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 на 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 расработчиками. Использование виртуальных машин для развертывания объектов достаточно в 80 процентах случаев использованияи 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 образоа виртуальных машин, исходных файлов для компонентов таких как веб-серверы, информация о состоянии приложений и важная бизнесс информация.

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

Все эти три модели,являющиеся первоначальными для CC так же рамширяются в условиях этих тенденций. Модель on-demand помогает поддерживать аспекты исполнения и качества на уровне сервисов. self-service позволяет организациям создавать эластичное окружение, которое являеся расширяемым и основывается на параметрах больших нагрузоки 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 окужений минимальна, потому что они могут сосуществовать в окружении одного и того же production server-а, потому что они используют несколько ресурсов. Приложения здесь являются масштабируемыми и имеют возможность быстрого удаленного доступа.

Инфраструктура программируемая

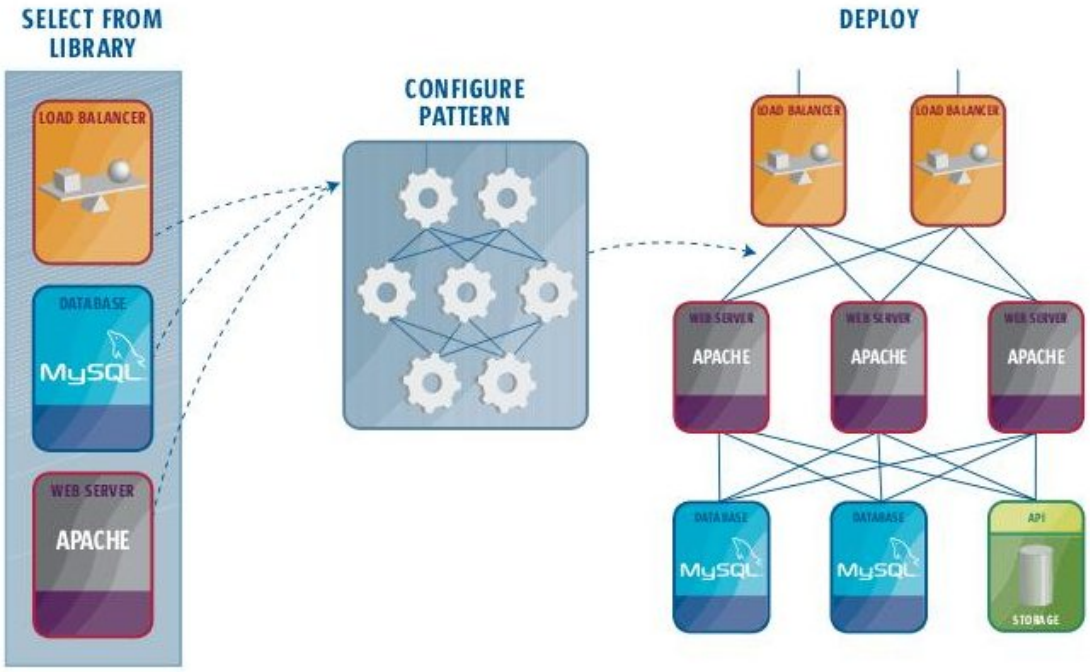
Раньше архитекторы должны были определить каким образом различные компоненты будут располагаться на серверах, как они будут связаны между собой, должны бвли обеспечить

определенную восстанавливаемость и масштабируемость.Сейчас разработчики могут исользовать 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 взаимосвязных машин. Очень важно смещение от архитектора к разработчику-архитектору.

Пример развертывания web-приложений

Пример как комбинация виртуализации и self-сервиса способствую развертыванию приложений, будем считать , что два связных 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 сконфигкрировать 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 специфичнеских приложений. 4. Разработчики выбирают шаблоны, которые берут образы для каждого уровня и разворачивают их, обрабатывают вопросы масштабирования, безопасности и сетевые. 5.Безопасности, высокий показатель надежночтии 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, нежеди ставить кучу патчей на виртуальные машины.



Сервисы предоставляемые через сеть

Почти само собой разумеется, что облачные вычисления имеют тентенцию расширения поскольку услуги становятся доступными через сеть. Практически каждая бизнес-организация использует веб-интерфейсы в своим приложениям,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 архитектурае доставка чрез интернет может обеспечить необходимы 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 нескаольких дней. Легкость компонентов с открытым кодом может быть использована для сборки больших приложения,это в свою очередь увеливичает 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 м приложения от различных клиентов похожи на смесь cloud’s servers, storage systems, и networks.

одним из преимуществ public cloud -это то, что они горвздо больше нежели private у частных компаний, и предлагают вохможности масштабирования вверх и вних 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.

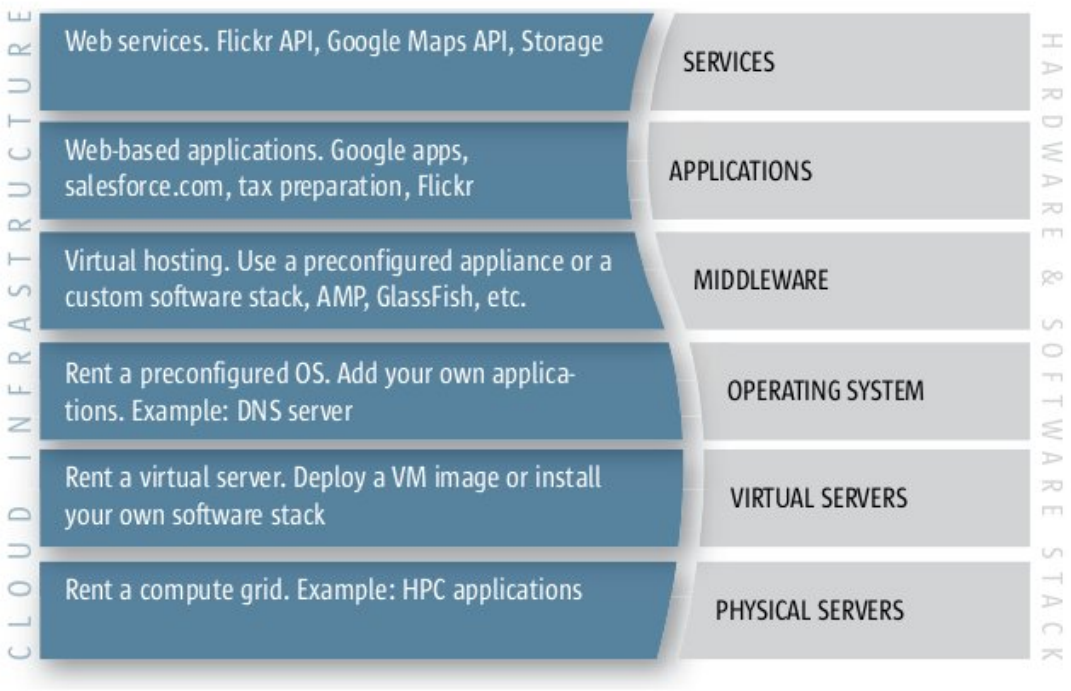
private cloud истроится для экслюзивного использования или для единичного клиента,предоставляет контроль над всей информацией. Инфраструктура компании собственниика имеет контроль на тем, как приложение разворачивается на ней. Private cloud может располагаться в датацентре ,компании так же можут быть запросто 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 приложенияю. 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 использования при быстрых колебаниях рабочей нагрузки(поддержка приложений Web2.0).Гибридная модель может так жи использоваться при запланированной рабочей нагрузке.Здесь среди вопросов, которые надо будет расспотреть -это отношение между жанными и обработкой данных.

Архитектурные уровни Cloud Computing

Cloud computing предлагает услуги, которые могут быть разделены на три катрегории:software as a service, platform as a service и infrastructure as a service. Общая структура соответствия аппаратного обеспечения и программного приведена ниже:



интерфейсы программирования приложений в cloud

Одной из ключевых характеристик, отличающих cloud computing стандартного-это то, что инфраструктура является программируемой. Вместо физического размещения серверов, хранилищ и сетевых ресурсов для поддержки приложений,разработчики определяют какие идентичные виртуальные компоненты настраивать и связывать , включая то, как образы виртуальных машин и данные приложений будут получаться и храниться в cloud storage. Они определяют как и когда компоненты будут развернуты через API, которые специфицируются cloud провайдером.

Это аналогично тому,как работает File Transfer Protocol (FTP) : FTP серверы контролируют связи с клиентом, которые остаются открытыми в течение всего срока сессии. Когда файлы должны быть переданы, управляющее соединение используется для обеспечить источника или имени файла на сервере, и обращаясь к источнику по нужному порту происходит передача фалов. В некотором смысле, “cloud computing API похож FTP управления каналом: он открыт в течение всего срока использования облаке, и он определяет, насколько используется облако в конце предоставить услуги предусмотренные разработчиками. Использование API для контроля как использующейся облачной инфраструктуры таит в себе подвозные камни. В отличие от ftp протокола cloud API не стандартизованы, что является естественным для любой отрасли, находящейся на началоной стадии развития.

Преимущества Cloud Computing

Использование cloud computing имеет множество преимуществ:

- сокращение времени запуска и времени отклика.

(достаточно быстро решаются ресурсоемкие задачи, что и уменьшает временные затраты)

- минизация рисков инфраструктуры.

(это очень удобно для малых компаний, когда неизвестен итоговый результат и нет желания тратить большие ресурсы на прибретение дорогостоящего оборудования)

- снижение стоимости.

Есть ряд признаков Cloud Computing, который позволяет снизить затраты:

1. Аренда инфораструктуры(нет затран на преобретение)
2. Приложения разрабатываются в основном путем сборки а не программирования, что увеличивает темпы развертывания новых приложений.

- увеличение темпов инноваций.

Cloud Computing может способствовать повышению темпов инновационной деятельности. Низкая стоимость вступления на новые рынки, создает условия, позволяющие старт-ап компании развертывать новые продукты быстро и при низких затратах. Это позволяет мелким компаниям более эффективно конкурировать с традиционными организациями.

Архитектурные вопросы IaaS

Изменяются архитектурные подходы

В 90-х годах заше разговор, о том,как декомпозировать приложения на различные компоненты и далее как разворачивать эти приложения на отдельных серверахв порядке оптимизации нефункциональных требования таких как масштабируемость,доступность , легкость управления и безопасность. На сегодняшний день поддерживается декомпозиция архитектур приложений, а в итоге развертывается на объединенной архитектуре, которая использует виртуализацию. Cloud Computing продолжает идти по пути предоставления программируемого развертывания архитектур приложений, в конечном счете обещается предоставление динамических data центров. С cloud computing резко повышается эффективность. Если приложение нельзя сделать быстрым и программируемым, значит оно не соответствует модели.

Изменяется дизайн приложений

В прошлом, приложения строились для обработки больших нагрузок путем вертикального масштабирования, то есть добавления больших физических ресурсов на машину, сейчас

произошел переход к горизонталдьному масштабированию, распределенным вычислениям.

Высокая производительность

Кластерные вычисления, позволяют в разы увеличивать производительность приложений,cloud computing явяется прямым потомком grid технологий.

Системы управления базами данных

Системы управления базами данных адаптированны для работы в облачных средах путем горизонтального масштабирования серверов баз данных и таблиц разделов между ними. Эта техника, известна как sharding, позволяет нескольким экземплярам БД(часто MySQL)масштабировать производительность в среде облака. Вместо того, чтобы иметь доступ к единой центральной БД, приложения теперь имеют доступ к одним из многочисленных БД, в зависимости от той чести данных, в коотрых они нуждаются.

CPU обработка

Приложения, ориентированные на активность типа frame rendering, теперь вместо создания отдельного потока могут создвавть отдельную виртуальную машину, что увеличивает производительность за счет горизонтального масштабирования.

Обработка данных

Основные инструменты-инструменты развивающиеся open source сообществом, которые помогают в обработке больших объемов данных и далее сливают результаты в единый процесс обработки. Например Hadoop это open source разработка для фреймворка MapReduce которая совмещает развертывание рабочих виртуальных машин с данными, которые оин используют.

Последовательный и стабильный слой абстрагирования

Cloud computing увеличивает уровень абстракции на столько, что все компоненты являются абстрактными или виртуализироваными,и могут быть использованы для быстрой компановки приложений и платформа более высокого уровня.Если компонент не является последовательныим и устойчивым слоем для клиента иди узла он является не подходящим для cloud computing. Стандартная единица развертывания -это виртуальная машина,которая по своей природе предназначен для работы на абстрактном уровне аппаратной платформы.В Cloud Computing важно поддержвать модель, а не образесли поддерживается модель, образ получается из модели. Образ виртуальной машины будет всегда меняться, потому что уровни программного обеспечения всегда будут нуждаться в патчах, обновлениях и реконфигурациях. Что не изменяется, так это процесс создания виртуальной машины и это то, на чем разработчики должны сфокусировать свое внимание. Разработчики должны стоить виртуальную машину в следующей иерархии(последовательности) web-сервер, сервер

приложения, сервер БД MySQL на образе ОС, установка патчей, изменения и общих компонентов каждого уровня. Фокусирование на модели (лучше чем образ виртуальной машины) позволяет обновлять по необходимости, по мере добавления в модель новых наборов компонентов.При наличии стандарртной единици развертывания cloud архитекторы могут использовать “приборы”, коотрые помогают обеспечить скорость развертывания при низкой стоимости. Разработчики должны использовать “приборы”, коотрые предварительно настроены для запуска Hadoop на OpenSollaris OS,заимодействую через appliance’s API. Архитекторы могут использовать содержимое коммутаторов для развертывания не как физических устройств, а как виртуальных “приборов”(устройств). Все что необходимо для развертывания-это вщаимодействие с API или GUI.

Стандарты, помогающие решить трудности

Cloud Computing имеет несколько стандартов и стандартных конфигураций,направленных на сокращение стоимости обслуживания и развертывания. Есть стандарты, которые делают процесс развертывания проще, это более важно, нежели наличие совершенно настроенной окружающей среды для работы. Здесь действует правило 80/20:CC фокусируется на нескольких стандартов, которые поддерживают 80% всех случаев применения. Стандарт вероятно будет включать тип виртуальной машины, опрационную систему в стандарте образа виртуальной машины и поддержку языков программирования:

- Типы виртуальной машины.Применительно к социальным сетям, изоляции для безопасности и высокий уровень абстракции для портирования рекомендуется использовать TypeII виртуальной машины. Для высокой производительности и визуализации приложений, непосредственно для доступа к аппаратуре для достижения максимальной производительности необходимо использовать TYPEI виртуальных машин.
- Предварительно устанавливаемы и настраиваемые системы. программное обеспечение на виртуальных машинах должно поддерживаться так же как и на физических серверах. Имеется небольшой набор стандартов для поддержки конфигураций, разрешающих расработчикам использовать текущую поддерживаемую VM.

Когда текущая конфигурация обновляется, модель, диктующая настройки должна быть разработана так,чтобы было легко вносить изменения в новую виртуальную машину. То же самое должно быть верно для устройств, где текущая версия может быть настроена с помощью стандартных интерфейсов API.

- Инструменты и языки. Предприятия должны стандартизировать использование языков java и Ruby on Rails. Малый бизнес может стандартизировать PHP как предпочтительный инструмент для построения приложений. Поскольку эти стандарты достаточно старые в контексте CC, они начинают формировать следующий слой, PaaS(платформа как сервис).

Системы виртуализации и инкапсуляции поддерживают рефакторинг

Кода над приложением производится рефакторинг и создаются путем комбинирования и настройки набор образов виртуальных машин и инструментов”приборов“, акцент делается на том, что конкретная виртуальная машина работает, но не знает как это реализовано.Виртуализация и инкапсуляция скрывают детали реализации и фокусируют разработчиков на интерфейсах и межкомпонетным взаимодействием.Эти компоненты должны

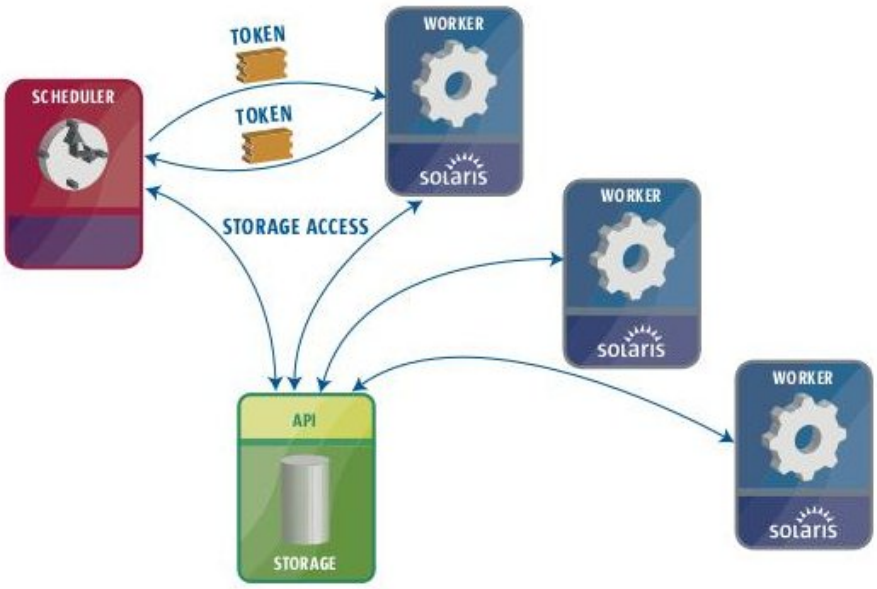
предоставлять стандартные интерфесы для того, чтобы разработчики могли быстро и просто разрабатывать приложения-это как использование тех или иных альтернативных компонентов с похожей функциональностью в зависимости от производительности и скорости. Развертывание приложений производится программно и даже программы развертывания приложений могут быть инкапсулированы и поэтому могут использоваться повторно. Философия облачных вычисления для поодержки приложений -не устанавливать патчи, а переразвертывать. Управление моделью, которая создает виртуальные образы машин упрощает эту процедуру. После развертывания или выпуска новых версий машин путем обновления компоненты виртуальных машин применяют шаблон проектирования для переразвертывания. Когда разработчики дорабатывают виртуальную машину, только один образ виртуальной машины должен быть создан, остальные должны быть тиражированы и развернуты программно. VM должна иметь версию для облегчения отката в случае необходимости.

Горизонтальное масштабирование

Используя горизонтальное масштабирование следует сосредоточить внимание на общей доступности приложенийс предволожением, что любой из компонентов в люой момент времени может быть не доступен.Большинство платформ облака строятся на виртуальных пулах серверных ресурсов. Горизонтальное расширение не дожно быть ограниченным одним облаком.

Параллелизация

При вертикальном масштабировани , при многопроцессорной машине переллелизация позволяла увеличить скорость выполнения операций. Но сегодня, когда вычислительные среды смещаются к архитектуре x86 с двумя или четырьмя сокетами, только вертикальное масштабирование имеет много параллельных процессов на ядрах серверов. На макроскопическом масштабе, программное обеспечение, которое может использовать распараллеливание на многих серверах может масштабироваться до тысяч серверов, предлагая огромную масштабируемость, это стало возможным благодаря симметричной многопроцессорной обработке. В физическом мире, распараллеливание часто реализуется с помощью балансировки нагрузки или содержимым коммутаторов, которые распределяют входящие запросы на несколько серверов. Есть много других способов использования параллельной обработки в среде облачных вычислений. Приложение, которое использует значительное количество процессорного времени для обработки пользовательских данных может использовать модель приведенную ниже на рисунке. планировщик получает задания от пользователей,располагает даные в репозитории и далее запускает новую виртуалдьную машину для каждой задачи,предоставляя виртуальной машине маркер, который позволяет получить данные из хранилища. Когда виртуальная машина заканчивает выполнение своего задания, результат последовательно возвращается обратно планировщику,которые последовательно завершает проект и передает результать пользователю и работы виртуальной машины завершается.



Разделяй и властвуй

приложение могут быть распараллелены если даные могут быть разделены по независимым системам, которые будут обрабатывать их параллельно. Хорошая архитектура приложений включает в себя план разделяй и влавствуй применительно к данным и имеет реальном мире наглядные примеры и множество подходов:

- Hadoop - это реализация MapReduce шаблона, который реализует master/worker паттерн распараллеливания.
- БД sharding может быть реализована путям ряда техник разделения, включающих вертикальное разделение,диапазонное разделение или разделение на основе дирректорий. Подход полностью зависит от того, какие данные должны быть использованы.
- Большинство финансовых институтов совершенствуют алгоритмы выявления мошенниченства, одни из них -это data-mining операции, которые можно распараллелить применительно к большим объемам дынных.
- Другие высокопроизводительные приложения, которые работают с трехмерной информацией были разработаны так, что состояние одного кубического объема (например газа) может быть вычислено за время t для одного процесса. Далее состояние одного куба-распространяется на состояние восьми прилегающих кубов, которыы вычисляются за всремя t+1.

Разделение данных имеет значение, когда объемы данных передаются по сети, что делает физические данные следующими в списке рассмотрения.

Физика данных

Физика данных рассматривает связь между элементами процессов и данными, которыми они оперируют. поскольку большинство данных хранятся в облаке, а не на локальных физических серверах, нужно время, чтобы доставить эти данные на сервер. Большие объемы данных и низкая пропускная способность канала удлиняет время, необходимое для перемещения данных.

time=(bites*8)/bandwidth;

Она очень часто помогает определиться с целесообразность принятия решений, например актуально ли перемещать данные из одного публичного облака в другое и т д

Взаимоотношение между данными и процессами

Перемещение данных из хранилищ для обработки может потребовать большое количество времени и денег Некоторые аспекты этих взаимоотношений необходимо рассмотреть:

1. дыннын хранящиеся без вычислительных мощностей поблизости имеют ограниченную ценность.И облако должнго быть прозрачном в сетевом плане:должны быть известны размер каналов,какова латентность,какова надежность соединенияю. Cloud провайдерам предствоит ответить на все эти вопросы.
2. Cloud архитекторы должны иметь способность четко указать место расположение виртуальных машин и список услуг которые определяют отношения между вируальными машинами и доступом к хранилищу.
3. Cloud провайдеры должны оптимизировать эти отношение для клиентов, но так же и равссмотреть вопрос о целесообразности оптимизации эхтих отношений вручную.
4. иногда более выгодно расчитать значение, нежеди доставать его из сетевого хранилища. Необходим компромис.

программные стратегии

cloud стратегии как прпавило руководствуются следующими правилами: 1. Перемещение указателей как правило лучше, чем перемещение фактических данных. 2. указатели должны рассматриватся как предосторожность, необходимая для того, чтобы труднее было подделать данные. 3. Объектный доступ. Протокол SOAP.