

## ***Сравнительный анализ инфраструктур для облачных вычислений***

### ***Введение***

Благодаря парадигме IaaS становится возможным виртуализировать практически любую аппаратную составляющую, что является безусловно полезным при желании предоставить пользователю значительные вычислительные ресурсы удаленно. Такие задачи возникают и в процессе обучения информационным технологиям: удобно становится один раз установить и настроить, скажем, образ операционной системы с нужными учебными примерами, а затем благодаря интерфейсу, предоставляемому облаком, запускать этот образ для каждого студента. Подобное «учебное» облако становится решением проблем, связанных с бесконечной настройкой и переносимостью учебных сред. Для того, чтобы внедрить подобное облако, необходимо прежде всего ознакомиться с существующими облачными решениями.

В классическом варианте облачной вычислительной системы можно выделить шесть основных компонент.

Во-первых, имеются физические машины и их операционные системы. Мы выделяем их, потому что: а) если процессоры, используемые в машинах, не поддерживают технологию аппаратной виртуализации, то мы можем говорить лишь о паравиртуализации, и это существенно снижает как вычислительные способности системы, так и выбор программного обеспечения в дальнейшем; б) системы с открытым кодом в большей степени, чем другие, должны быть достаточно гибки для работы со многими гостевыми ОС (а коммерческие могут быть узко специализированы на работу с определенным набором программных компонент).

Во-вторых, сетевая компонента. Использование сети влечет за собой DNS, DHCP и организацию подсетей физических машин. Это также включает виртуализацию внутри сети для того, чтобы дать каждой виртуальной машине уникальный MAC адрес. Все компоненты должны быть соответственно сконфигурированы для работы с виртуальными узлами в той же степени, что и с физическими.

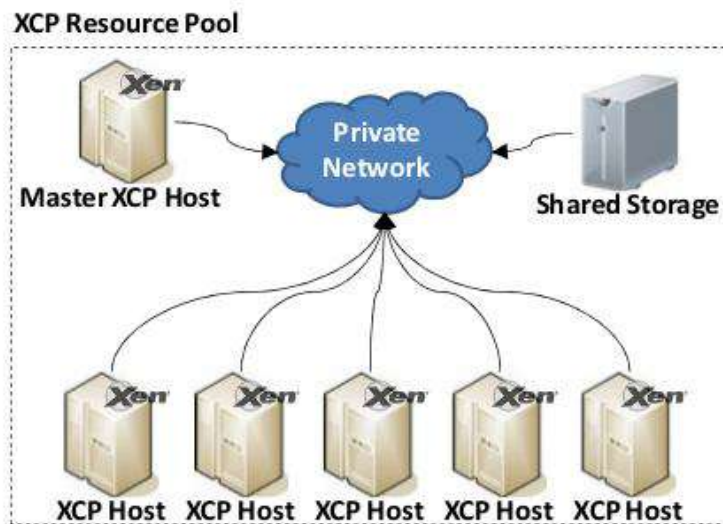
В-третьих, использование гипервизора. Гипервизор позволяет работать виртуальным машинам. В добавок к мониторингу виртуальных машин, гипервизоры используют средство libvirt для того, чтобы контролировать запуск и останов машин. Кроме того, каждый гипервизор должен быть запущен и настроен индивидуально под каждую машину.

Четвертая компонента облачной инфраструктуры – образы виртуальных машин, хранящиеся на виртуальном «жестком диске» – репозитории, охватывающем все образы всех машин, запущенных когда-либо в облаке. Естественно, необходимо предусмотреть и архивацию, и надлежащее обслуживание репозитория.

В-пятых, необходимый интерфейс для пользователей.

И, наконец, облачная среда как таковая – система, связывающая воедино все шесть компонент, распределяющая ресурсы пользователям, обслуживающая сеть и так далее.

### **Сравнение облачных инфраструктур**



*Рисунок 1. Архитектура Xen Cloud Platform*

### **Xen Cloud Platform**

Гипервизор Xen – готовое решение для виртуализации оборудования, который предоставляет промежуточный слой между аппаратным обеспечением сервера и его операционной системой. Гипервизор Xen поддерживает несколько «виртуальных серверов», работающих под разными операционными системами на единственном физическом сервере. Путь Xen проделали многие современные облачные сервисы, такие как Amazon EC2, Nimbus и Eucalyptus.

В настоящее время Xen представил Xen Cloud Platform (XCP) как способ облачной виртуализации инфраструктуры. Однако, в отличие от других облачных проектов с открытым кодом, XCP не предоставляет полной архитектуры для облачных вычислений. Их цель – предоставить средство с автоматической конфигурацией для поддержки облачных платформ.

Архитектура XCP основана на XCP хостах, которые ответственны за обслуживание гостевых операционных систем. В соответствии с Xen.org, эти хосты объединены в общее хранилище ресурсов, и, используя это хранилище, виртуальные машины могут быть запущены или перезапущены на любом хосте. Мастер-хост XCP обеспечивает интерфейс

администратора и пересылает управляющие сообщения другим хостам.

К несомненным достоинствам Xen можно отнести гибкость настройки, надежный гипервизор с огромными возможностями, такими как динамическое изменение ресурсов для каждой виртуальной машины; поддержку миграции образов в процессе работы (live migration). XCP может быть комбинирован с другими инфраструктурами, прежде всего с OpenStack, который использует XenServer в качестве координирующего узла.

К недостаткам XCP относится, прежде всего, сложность внедрения, необходимость ручной настройки практически всех компонентов. С другой стороны, буквально за два года развития XCP документация стала существенно подробнее, многие дистрибутивы Linux стали поддерживать Xen и XCP.

### **OpenNebula**

Идея проекта OpenNebula с открытым кодом – частное или корпоративное облако с большими вычислительными возможностями. Позиционируется как замена стандартным дата-центрам для компаний.

Интерфейс, предоставляемый OpenNebula – обертка вокруг XML-RPC, который тоже может быть использован. Настраиваемость OpenNebula касается как администраторов, так и пользователей облака. С точки зрения администратора, богатые возможности для переноса виртуальных машин имеются у OpenNebula благодаря файловой системе NFS, хранящей все образы дисков облака. Централизация, полученная в OpenNebula, хороша для ее администрирования.

Пользователь в OpenNebula имеет возможность задавать конфигурационный файл при порождении виртуальной машины, за счет чего можно получить, вообще говоря, любые ресурсы, которые может предоставить облако. Обратной стороной является высокая вероятность ошибочной конфигурации виртуальной машины.

Благодаря NFS вычислительные узлы могут не иметь много места на их хранилищах. Однако, сама NFS может стать узким горлышком в системе, требуя много ресурсов – стандартная установка на 64 узла предполагает использование одного терабайта памяти.

К тому же, NFS не предполагает шифрования трафика внутри файловой системы, что может сильно сказаться на безопасности.

### **Ubuntu Enterprise Cloud**

Ubuntu Enterprise Cloud (UEC) – программная среда для создания облачной инфраструктуры, поддерживаемая компанией Canonical. Главная цель, которую ставили перед собой разработчики UEC – простота развертывания облачной инфраструктуры.

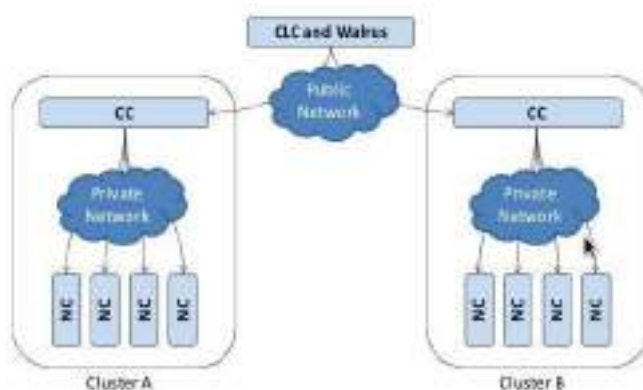
UEC определяет три уровня безопасности: идентификация и аутентификация, изоляция сетей и изоляция вычислительной машины как таковой. Уровень идентификации и аутентификации обеспечиваются локально генерирующимися сертификатами X.509, изоляция сети важна

для предотвращения резких скачков трафика, и, наконец, уровень изоляции вычислительной машины состоит из сетевой изоляции, изоляции операционной системы машины и изоляции машины-гипервизора.

Инфраструктура UEC была основана на архитектуре Eucalyptus вплоть до версии Ubuntu 11.10 Server, сейчас используется архитектура OpenStack.

### **Eucalyptus**

Eucalyptus был разработан как open-source ответ на коммерческое облако EC2. Eucalyptus использует в качестве интерфейса для пользователя программу, которая называется euca2ools, которая очень похожа на программу из облака Amazon, использующуюся в качестве front-end для пользователей.



**Рисунок 2:** Компоненты системы Eucalyptus

Полная архитектура и спецификация Eucalyptus была опубликована, и много документации по Eucalyptus в настоящее время доступно в сети Интернет. Основные компоненты среды Eucalyptus:

1. Node Controller (NC). Эта компонента запускается на всех машинах, которые будут в дальнейшем использоваться для запуска гостевых ОС. NC отвечает за очередность и контроль системного программного обеспечения (собственно, ОС и гипервизора) и за выполнение запросов «отвечающего за него» Cluster Controller (CC). Роль NC во всем этом – сбор системной информации, такой как физические ресурсы узла и состояния виртуальных машин на узле. Всю эту информацию он отправляет на CC. Также NC помогает CC в авторизации, исполнении запросов гипервизора.
2. Cluster Controller (CC). Эта компонента обычно выполняется на машине, являющейся кластером, или на любой машине, имеющей две связи – со множеством NC и Cloud Controller (CLC). CC отвечает за сбор и отправку информации о виртуальных машинах, запущенных на различных узлах и составляет для них расписание.
3. Storage Controller (Walrus). Эта компонента представляет собой сервис хранилища данных, который предоставляет механизм для доступа и содержания образов виртуальных машин и пользовательской информации. Walrus основан на технологиях web-сервисов и

полностью совместим с интерфейсом Amazon's Simple Storage Service.

4. Cloud Controller (CLC). Эта компонента является отправной точкой для «входа» в облако для пользователей. Ее основная цель – предоставление и управление виртуализированными ресурсами. CLC ответственен за предоставление менеджерам ресурсов узлов, за составление расписания, и внедрение его в СС.

Вся инфраструктура Eucalyptus подчеркивает его коммерческого вдохновителя. Во-первых, сильное разграничение пользовательского пространства и пространства администратора. Доступ как root позволяет администратору системы делать на узле все, что возможно вплоть до физического уровня. Пользователи системы могут пользоваться только теми инструментами, которые предоставляет web-интерфейс. За некоторым исключением, Eucalyptus предотвращает практически все попытки пользователей как-то вмешаться в работу вычислительного узла.

Администратор Eucalyptus определяет пять доступных конфигураций ресурсов, которые пользователь вправе назначить для своей виртуальной машины.

Конфигурация программного обеспечения предполагает использование децентрализованного подхода, и как кластеры могут быть представлены в такой системе в качестве одного узла, так и отдельные узлы могут быть объединены в кластер.

Eucalyptus поддерживает распределенное хранилище данных, Walrus, которое на самом деле повторяет аналогичное хранилище в технологии от Amazon. Пользователям выделена некоторая часть общего хранилища, которое они могут использовать. Хранилище разделено на структуры, которые «раскиданы» по облаку. Однако, образ виртуальной машины хранится на вычислительном узле. Кроме того, виртуальная машина может быть поднята и на узле хранилища, однако она может работать только обособленно от других машин.

Децентрализованный подход Eucalyptus со множеством кластеров, распределенным хранилищем и локально хранимыми образами дисков способствует включению в облако множества машин.

Основными достоинствами Eucalyptus являются простота внедрения, совместимость с коммерческим облаком EC2, обширная документация и большое сообщество пользователей. К недостаткам можно отнести невысокую гибкость и настраиваемость.

### **OpenStack**

Разработчики OpenStack заявляют те же цели, которые заявляли разработчики OpenNebula – готовые решения для частных и корпоративных облаков, с хорошей масштабируемостью и быстрым доступом к ресурсам. Рассмотрим подробнее компоненты этой архитектуры.

1. Хранилище файлов «Swift» позволяет обмениваться файлами. Здесь лежат рабочие файлы пользователей, хранилище реализует обмен

- ими в ходе работы.
2. Репозиторий образов виртуальных машин «Glance». По сути, является интерфейсом для «Swift». Пользователь оперирует образами репозитория, но информация на самом деле хранится в «Swift».
  3. Вычислительная компонента «Nova». Дает доступ непосредственно к ресурсам системы по запросу.
  4. Пользовательский интерфейс предоставляет компонента «Horizon». Это веб-интерфейс, позволяющий выполнять все функции по администрированию образов виртуальных машин, в том числе и контролю доступа.
  5. Модуль авторизации и аутентификации «Keystone» решает эти проблемы для всех компонент OpenStack, а также предоставляет каталог всех сервисов конкретного облака.
  6. Сетевая компонента «Quantum» позволяет пользователям создавать свои сети и затем подключать к ним различные компоненты.

OpenStack в современном виде поддерживается компанией Canonical, на базе этой технологии работает Ubuntu Enterprise Cloud, который входит во все серверные версии Ubuntu Server, начиная с 11.10. Стоит заметить, что минимально рекомендуемое число машин – 6, в то время как Eucalyptus можно было установить даже на одну машину.

К преимуществам OpenStack можно отнести современную поддержку крупнейших компаний, легкое внедрение и дальнейшую эксплуатацию, а также хорошую возможность масштабирования.

К недостаткам этой технологии относятся требовательность к ресурсам и сложная архитектура, сочетающая в себе большое число компонент.

### ***Заключение***

Рассмотренные выше технологии позволяют реализовывать как большие, сложные системы с привлечением больших вычислительных мощностей, востребованные крупными компаниями, так и небольшие частные облака. Большие возможности настройки позволяют эффективно администрировать имеющиеся ресурсы. Хорошая поддержка позволяет реализовывать практически любые задачи, в том числе и образовательные.

### **Литература**

1. Peter Sempolinski, Douglas Thain. A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus. Париж, 2009.
2. Damien Cerbelaud, Shishir Garg, Jeremy Huylebroeck. Opening The Clouds: Qualitative Overview of the State-of-the-art Open Source VM-based Cloud Management Platforms. Сан-Франциско, 2010.
3. Patrícia Takako Endo, Glauco Estácio Gonçalves, Judith Kelner, Djamel Sadok. A Survey on Open-source Cloud Computing Solutions. Universidade Federal de Pernambuco, 2010.