (по крайней мере частично) в нерабочее состояние. Как результат, для многих пользователей актуальным вопросом является автоматический возврат поломанной системы в рабочее состояние.

Инструменты уровня пакетов

Поскольку в Росе, как и в большинстве дистрибутивов Linux, установка, удаление и обновление программ сводится к манипуляции с пакетами, то и откат действий по управлению ПО логично осуществлять на уровне пакетов — то есть если с обновленным пакетом что-то не так, то надо откатиться на предыдущую версию. На практике такой подход сталкивается с двумя проблемами — во-первых, пользователь не всегда может однозначно идентифицировать пакет, который привел к сбою, а во-вторых — надо где-то взять старую версию пакета, ведь в официальных источниках (если они использовались для обновления) она уже заменена новой.

Если проблемы вызваны пакетами из сторонних репозиториев, то на помощь может прийти инструмент urpm-reposync (аналог reposync из Fedora), способный привести состояние пакетной базы системы в соответствие с подключенными репозиториями — в частности, удалить пакеты, в этих репозиториях отсутствующие, а для имеющихся

пакетов обновить или откатить их до официальных версий. Urpmreposync обладает некоторой гибкостью (в частности, позволяет не удалять «лишние» пакеты, если их зависимости по-прежнему будут присутствовать в системе после синхронизации с репозиториями), однако в целом является достаточно тяжеловесным инструментом. С его помощью можно провести синхронизацию всех пакетов, установленных в системе, но нельзя откатить конкретные обновления.

Более избирательным является urpmi.recover, работающий на уровне транзакций rpm. Этот инструмент позволяет производить откат пакетной базы по датам или транзакциям — можно откатить заданное число транзакций либо все транзакции до определенной даты. Алгоритм работы инструмента прост — при удалении или обновлении пакетов делается резервная копия удаляемой/обновляемой версии пакета, которая и устанавливается в систему при откате. Отметим, что откатывать транзакции выборочно инструмент не позволяет, однако при сильном желании можно попробовать сделать такой откат вручную с помощью rpm — ведь старые версии пакетов доступны в кэше urpmi.recover.

Полной гарантии успешности отката urpmi.recover дать не может — ведь мэйнтейнеры далеко не всегда задумываются о том, что их пакеты могут не только обновляться, но и откатываться к предыдущим версиям. Соответственно после отката можно ожидать «сюрпризов» в виде неожиданной отработки post-скриптов или некорректного изменения конфигурационных файлов, входящих в пакет. Тем не менее, urpmi.recover снискал популярность у тестировщиков РОСЫ, поскольку позволяет откатывать тестовые версии пакетов проще и быстрее, чем urpm-reposync.

Инструменты уровня файловой системы

Как бы ни привлекательно выглядел откат изменений на уровне пакетов, полностью гарантировать возврат системы в работоспособное состояние он не может. Во-первых, этому может воспрепятствовать специфика пакетов (например, мэйнтейнеры не проработали удаление или откат пакета до предыдущей версии), а во-вторых — ошибки в системе могли появиться и не из-за пакетов, а просто из-за некорректных действий пользователя. Традиционным способом возврата системы в рабочее состояние в таких ситуациях является использование резервных копий.

Сценарии использования систем резервного копирования подразумевают, что «плохие» ситуации в ОС происходят относительно редко, и откат ОС к предыдущему состоянию осуществляется только при выявлении подобных ситуаций. Однако в жизни бывают и другие сценарии, когда с большой вероятностью все действия пользователя в рамках текущей сессии необходимо будет откатить. Такие ситуации нередко встречаются на предприятиях, когда сотрудникам при загрузке машины предоставляется некоторая фиксированная система, которая при выключении машины или перезагрузке возвращается в исходное состояние (возможно, за исключением домашних папок сотрудников и других мест, где они могут сохранять результаты своей работы).

Нередко подобные схемы работы реализуются с помощью виртуальных машин — например, после выключения виртуальной машины можно откатывать ее к заранее сделанному снимку, а домашние директории пользователей монтировать с удаленных ресурсов. Однако Linux предоставляет средства для реализации более легковесных решений, не требующих больших накладных расходов (как в случае виртуализации) и вполне подходящих для домашнего использования.

Подобное решение — ROSA Freeze — мы реализовали в Росе. При использовании ROSA Freeze, ОС может работать в двух режимах — обычном и режиме «заморозки». При использовании последнего поверх всех системных директорий верхнего уровня (/bin, /etc, /usr и так далее; список может быть скорректирован администратором) с помощью aufs монтируются директории-«перехватчики», находящиеся в tmpfs либо на отдельном разделе жесткого диска.

Любые изменения, вносимые в «замороженные» директории, реально попадают в директории-«перехватчики», а содержимое оригинальных директорий остается неизменным. При перезагрузке машины все содержимое этих перехватчиков удаляется и системные директории возвращаются в первоначальное состояние. При этом после перезагрузки система продолжает оставаться в замороженном состоянии — выход из этого режима надо производить явно. Предусмотрена возможность переноса всех изменений из aufs, сделанных в рамках текущей сессии, в оригинальные директории.

Инструмент ROSA Freeze очень молодой, однако уже готов к использованию. В настоящее время инструмент рассчитан на работу в схеме, когда все замораживаемые директории находятся на одном и том же разделе жесткого диска. В будущем мы планируем добавить

поддержку заморозки директорий на разных разделах, а также реализовать в рамках ROSA Freeze механизма точек восстановления — ведь используемые подход с aufs позволяет нам автоматически получать инкрементальные различия между состояниями системы, остается только реализовать средства сохранения таких различий на диск и отката к ним.

Андрей Ненахов
Челябинск, ООО «Редсолюшн»
Проект: Xabber http://xabber.com

Xabber. XMPP-клиент для платформы Android. Проблемы и перспективы протокола XMPP

Аннотация

Экосистема XMPP, текущее состояние, тренды и перспективы. Проект Xabber.

XMPP — основанный на XML открытый протокол для обмена мгновенными сообщениями. Он появился в конце 90-х годов в качестве свободной и открытой альтернативы таким сервисам, как ICQ, AOL Instant Messenger, MSN и другим. Всё шло хорошо где-то до 2008-го года, когда аудитория чувствовала необходимость отказаться от проприетарных сервисов и иметь свободную возможность обмениваться мгновенными сообщениями на базе открытого стандарта. Такие компании, как Google, Yandex, Facebook внедряли свои сервисы обмена мгновенными сообщениями (Gtalk, Я.Онлайн), основанные на протоколе XMPP. Facebook, Вконтакте и Livejournal предоставляли XMPP-сервис (правда, Facebook и Вконтакте делали это без поддержки обмена сообщениями со сторонними серверами). Аудитория ICQ сокращалась. Но потом всё изменилось.

Мобильная угроза

В 2007-м году публике был представлен iPhone, за которым год спустя последовали устройства на платформе Android. Внезапно оказалось, что простота установки и распространения ПО на мобильных платформах позволила стремительно развиться таким сервисам, как

WhatsApp, Tango, Viber и другим. Предоставляя пользователям такие возможности, как обмен файлами, фотографиями, аудио и видеовызовов, сервисы стремительно набрали абонентскую базу. Формально, XMPP протокол также поддерживает все эти возможности, существуют различные XEP на все случаи жизни, но ввиду большого количества различных реализаций клиентского и серверного ПО приходится констатировать факт: пользовательские характеристики XMPP-сервисов сильно отстают от альтернативных предложений. Ввиду этого протокол XMPP так и не смог стать тем, чем стал email: общим стандартом, отказаться от которого не может себе позволить ни одна компания.

Внезапное, но неизбежное предательство

Почувствовав угрозу со стороны альтернативных мобильных сервисов обмена сообщениями, со стороны ведущих технологических компаний призошёл отказ от поддержки федерации. Все захотели построить собственный закрытый сервис. По этому пути пошёл Google, в 2013-м году представивший публике свой новый продукт Hangouts, заменивший Gtalk. Facebook принуждает пользователей использовать только Facebook Messenger их собственной разработки. Отдельно стоит отметить иезуитский подход компании Google: изначально заявив, что новые возможности Hangouts невозможно было реализовать при помощи открытых стандартов (это неправда), и обещав сохранить совместимость с внешними XMPP-сервисами, совместимость отключалась поэтапно.

Проблемы экосистемы XMPP

Было бы глупо объяснять произошедший отход от поддержки открытых стандартов крупными корпорациями одной лишь капиталистической жадностью. Действительно, широкая номенклатура клиентских XMPP-приложений на разных платформах, неполное покрытие ими расширений протокола (т. н. XEP — XMPP Extension Protocol) и, как правило, плохой дизайн отталкивают пользователей от этого решения. Так, автору доклада ни разу не удалось успешно осуществить видео или аудиовызов при помощи протокола XMPP с использованием двух различных клиентских приложений.

Проект Xabber

Несмотря на все свои нынешние недостатки, XMPP имеет свою аудиторию. У протокола значительное количество преимуществ, которых лишены любые закрытые централизованные сервисы. Это децентрализованность, открытость стандарта, безопасность и гибкость. Протокол XMPP подходит для обмена данными с различными устройствами, управления оборудованием и т. д. Наиболее широкое применение XMPP нашёл в корпоративных сетях, где используется для передачи конфиденциальной информации между сотрудниками компании.

В 2010-м году компания «Редсолюшн» выпустила XMPP-клиент Xabber для платформы Android под свободной лицензией GNU GPL v3. Отличием Xabber от прочих мобильных XMPP-клиентов является простой и удобный интерфейс, широкие возможности по настройке отображения списка контактов. В настоящий момент приложение было установлено около 400000 раз. Особой популярностью пользуется функция OTR, позволяющая обмениваться шифрованными сообщениями.

Мы считаем, что главной проблемой XMPP в настоящий момент является комбинация следующих факторов:

- Плохой дизайн приложений;
- Отсутствие мультимедийных возможностей (обмен файлами, изображениями);
- Невозможность удобно и надёжно осуществлять аудио и видеовызовы;
- Отсутствие надёжного контроля доставки сообщений,

Главной причиной такого положения дел является то, что разработкой клиентских и серверных приложений XMPP занимаются разные разработчики, как правило, никак не связанные между собой. Мы считаем, что для разработки высокотехнологичного и удобного в использовании продукта необходимо тесное взаимодействие разработчиков серверного и клиентского программного обеспечения. Команду должно объединять общее видение продукта. Серверная реализация должна поддерживать все требующиеся пользователям функции, реализуя их с помощью XEP, а клиентское ПО с поддержкой этих функций должно быть представлено на всех ведущих платформах, как мобильных (Android, iOS), так и настольных (GNU/Linux, MacOS,

Windows). Приложения должны иметь отличный дизайн, быть просты и удобны в использовании для новичков и иметь возможность гибкой настройки, востребованной опытных пользователей. Мы рассчитываем, что появление на рынке приложений, отвечающих таким стандартам подтолкнет разработчиков прочих XMPP-клиентов также начать поддержку всех необходимых современным пользователям функций.

Для реализации этой задачи мы работаем над новыми версиями Xabber для Android и iOS.

Шигорин Михаил
Москва, Альт Линукс

Проект: ALT Linux Rescue http://altlinux.org/rescue

ALT Linux Rescue

Аннотация

Доклад посвящён развитию и применению образа спасательного диска, созданного на базе технологий и пакетной базы ALT Linux [1].

Введение

Лучше бы никогда не была востребована тематика данного доклада, но в реальном мире люди продолжают ошибаться, «железо» продолжает отказывать и потому вопросы диагностики сбоев, исправления низкоуровневых разрушений, восстановления утерянных либо предположительно уничтоженных данных, оптимизации работы систем остаются всё так же актуальными.

Существует большое количество прикладных пакетов свободного программного обеспечения, предназначенных помочь с решением этих вопросов. Поскольку для их практического использования нередко требуется гарантированно работоспособная или «чистая» среда, создано также и немало спасательных LiveCD («живых дисков», позволяющих загрузить ПК вне зависимости от состояния его собственных носителей). Так, стоит упомянуть и две дискетки tomsrtbt, и специализированные проекты вроде Recovery Is Possible; «обычные» дистрибутивы также склонны включать в состав расширенных вариантов возможность загрузки небольшого «спасателя».

Зачем было делать ещё один? Во-первых, в составе дистрибутивов ALT Linux давно был свой «аварийный кусочек»; во-вторых, оказалось, что наш нестабильный репозиторий прекрасно подходит для автоматического формирования подобных сборок, которые при надобности оказываются доступны вместе со свежими версиями ядра Linux и требуемых программ.

Реализация

В 2007 году был оформлен набросок на базе основной технологии сборки mkimage-profiles-desktop, в 2011 перенесён на разрабатываемые mkimage-profiles; удачность публикуемых в частном порядке «исошек» определила их включение в проект регулярных сборок [2] и еженедельное автоматизированное обновление, причём вскоре «спасатель» оказался наиболее востребованным с широкой географией загрузки. Далее был запущен и проект стартовых наборов [3] с ежеквартальным выпуском расширенного комплекта на базе стабильного репозитория, куда вошёл и rescue.

Образ использует вариант ядра un-def, который сопровождается на основе mainline-ветки kernel.org и обычно соответствует текущей либо предыдущей стабильной версии ядра Linux; как следствие, новое аппаратное обеспечение имеет больше шансов заработать вообще или как положено. Также включены, как правило, недавние версии прикладных пакетов вроде smartmontools или scalpel с аналогичным положительным влиянием на общую функциональность набора.

Контроль качества

Что же делает возможным столь частое обновление сборки на основе нестабильной ветки с сохранением работоспособности, ведь ручная проверка отсеяла считанные единицы результатов?

С одной стороны, это тривиальная выпечка проверочных образов накануне с возможностью внести исправления в пакеты и сборочный профиль до запуска «релизного» процесса; так обычно отлавливались существенно обновлённые ядра без поддержки AUFS, необходимой для всех альтовских «живых» дисков и инсталяторов при нынешнем подходе.

С другой — это следствие фундаментального подхода, выбранного при создании mkimage-profiles, когда «начинка» образов опреде-

ляется в процессе итеративного уточняющего конфигурирования и собирается в основном из «блоков»: при этом оказывается возможно вкладывать усилия в повторно используемые блоки, а проверка качества промежуточных результатов косвенно характеризует и их «потомков».

Так, у всех регулярных сборок есть общая часть – «ствол» дерева конфигурации; найденная при проверке любого образа ошибка после исправления будет ликвидирована во всех производных.

Соответственно и вновь разработанная функциональность может быть применена в различных продуктах путём затребования соответствующего блока (например, обеспечения загрузки в режиме UEFI).

Применение

Нет прока от того, чем не пользуются. Какие-то изменения вносились в rescue по результатам вынужденного задействования на своих системах, другие — по просьбам пользователей или после знакомства с аналогичными изделиями коллег.

Здесь интересно выделить случаи необычного применения, когда люди говорили — «ваша штуковина заработала лучше всего остального». Их пока было три.

Криминалистический режим

Существует узкая специализация подобных продуктов, вызванная необходимостью экспертного анализа остаточных данных на носителях информации; в её рамках важны два свойства применяемого операционного окружения:

- недопущение изменения хранимых, остаточных и системных данных на изучаемых носителях информации;
- защита от исполнения недоверенного кода с этих носителей при загрузке.

Оба вопроса кажутся тривиальными, но благодаря участию Максима Суханова удалось понять нюансы вроде отката журнала файловых систем при монтировании вроде бы readonly и возможность подмены образа второй стадии, а также включить средства противодействия в составе forensic-scripts и проверки контрольной суммы squashfs при загрузке в forensic mode; в результате ALT Linux Rescue весьма неплохо

выглядит даже на фоне разрекламированных «специализированных» продуктов. [4]

Hyper-V gen. 2

Следующая история начиналась так:

<lewellyn> i'm a bit bummed because literally every distro has had a different issue. the rescue image was the only thing which was perfect... and it's not installable. :)

Этот случай к целевому назначению спасательного образа относится весьма косвенно, хотя для вынужденного работать с данным гипервизором в каком-то смысле он оказался спасительным, позволив открыть для себя ALT Linux как таковой и в дальнейшем получить необходимый серверный инсталятор :-)

Кластерный узел

Ещё одному человеку оказалось удобным разобрать наш rescue на запчасти и доработать под себя для организации сетевой загрузки узлов небольшого кластера:

Более подходящего я не нашёл, правда. Много перепробовал, но этот самый маленький, чуть ли не единственный, у которого правильно изначально сделана загрузка с дровами под сетевые и необходимые службы для сетевого монтирования. Он не перегружен сервисами, а так как [...], то библиотек доставлять почти не надо.

Что дальше?

Несмотря на неплохие качества полученного консольного продукта кажется желательной дальнейшая доработка по нескольким направлениям.

Графическая среда

Ряд полезных пакетов применяет графический интерфейс; в первую очередь это инструмент для неразрушающей переразбивки дисков gparted, но во многих аварийных ситуациях могут помочь браузер (особенно с NetworkManager) и файл-менеджер. Существуют и более узкоспециализированные инструменты с GUI, до-

ступные в ALT Linux и находящиеся в русле развития спасательно-восстановительного варианта – например, autopsy.

Собрать LiveCD с X11 несложно; собственно, большинство регулярных сборок являются именно такими. Вопрос в том, как предоставить опытному пользователю наиболее подходящий и удобный инструментарий, при этом не спровоцировав паникующего новичка на необдуманные попадания по кнопкам с необратимыми последствиями. Этот вопрос на данный момент открыт.

Сетевая загрузка

Нет, сам-то Rescue и так пригоден для запуска посредством DHCP+TFTP+NFS – но это требует конфигурирования двух транспортных служб, в то время как известны реализации спасательных комплектов, состоящие из ядра с initrd и потому довольствующиеся лишь DHCP+TFTP (что может быть особенно важно в случае гетерогенных сетей).

Также может оказаться полезной возможность загрузки с флэшки или сидишки на одной из машин в сети и организация на ней сервера сетевой загрузки для остальных нуждающихся в ремонтных работах; здесь наиболее неприятным из возможных побочных эффектов является конфликт с существующим в сети DHCP-сервером с блокированием или существенным затруднением работы клиентов в том же широковещательном сегменте. Надо делать проверку и думать над тем, как запускать такой режим.

Bacula «bare metal» restore

В дистрибутивы Альт Линукс входит мощная система резервного копирования Bacula; инструмент для автоматизации восстановления клиентских узлов «на железе» с новыми дисками выглядит потенциально полезным и стоящим исследования.

...и так далее...

Разумеется, есть и «длинный-предлинный хвост» мелочей, которые в различных ситуациях облегчают произведение и без того требовательных к опыту работ. Из подобных мелочей полезными могут оказаться поддержка последовательной консоли (в т.ч. организованной

средствами IPMI BMC), возможность загрузки во FreeDOS (можно и сейчас добавить, но вопрос в том, как с эмулируемой дискеты штатно добраться до флэшера с прошивкой, которые и потребовали DOS)... С формированием этого списка и в особенности с его реализацией «в буквах» может помочь каждый желающий.

Литература

[1] `http://altlinux.org/rescue`

[2] `http://altlinux.org/regular`

[3] `http://altlinux.org/starterkits`

[4] `http://www.forensicswiki.org/wiki/Forensic_Live_CD_issues`

Илья Щепетков
Москва, Институт Системного Программирования РАН
Проект: Rodin http://www.event-b.org

Rodin — платформа для разработки и верификации моделей на Event-B

Аннотация

При проектировании и анализе сложных систем часто возникает потребность в описании моделей этих систем. Одним из подходов для решения этой задачи является использование формального метода Event-B. В докладе рассказывается о свободной платформе Rodin, которая предоставляет среду для разработки, анализа и верификации моделей на Event-B.

Критичные по безопасности системы, ошибки в которых могут привести к гибели или травмам людей, крупным финансовым потерям и ущербу окружающей среды, имеют повышенные требования к своей корректности. Один из способов повышения уверенности в корректности заключается в моделировании системы и требований к ней и доказательстве их непротиворечия с использованием формальных методов.

Существуют различные формальные методы, многие из которых были использованы в большом количестве проектов индустриального

уровня. Одним из наиболее активно развивающихся и используемых является Event-B. Средства для разработки, анализа и верификации моделей на Event-B предоставляются свободной платформой Rodin — она содержит текстовый редактор моделей, набор систем автоматического доказательства, поддержку интерактивного доказательства. Функционал платформы может быть расширен за счет использования плагинов. Доступны плагины, позволяющие писать собственные расширения формального метода Event-B, дополнительные системы автоматического доказательства, инструменты для анимации и model checking, и т. д.

Наш опыт использования Rodin состоит в формализации и доказательстве корректности модели управления правами доступа и информационным потоками операционной системы специального назначения Astra Linux Special Edition. Модель была полностью формализована и верифицирована, что позволило обнаружить и исправить ряд неточностей в её изначальном текстовом описании.

В результате мы можем сказать, что применение формальных методов даёт следующие преимущества: нахождение ошибок, которые иначе не были бы найдены; повышение уверенности в корректности системы; решение задачи сопровождения системы при последующих правках и расширениях в её описании. Однако, при использовании Rodin мы столкнулись и с трудностями. Следует отметить высокий уровень трудозатрат на проведение формальной верификации; ограниченную поддержку Rodin командной разработки; отсутствие поддержки выделения часто используемых выражений в отдельные сущности с последующим доступом к ним по ссылке.

Платформа Rodin это совместный проект различных команд. Наибольший вклад в разработку вносят Саутгемптонский университет, компания Systerel и Дюссельдорфский университет. В результате мы имеем свободный инструмент, не уступающий коммерческим аналогам, а также успешно используемый для повышения качества критичных по безопасности систем.

Вадим Кузнецов, Николай Крючков
Москва, Калужский филиал МГТУ им. Н.Э. Баумана, Московский институт электроники и математики НИУ «Высшая школа экономики»

Qucs: Использование свободного ПО для моделирования электронных схем в учебном процессе

Аннотация

Описание применения моделировщика электронных схем с открытым исходным кодом Qucs в учебном процессе в МИЭМ НИУ ВШЭ и в Калужском филиале МГТУ им. Н.Э. Баумана. Рассматриваются аспекты использования Qucs, и изменения внесённые в исходный код автором.

Традиционно при подготовке студентов специальностей, связанных с электроникой и радиотехникой используется, используется моделирование электронных схем на ПК. Моделирование позволяет наглядно продемонстрировать работу электронного устройства и провести измерения, недоступные для физического измерительного оборудования.

Как правило, в учебном процессе используется проприетарное ПО для схемотехнического моделирования: MicroCAP, MultiSim, OrCAD и т.п. Всё это сложные программные комплексы стоимостью несколько тысяч долларов. И применение их в учебном процессе для выполнения лабораторных работ, на наш взгляд, не оправдано. В последние годы им появилась альтернатива — программа для моделирования электронных схем Qucs [1, 2]. Название проекта расшифровывается как Quite Universal Circuit Simulatior (почти универсальный симулятор схем). Разработчиками Qucs являются сотрудники Берлинского института высокочастотной техники M. Margraf и S. Jahn. В настоящее время руководителем проекта являются F. Schreuder (Нидерланды) и G. Torri (Италия). Qucs базируется на ядре моделирования специальной разработки. Текущей версией проекта является 0.0.18. В настоящее время ведётся подготовка к релизу версии 0.0.19. Система является кроссплатформенной и работает под управлением ОС Linux, Windows и MacOS. Для достижения наилучших результатов рекомендуется использовать ОС Linux.

Qucs позволяет проводить следующие виды моделирования:

1. Моделирование на постоянном токе (DC analysis)
2. Моделирование в частотной области (AC analysis)
3. Моделирование во временной области (Transient analysis)
4. Параметрический анализ (Parameter sweep)
5. Моделирование S-параметров в частотной области (S-parameter)
6. Синтез пассивных фильтров, согласованных схем, расчёт коаксиальных и микрополосковых линий.

Результаты моделирования можно визуализировать в виде графиков в декартовых (2D и 3D) и полярных координатах, таблиц и диаграмм Смита.

Отличительной особенностью Qucs является возможность анализа комплексных частотных характеристик (КЧХ), построение графиков на комплексной плоскости и диаграмм Смита, анализ комплексных сопротивлений и S-параметров. Эти возможности отсутствуют в системах MicroCAP и MultiSim, и здесь Qucs даже превосходит коммерческое ПО и позволяет получить недостижимые для симуляторов электронных схем, основанных на Spice результаты. Возможности Qucs достаточны для проведения научных исследований [3], связанных с моделированием электронных схем.

Недостатком системы Qucs является малое количество библиотечных компонентов. Но этот недостаток не является препятствием к использованию, так как Qucs совместим с форматом Spice в котором приводятся модели электронных компонентов в даташитах.

В настоящее время система Qucs применяется в Калужском филиале МГТУ им. Н.Э. Баумана на кафедре «Конструирование и производство РЭА» (ЭИУ1-КФ) при проведении лабораторных работ по курсу «Радиотехнические устройства и системы» для знакомства студентов с принципом работы различных радиотехнических схем: усилителей высокой частоты, детекторов, смесителей, фильтров. Также Qucs применяется в Московском институте электроники и математики НИУ «Высшая школа экономики» при курсовом и дипломном проектировании для выполнения компьютерного анализа электронных схем.

Формат схемного файла Qucs основан на XML и к нему поставляется документация. Поэтому схема Qucs может быть легко сгенерирована сторонними программами. Это позволяет создавать ПО для

синтеза схем, которое является расширением Qucs. Коммерческое ПО использует бинарные форматы

Qucs написан на языке C++ и его графический интерфейс реализован с применением библиотеки Qt4, что позволяет легко вносить изменения в исходный код. Проект имеет свой репозиторий на Github `https://guthub.com/Qucs/qucs`, в который каждый может предложить pull request.

Для оптимизации использования Qucs в учебном и исследовательском процессе в исходный код Qucs аспирантом МИЭМ НИУ ВШЭ Кузнецовым В.В. в исходный код были внесены следующие изменения:

1. Реализован список недавних открытых документов в главном меню.
2. Реализован экспорт графиков, схем в растровые и векторные форматы: PNG, JPEG, PDF, EPS, SVG, PDF+LaTeX. Эта функция полезна при подготовке статей и отчётов, содержащих результаты моделирования
3. Возможность открытия документа схемы из будущей версии программы. Функция отсутствует в коммерческом ПО.
4. Исправлены баги, связанные с зависанием моделировщика при определённых условиях.
5. В настоящее время ведётся разработка средства синтеза активных фильтров для Qucs.

Все изменения включены в главную ветку проекта и присутствуют в недавно вышедшем релизе 0.0.18. Средство синтеза активных фильтров `qucs-activefilter` включено в тестовую ветку `https://github.com/Qucs/qucs/tree/ra3xdh-activefilter` и ожидается в следующем релизе 0.0.19. Данный проект имеет свой собственный репозиторий на Github: `https://github.com/ra3xdh/qucsactivefilter`, который использовался до интеграции проекта в Qucs.

Таким образом опыт использования Qucs показывает, что применяя свободное для схемотехнического моделирования можно получать результаты, сравнимые с коммерческим ПО, а по отдельным показателям Qucs даже превосходит коммерческео ПО. Qucs полностью удовлетворяет всем требованим, предъявляемым к моделировщику электронных схем для использования в учебном процессе студентов специальностей, связанных с радиотехникой и электроникой.

Литература

[1] Сайт проекта Qucs. http://qucs.sf.net

[2] Brinson M. E., Jahn S. Qucs: A GPL software package for circuit simulation, compact device modelling and circuit macromodelling from DC to RF and beyond // International Journal of Numerical Modelling (IJNM): Electronic Networks, Devices and Fields. 2008. — September. Vol. 22, no. 4. Pp. 297 – 319. http://www3.interscience.wiley.com/journal/121397825/abstract

[3] Кечиев Л.Н., Кузнецов В. В. Исследование стойкости печатных узлов к воздействию электростатического разряда // Технологии ЭМС. — 2013. — № 1 (44). — С. 29–38.

Андрей Ненахов
Челябинск, ООО «Редсолюшн»
Проект: Электроочередь http://mestovsadik.ru

Электроочередь — свободная система для реализации госуслуги

Аннотация

Опыт внедрения и эксплуатации свободного продукта «Электроочередь», предназначенного для предоставления в электронном виде государственной услуги «Прием заявлений, постановка на учет и зачисление детей в образовательные учреждения, реализующие основную образовательную программу дошкольного образования (детские сады)»

В настоящий момент одной из наиболее актуальных задач, стоящих перед образовательной системой является обеспечение детей местами в детских садах. После разрушения системы дошкольного образования в 1990-е годы возникла хроническая нехватка мест, что привело к длительным очередям на получение мест в детский сад. Традиционно очередь организовывалась на местах силами работников образования, и является значительным источником коррупции.

Автоматическая Информационная Система «Электроочередь» предназначена для предоставления в электронном виде государственной услуги «Прием заявлений, постановка на учет и зачисление де-

тей в образовательные учреждения, реализующие основную образовательную программу дошкольного образования (детские сады)». Система полностью соответствует требованиям Стандарта электронной услуги, разработанного НИУ Высшей школой экономики совместно с Минэкономразвития РФ, Единым функционально-техническим требования к региональному информационному ресурсу, обеспечивающему прием заявлений, учет детей, находящихся в очереди (электронная очередь в ДОО), постановку на учет и зачисление детей в дошкольные образовательные организации в субъектах Российской Федерации».

Реализуя часть программы «Информационное общество», АИС «Электроочередь» позволяет органу местного самоуправления в сфере образования исполнить распоряжение Правительства РФ № 1993-р от 17 декабря 2009 г. и оказывать первоочередные услуги в электронном виде: «Прием заявлений, постановка на учет и зачисление детей в образовательные учреждения, реализующие основную образовательную программу дошкольного образования (детские сады)», а также «Предоставление информации об организации общедоступного и бесплатного дошкольного образования...».

Система направлена на решение социальной задачи обеспечения публичного, гласного и открытого механизма зачисления детей в муниципальные дошкольные образовательные учреждения. В настоящий момент «Электроочередь» является единственным в России программным продуктом, реализующим первоочередную услугу «Прием заявлений...», в котором реализована функция общественного контроля граждан над предоставлением услуги, что уже сейчас соответствует требованиям V этапа реализации распоряжения Правительства.

Основные задачи АИС «Электроочередь»

АИС «Электроочередь» предназначена для решения следующих задач:

- Реализация муниципальной услуги по приему заявлений, постановке на учет и зачисление детей в дошкольные образовательные учреждения (ДОО);
- Автоматизированное формирование отчетности по электронной очереди;

- Обеспечение «прозрачности» процедуры приема детей в ДОО, избежание нарушений прав ребенка при приеме в ДОО;
- Обеспечение единых подходов к учету численности детей, нуждающихся в предоставлении места в ДОО;
- Обеспечение интеграции внешними системами: сервисами СМЭВ, ЕПГУ и РПГУ, порталом Федеральной системы показателей электронной очереди.

Общественный контроль за распределением мест

АИС «Электроочередь» была изначально спроектирована для обеспечения максимально гласного и открытого контроля за распределением мест в детских садах. Прозрачность системы позволяет любому пользователю следить за состоянием очереди. Все данные, необходимые для контроля, соблюдения очерёдности и процесса распределения заявок предоставляются в открытом виде:

- Информация о муниципальных дошкольных образовательных учреждениях;
- Информация о состоянии очереди в текущий момент времени;
- Перечень заявок, распределённых в каждый детский сад;
- Данные об изменениях в заявках, влияющие на их положение в электронной очереди;
- Актуальная статистика, показывающая достоверную информацию о состоянии очереди;
- Для просмотра доступна публичная история изменений каждой заявки.

В комплексе всё это предоставляет каждому гражданину возможность самостоятельно контролировать процесс распределения мест в детских садиках, и, в случае выявления нарушений, апеллировать к достоверным данным, содержащимся в АИС «Электроочередь».

Система, к сожалению, не может решить задачу по обеспечению местами в детских садах всех желающих, но показала свою высокую эффективность в вопросе организации открытого и гласного процесса справедливого распределения мест в детских садах.

История создания

Работы по созданию АИС «Электроочередь» были начаты в 2010м году по запросу Управления по делам образования г. Челябинска. Ознакомившись с опытом эксплуатации подобной системы в г. Екатеринбурге, где система записи в очередь через Интернет позволяла операторам бесконтрольно перемещать заявки в очереди, а гражданам был доступен только их порядковый номер в очереди. Номер, при этом, мог без объяснения причин как увеличиваться, так и уменьшаться. Мы решили, что это неприемлемо, и положили в основу системы максимально допустимую открытость всех данных об состоянии очереди, распределении мест, действиях операторов. Все действия журналируются и,в деперсонализированном виде, доступны всем желающим.

В 2011-м году первая версия системы была внедрена в г. Челябинске. По опыту эксплуатации было принято решение о доработке системы. Новая версия получила возможность интеграции с Единым Порталом Государственных Услуг, органами ЗАГС, ФМС, и была успешно внедрена в 30 муниципалитетах Челябинской области. К сожалению, в дальнейшем продвижении продукта в регионы мы столкнулись с проблемой того, что, несмотря на ряд функциональных преимуществ перед любыми аналогичными системами и востребованность системы гражданами РФ, открытость в работе системы максимально невостребована органами управления образованием.

В связи с этим, а также с учётом философии нашей организации, было принято решение выпустить АИС «Электроочередь» под открытой лицензией GNU AGPL v.3. Исходные коды системы доступны на Github: `https://github.com/redsolution/electroochered`

Любой муниципалитет может прибегнуть к нашим услугам по внедрению продукта, либо самостоятельно внедрить его.