Above the Clouds: A Berkeley View of Cloud Computing

cloud computing, berkeley

Michael Armbrust Armando Fox Rean Griffith Anthony D. Joseph Randy H. Katz Andrew Konwinski Gunho Lee David A. Patterson Ariel Rabkin Ion Stoica Matei Zaharia

Electrical Engineering and Computer Sciences University of California at Berkeley Technical Report No. UCB/EECS-2009-28

http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

Тезисы

Utility Computing. Это понятие подразумевает отношение к потреблению ИТ-ресурсов как к получению электричества из розетки или воды из крана. То есть необходимые аппаратные мощности выделяются по требованию из облака. Когда мощности не требуются, они не потребляются пользователями, а соответственно, деньги за них не взимаются.

Software as a Service (SaaS). Это фундаментальное понятие Cloud Computing подразумевает изменение модели доставки программного обеспечения пользователям таким образом, что компания прекращает покупать программное обеспечение как коробочные или заказные продукты, а начинает “потреблять” его из облака на основе подписки. Типичный пример – это электронная почта Gmail от Google, которой пользуются многие компании, не тратя при этом на оборудование для почтовых серверов, серверное ПО и администраторов.

С аппаратной точки зрения, три аспекта являются новыми в Cloud Computing.

1. Иллюзия бесконечных вычислительных ресурсов, доступных по запросу, тем самым устраняя необходимость в cloud computing

предлагающая пользователям планировать заранее выделение ресурсов.

1. Ликвидация первоначальных обязательств пользователей, что позволило компаниям начать с малого и увеличивать аппаратные ресурсы только при увеличении потребностям в них.
2. Возможность платить за использование вычислительных ресурсов на краткосрочной основе по мере необходимости и освобождать, когда необходимость в них исчезает, тем самым происходтит консервация, позволяя машинам и носителям данных быть использованными другими пользователями.

Azure и Amazon

Смысл Azure в том, что теперь создать интернет-сервис для .Net разработчика станет совсем просто. Для этого нужно всего лишь создать работающее приложение в среде .Net. А о том, чтобы это приложение заработало (и исправно работало) на интернет-сервере разработчику заботиться не нужно.

В отличии от Amazon Web Services, которая предоставляет только инфраструктуру (процессор, память и операционную систему) и перекладывает на плечи разработчиков многие заботы,

связанные с масштабирование приложений, установкой вспомогательного ПО,Azure позволит концентрироваться только на создании бизнес-логики. Например, СУБД (MS SQL Server), сервисы для совместного доступа к файлам уже установлены и настроены, а масштабирование производится за пару кликов. Azure является промежуточным звеном между между такими фреймворками приложений как : AppEngine и аппаратными виртуальными машинами, как EC2.

Когда Utility Computing предпочтительным запуску частного Cloud?

1. Когда спрос на услуги меняется со временем (например когда пиковые нагрузки всего несколько дней в месяц, в остальное время ресурсы простаивают)
2. Когда спрос заранее известен.(например web приложение, когда оно находится на пике популярности)
3. И, наконец, это полезно для пользователя которому нужно в короткие срокипроизвести объемные вычисления.

Для поставщика Cloud Computing начальные затраты достаточно велики. Начнем с того, что мультиплексирование и нагрузка требуют очень больших центров обработки данных.Строительство, снабжение и запуск такого механизма требует около ста миллионов долларов . Однако из-за феноменального роста услуг через веб-начале 2000, AOS, многие крупные компании в Интернете, включая Amazon, eBay, Google, Microsoft и другие, уже являются поставщиками.

Не менее важно, что компаниям так же необходимо разрабатывать масштабируемую инфраструктуру программного обеспечения (таких, как MapReduce, Google File System, BigTable) и защиту своих центров обработки данных от возможных физических и электронных атак.

Необходимо, но не достаточным условием для того, чтобы компания стала поставщиком Cloud Computing является то, что она должна иметь существенные инвестиции не только в очень большие вычислительные центры, но и обладать масштабируемой инфраструктурой программного обеспечения и опытом, необходимым для ее работы.

Продвижению Cloud Computing способствуют новые тенденции и бизнесс модели,а так же новые возможности приложений:

1. Интерактивные приложения- приложения, доступные с мобильных устройств в режиме реального времени).
2. Параллельная пакетная обработка: обработка большин объемов данных и ресурсоемких вычислительных задач.
3. Особым подвидом является бизнесс аналитика(анализ статистики связанной с продажами, спросом, отслеживание цепочек поставок и т д)

Классы utility computing

Различные предложения utility computing будут основываться в основном на уровнях абстракции предложенных программисту и уровнях управления ресурсами. выделяют следующие классы:

1. Amazon Web Services (I)
2. Microsoft Azure (II)
3. Google AppEngine (III)

ниже приведена сравнительная характеристика этих классов.

Модель вычислений

- (I) Основывается на инструкциях архитектуры x86 под управлением VM XEN
- (II) Common Language Runtime VM
- (III) Предопределенные структуры приложений и фреймфорков. Необходимо вручную писать скрипты на Python, чтобы получить доступ к web интерфейсам. Преимущество-автоматическое масштабировани

Модель хранения:

- (I) SimpleDB
- (II) SQL Data Services
- (III) MegaStore/BigTable

Сетевая модель:

- (I) декларативной спецификации IP на уровне топологии; скрыты детали внутреннего размещения securisy group позволяет установить, какие узлы могут общаться между собой.
- (II) Основывается на декларативном описании программиста
- (III) Фиксированная топология для размещения трехуровневой структуры web приложений. Автоматическое масштабирование в обе стороны. Программист невидим.

10 препятствий и возможностей для Cloud Computing

1. Необходима постоянная и бесперебойная доступность услуг.
2. Трудность извлечение данных из облака вследствие нестандартизованности API
3. Сохранение конфеденциальности данных в частных облаках
4. Необходимо уменьшение затрат на передачу больших объемов данных(например путем предоставления дисков хранения)
5. Непредсказуемая производительность . Одной из возможностей является улучшение архитектур и операционных систем для эффективной виртуализации прерывания и каналов ввода / вывода.
6. Масштабируемость хранения. Необходима система хранения данных, хоторая не только будет удовлетворять всем потребностям пользователя БД, но и сочетать их с облаком.
7. Устранение ошибок в больших распределенных системах.
8. Необходимо быстрое масштабирование в целях экономии занимаемого времени и соответственно денежных ресурсов.
9. Распространение репутации. Например плохая работа одного клиента, может повлиять на репутацию облака в целом .
10. Лецинзирование программного обеспечения(это достаночно дорогое удовольствие,

поэтому многие поставщики cloud computing прибегают к ПО с открытым исходным кодом.)

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

Permanent link:
http://wiki.osll.ru/doku.php/etc:users:kea:berkleyview

Last update: **2016/08/08 20:53**