

УДК 004.45

МОДУЛЬНАЯ РЕАЛИЗАЦИЯ ВЕБ-РЕСУРСОВ ФАКУЛЬТЕТА**Власенко А.П., Самощенко А.В.**

Донецкий национальный технический университет

кафедра компьютерной инженерии

E-mail: antares.spica@gmail.com

Аннотация

Власенко А.П., Самощенко А.В. Модульная реализация веб-ресурсов факультета. Рассмотрены варианты реализации веб-ресурсов факультета согласно поставленной задаче. Описана реализация модульного виртуализированного метода решения. Определены проблемы в существующей реализации и найдены пути их устранения.

Общая постановка проблемы

На практике в ДонНТУ появилась следующая задача: предоставить современный и удобный хостинг сайтов для всех кафедр. Фактически, это задание входит в общее описание модульной системы — каждой кафедре необходим похожий ресурс, который может быть модулем общей системы. Задача имела особые требования к надежности и эффективности.

Исследования

Стандартный метод реализации подразумевает выделение физической машины каждой кафедре. Это самый распространенный и тривиальный вариант решения задачи, его недостатки можно оценить — таким образом реализовывались все проекты до этого. Кроме недостатков, которые были выявлены в обычных системах на этапе начала построения концепции модульной системы, выявились и практические недостатки — системы слишком отличались друг от друга, чтобы применять комплексный подход к их поддержке, это же создавало существенные проблемы при автоматизации поддержки. Каждая система имеет индивидуальные трудности и реализована индивидуально, реалии учебных заведений таковы, что самым оптимальным решением будет система, которая совершенно не потребует поддержки, даже в самых мелких масштабах. Модульность, использующая машины-клоны, является важным параметром, упрощающим поддержку системы в будущем. Один патч применяется сразу ко всем машинам. По этим же соображениям в пределах одной компании, по возможности, принято использовать один дистрибутив или одну версию операционной системы — так значительно проще содержать всю систему.

Основные методы реализации

Рассмотрим основные возможные методы решения задачи и отметим преимущества и недостатки данных методов.

При реализации хостинга нескольких ресурсов существуют различные подходы:

- Выделение каждому ресурсу по физической машине.
- Выделение каждому ресурсу по виртуальному хосту одного веб-сервера.
- Смешанный подход.
- Выделение каждому хосту по собственной OpenVZ машине.
- Выделение каждому хосту полноценной виртуальной машины.

Методы под номер 1, 4 и 5, фактически, предоставляют услугу «выделенный сервер», то есть при необходимости пользователям может быть предоставлен полный или частичный контроль над машиной.

Рассмотрение существующих подходов.

Выделение физической машины достаточно простой, относительно минималистичный в плане реализации метод, однако, имеющий существенные трудности. Во-первых, необходимо приобрести соответствующее число физических машин — часто задача может быть экономически невозможной. Во-вторых, эти машины должны где-то храниться, они будут занимать достаточно много места, и для машин всех кафедр факультета КНТ ДонНТУ потребуется небольшой шкаф и комната, где его хранить (помещение, где сейчас хранится сервер, не подойдет). В-третьих эти машины будут пропорционально их количеству потреблять энергию. Каждая машина потребует пропорционального и одинакового контроля со стороны администрации, скорее всего, потребуется несколько человек на каждую машину.

Второй подход имеет такие же достоинства, что и четвертый и пятый, но абсолютно лишает возможности пользователям использовать машину для каких-либо других нужд (установка модулей для Python потребует вмешательства администрации). Практически, можно выполнять требования всех желающих и предоставлять нужное программное обеспечение и порты, однако, каждое новое ПО компрометирует безопасность всей системы, всех сайтов, всех ресурсов, и любая уязвимость сайта может быть использована для манипуляции над всей системой[1].

Третий подход имеет преимущества и недостатки первого и второго метода.

Теперь рассмотрим более современные подходы. Первый из них — OpenVZ. OpenVZ это исключительно Linux-система виртуализации, которая реализует все машины при помощи одного ядра. «К преимуществам OpenVZ по сравнению с более универсальными инструментами виртуализации, такими как Xen и KVM, является прозрачный доступ из внешней системы к процессам, файлам и прочим ресурсам в гостевых.» [2]

Эта система производительна и универсальна, и она рассматривалась как метод реализации, когда начинались работы над новой инфраструктурой факультета в 2010 году. Но тогда же была найдена новая свободная система — Xen, которая постепенно вытесняет OpenVZ с рынка VS-услуг[3]. Xen предоставляет полное разделение виртуальных систем друг от друга, каждая система может использовать собственное ядро (в том числе OS Windows, при крайней необходимости), модули — все как на физической машине, но без лишнего энергопотребления и затрат [3]. Наоборот, применение нескольких машин на одной физической позволит рациональнее использовать серверные ресурсы, т. к. большинство сайтов кафедр не имеют достаточных нагрузок, чтобы загрузить даже устаревшую аппаратную систему более чем на 10%.

Состояние ресурсов на 2010 год

На 2010 год состояние ресурсов на факультете КНТ было следующим: ресурсы деканата обслуживались на устаревшей версии Windows, серверные ресурсы деканата разрабатывались собственным штатом разработчиков. Интернет-ресурсы факультета представлялись довольно разрозненно, собственным набором серверов, при том у каждой кафедры индивидуально — кафедры работали над индивидуальными веб-проектами и движками, которые не использовались на других кафедрах. Человек, обслуживающий веб-сервер факультета, больше не мог продолжать работу над ним. Хотя и сервер был передан в обслуживание другим людям, его развитие остановилось.

Создание системы виртуальных машин начало реализацию модульной виртуализированной системы. Начало обустройства пришлось на наиболее легкой в интеграции области — хостинге веб-ресурсов факультета.

Конфигурация сервера

Аппаратная платформа была предоставлена факультетом — серверная платформа HP Proliant с комплексной системой вентиляции и поддержкой до двух процессоров на плате.

Сервер представляет из себя обычный tower системный блок. Тем и примечательно, что этот системный блок уже сейчас заменяет целую комнату серверных компьютеров.

Первым этапом настройки сервера является его сборка и установка. Данный сервер был собран заранее обслуживающей компанией, поэтому единственной задачей была установка. Но даже этот этап в будущем может отсутствовать, если сборка будет осуществляться удаленным дата-центром, или же, если же виртуальные машины будут выделяться как «облачные» ресурсы.

Установка базовой системы

Для базовой системы была выбрана ОС Debian, которая известна своей стабильностью и популярностью в различных вузах и научных проектах [4]. Данная статья не описывает особенности развертывания операционной системы Linux, а сосредотачивается в первую очередь на превращении базовой системы в виртуализированный сервер для различных проектов.

Установка Xen

Как и в предыдущем разделе, здесь не будут описываться особенности и сложности установки Xen как системы. Также не будут рассматриваться вопросы, почему именно Xen используется в качестве базовой системы для виртуальных машин, так как эти вопросы уже рассматривались в рамках исследования. За последние 6 месяцев ситуация на рынке виртуализации кардинально не изменилась. Xen устанавливается на базовую систему из стандартного репозитория и оттуда же получает регулярные обновления безопасности до тех пор, пока дистрибутив поддерживается (в случае с текущей версией Debian Stable — как минимум до 2014 года). После система переводится в новую стабильную ветку.

Создание виртуальных машин

Для удобного и быстрого создания виртуальных машин в Debian и Xen есть соответствующие утилиты, которые в традиционном UNIX стиле кооперируют друг с другом. В первую очередь стоит отметить утилиту Debootstrap.

В Debian ее установка осуществляется через репозиторий:

```
aptitude install debootstrap
```

Эта утилита позволяет создавать образы операционных систем Debian и Ubuntu, скачивая весь необходимый набор данных с серверов обновлений и извлекая их в целевую папку.

Для этого необходимо предоставить приложению готовую файловую систему. Linux позволяет создавать файловые системы внутри файловых систем, то есть любой файл может служить как настоящий раздел жесткого диска. Поэтому при конфигурировании базовой машины на RAID-массиве был выделен раздел для хранения виртуальных машин, состоящий из папок и файлов для виртуальных машин. Каждая машина имеет свой конфигурационный файл, файл файловой системы и файл подкачки.

Ниже представлен набор команд, по которому можно проанализировать этапы работы. Это стандартные Linux-утилиты, подробная информация о параметрах которых доступна в системном справочнике man.

Создание образов:

```
dd if=/dev/zero of=/xen/domains/name/disk.img bs=1k seek=30G count=1
```

Создание файла подкачки:

```
dd if=/dev/zero of=/xen/domains/name/swap.img bs=1k seek=1G count=1
```

Разметка файлов, создание файловой системы:

```
mkswap /xen/domains/name/swap.img  
mkfs.ext4 /xen/domains/name/disk.img
```

Образы подготовлены, можно загрузить в них виртуальную машину.

Создается директория для монтирования:

```
mkdir /xen/domains/name/mount
```

Монтирование образов:

```
mount -o loop /xen/domains/name/disk.img /xen/domains/name/mount
```

Скачивание и установка дистрибутива в директорию.

```
debootstrap --arch amd64 {distro-name} /xen/domains/name/mount {repository-name}
```

Таким образом, путем базовых манипуляций была получена полноценная операционная система внутри файла. И она обладает всеми теми качествами, которые свойственны файлам — ее можно копировать и переносить куда угодно.

Для распознавания файла системой Xen необходим конфигурационный файл, для примера ниже представлен конфигурационный файл, сгенерированный для машины кафедры КИ:

```
# Configuration file for the Xen instance ki, created  
# by xen-tools 4.2 on Wed Feb 29 12:18:01 2012.  
# Kernel + memory size  
bootloader = '/usr/lib/xen-default/bin/pygrub'  
vcpus      = '1'  
memory     = '512'  
# Disk device(s).  
root       = '/dev/xvda2 ro'  
disk       = [  
              'file:/xen/domains/ki/disk.img,xvda2,w',  
              'file:/xen/domains/ki/swap.img,xvda1,w',  
            ]  
# Hostname  
name       = 'ki'  
# Networking  
vif        = [ 'ip=10.0.0.16,mac=XX:XX:XX:XX:XX:XX' ]  
# Behaviour  
on_poweroff = 'destroy'  
on_reboot   = 'restart'  
on_crash    = 'restart'
```

Система готова к запуску и наполнению — пользователь может выбрать, представить систему для веб-ресурсов или для каких-либо иных задач. Дальнейшие действия выполняются всего лишь один раз — базовую систему можно и нужно клонировать, после того, как ей будет предоставлен весь функционал. Таким образом, была получена легко масштабируемая модульная система, использующая последние наработки в области виртуализации, и предоставляющая практически 100% отдачу по производительности.

Для общей статистики: виртуализированная среда факультета КНТ представляет собой набор из виртуальных машин для кафедр, конференции, одна машина является обратным прокси-сервером, который распределяет запросы и кеширует ответы. Итого — больше 10 виртуальных машин, представленных в одной физической.

Сложности при реализации

В процессе реализации системы возникли сложности. Самой большой сложностью на данный момент является отсутствие специалистов, способных управлять данной системой. Это не является проблемой для глобальной реализации системы, так как обучение системе Xen является не сложным для администраторов Linux, однако на данном этапе это вызывает сложности. К другим сложностям относятся вопросы оптимальности сетевой архитектуры, которые будут обсуждаться с квалифицированными специалистами из дата-центра университета.

Поиск решений

Именно сложности с отсутствием специалистов вынуждают изобрести инструмент для облегченного управления всей системой, чтобы даже неопытные пользователи могли создавать виртуальные машины и регулировать правила. Также практическая реализация уже сейчас требует создания особой документации к системе. В итоге получается комплексное решение — исчерпывающая документация для администраторов системы и утилита, которая будет помогать и администраторам, и обычным пользователям системы. Написанию документации и разработке этой утилиты будет посвящена дальнейшая работа.

Список литературы

1. Ищем и лечим web shell на взломанном сайте / Интернет-ресурс. - Режим доступа : [www/ URL: \[http://wiki.linux.ru/wiki/index.php/ищем и лечим web shell на взломанном сайте\]\(http://wiki.linux.ru/wiki/index.php/ищем_и_лечим_web_shell_на_взломанном_сайте\)](http://wiki.linux.ru/wiki/index.php/ищем_и_лечим_web_shell_на_взломанном_сайте)
2. Евсеев И. Система виртуализации OpenVZ [IBM developerWorks Россия] / Интернет-ресурс. - Режим доступа : [www/ URL: \[http://www.ibm.com/developerworks/ru/library/l-openvz_1/index.html\]\(http://www.ibm.com/developerworks/ru/library/l-openvz_1/index.html\)](http://www.ibm.com/developerworks/ru/library/l-openvz_1/index.html)
3. OpenHosting - XEN vps против OpenVZ / Интернет-ресурс. - Режим доступа : [www/ URL: <http://openhosting.ru/vps/xen-vs-openvz.jsp>](http://openhosting.ru/vps/xen-vs-openvz.jsp)
4. Debian Project - Who's using Debian? / Интернет-ресурс. - Режим доступа : [www/ URL: <http://www.debian.org/users/>](http://www.debian.org/users/)