

Introduction to cloud computing architecture

Виртуальные машины как стандарт развертывания объектов

В течение последних нескольких лет виртуальные машины стали стандартом развертывания объектов. Виртуализация расширяет гибкость, так как на одном и том же аппаратном обеспечении в одном месте могут разворачиваться и сворачиваться приложения, не привязываясь к конкретному серверу. Виртуальные машины становятся распространенной абстракцией-единицей развертывания, потому что они приводят к общему знаменателю интерфейсы между поставщиками услуг и разработчиками. Использование виртуальных машин для развертывания объектов достаточно в 80 процентах случаев использования и помогает удовлетворять потребностям быстрого развертывания и масштабирования приложений. Виртуальные приборы-это виртуальные машины, которые включают программное обеспечение, которое частично или полностью настроено для запуска специфических программ, таких как Web серверы или серверы баз данных и далее увеличивается возможность по созданию и быстрому развертыванию приложений. Комбинации виртуальных машин и виртуальных приборов, как стандарта развертывания объектов -ключевое свойство Cloud computing. Облачные вычисления обычно дополняются облачными хранилищами, которые предоставляются виртуальными хранилищами через API, которые облегчают хранение образа виртуальных машин, исходных файлов для компонентов таких как веб-серверы, информация о состоянии приложений и важная бизнес информация.

The on-demand, self-service, pay-by-use model

Все эти три модели, являющиеся первоначальными для СС так же расширяются в условиях этих тенденций. Модель on-demand помогает поддерживать аспекты исполнения и качества на уровне сервисов. self-service позволяет организациям создавать эластичное окружение, которое является расширяемым и основывается на параметрах больших нагрузки параметрах целевого исполнения. Pay-by-use- может предоставлять различные виды аренды оборудования, минимальный уровень обслуживания(услуг) которого гарантируется провайдером.

Виртуализация является ключевым моментом этих моделей. IT организации имеют представления о том, что виртуализация позволяет быстро и просто создавать копии существующего окружения, иногда вовлекая сложные виртуальные машины (поддержку тестов, разработки и перераспределение активности) Стоимость этих окружений минимальна, потому что они могут сосуществовать в окружении одного и того же production server-a, потому что они используют несколько ресурсов. Приложения здесь являются масштабируемыми и имеют возможность быстрого удаленного доступа.

Инфраструктура программируемая

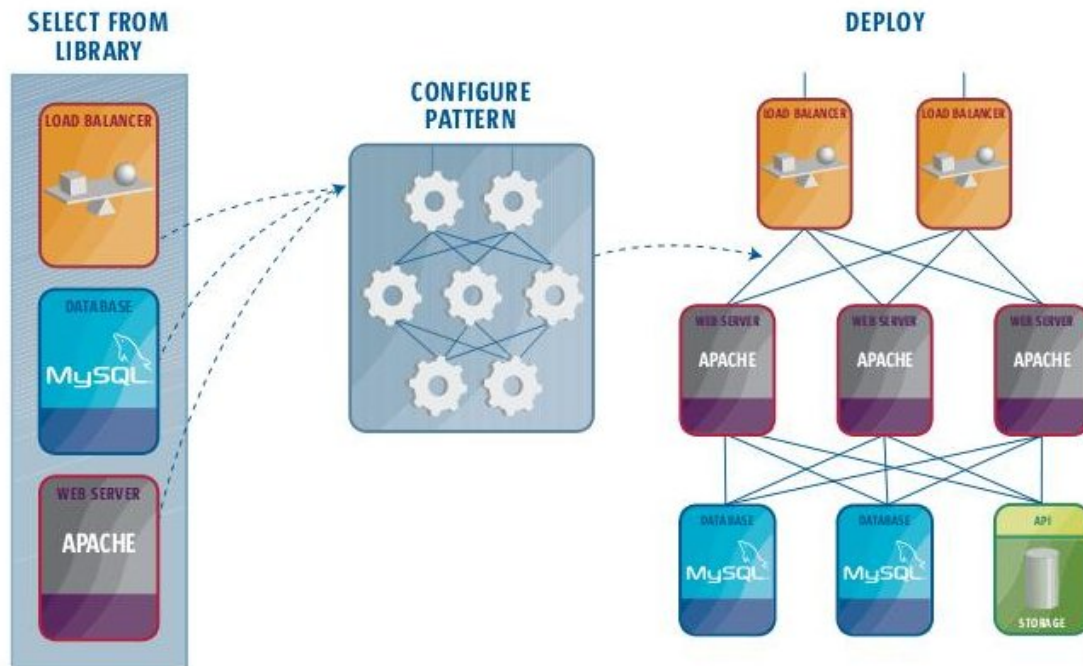
Раньше архитекторы должны были определить каким образом различные компоненты будут располагаться на серверах, как они будут связаны между собой, должны были обеспечить

определенную восстанавливаемость и масштабируемость. Сейчас разработчики могут использовать cloud API не только для создания структур приложений на виртуальных машинах, но так же изменять их при изменении рабочей нагрузки.

Раньше для обеспечения многопоточного параллельного выполнения пользовались языком java, теперь существует возможность одновременного создания нескольких взаимосвязанных машин. Очень важно смещение от архитектора к разработчику-архитектору.

Пример развертывания web-приложений

Пример как комбинация виртуализации и self-сервиса способствуют развертыванию приложений, будем считать, что два связанных приложения развертываются в облаке. 1. Разработчик должен выбрать балансировщик загрузки, Web служб и серверов БД из библиотеки с заданной конфигурацией образа виртуальной машины 2. Разработчик должен сконфигурировать каждый компонент для получения требуемого образа. Балансировщик загрузки должен быть настроен. web- сервер заполняется этими статистическими данными для загрузки их на storage cloud, приборы сервера приложений заполняются динамическим контентом для сайта. 3. Разрабатываемый код уровня надстройки в новой архитектуре соответствует требованиям специфических приложений. 4. Разработчики выбирают шаблоны, которые берут образы для каждого уровня и разворачивают их, обрабатывают вопросы масштабирования, безопасности и сетевые. 5. Безопасности, высокий показатель надежности веб-приложений растет. Когда приложение нуждается в обновлении, образы виртуальных машин могут быть обновлены. Cloud computing принимает за истину то, что все временно и проще переразвернуть все приложение, нежели ставить кучу патчей на виртуальные машины.



Сервисы предоставляемые через сеть

Почти само собой разумеется, что облачные вычисления имеют тенденцию расширения поскольку услуги становятся доступными через сеть. Практически каждая бизнес-организация использует веб-интерфейсы в своих приложениях, для возможности доступа клиентов к этим приложениям через интернет, или же они являются внутренними приложениями, которые становятся доступными для уполномоченных сотрудников, партнеров, поставщиков, и консультантов. Красота Интернет-услуг на базе, конечно, является то, что приложения могут быть доступны в любом месте и в любое время. Хотя предприятия, хорошо осведомлены о возможности безопасной связи с помощью Secure Socket Layer (SSL) шифрования, наряду с сильной аутентификацией, увеличение доверия к вычислительной среде облака требует тщательного изучения различий между компьютерной техникой предприятий и облачными вычислениями. При правильной архитектуре доставка через интернет может обеспечить необходимые гибкость и безопасность, предприятиям всех размеров.

Роль открытого программного обеспечения

Программное обеспечение с открытым исходным кодом позволяет приложениям, созданным Animoto, масштабироваться до 3500 случаев в течение нескольких дней. Легкость компонентов с открытым кодом может быть использована для сборки больших приложений, это в свою очередь увеличивает роль открытого программного обеспечения, например алгоритм

MapReduce, который может быть запущен в окружении cloud computing был одним из факторов стимулирования разработчиков.

Модели инфраструктур в cloud computing

Существуют три вида моделей: public, private, hybrid clouds public clouds запускаются тремя сторонами и приложения от различных клиентов похожи на смесь cloud's servers, storage systems, и networks.

одним из преимуществ public cloud -это то, что они гораздо больше нежели private у частных компаний, и предлагают возможности масштабирования вверх и вниз по требованию. Части public cloud могут быть вырезаны для эксклюзивного использования единичными клиентами, создавая виртуальные приватные центры данных. Теперь клиенты могут манипулировать не только виртуальными машинами, но и серверами, системами хранения данных, сетевыми устройствами и топологией сети.

private cloud строится для эксклюзивного использования или для единичного клиента, предоставляет контроль над всей информацией. Инфраструктура компании собственника имеет контроль над тем, как приложение разворачивается на ней. Private cloud может располагаться в датацентре, компании так же могут быть за просто развернуть приложение.

Частные облака строятся исключительно для использования одного клиента, обеспечивая максимальный контроль над данными, безопасности и качества обслуживания. Компания владеет инфраструктурой, а также контролирует размещенные на ней приложения. Частные облака могут быть размещены в центре обработки данных предприятий, и они также могут быть развернуты размещенном на объекте. Частные облака могут быть созданы и управляются собственными ИТ-организациями предприятия или облаком услуг. Эта модель дает компаниям высокий уровень контроля над использованием ресурсов.

Смешанные облака Это комбинация private и public моделей. Возможность увеличить частный облака за счет публичных может быть использована при быстрых колебаниях рабочей нагрузки(поддержка приложений Web2.0). Гибридная модель может так же использоваться при запланированной рабочей нагрузке. Здесь среди вопросов, которые надо будет рассмотреть -это отношение между данными и обработкой данных.

Архитектурные уровни Cloud Computing

From: <http://wiki.osll.ru/> - Open Source & Linux Lab

Permanent link: http://wiki.osll.ru/doku.php/etc:users:kea:introduction_to_cloud_computing_architecture?rev=1265325649

Last update: 2010/02/05 02:20

