



Cloud Computing

*Новый способ предоставления
вычислительных ресурсов*

Основы облачных вычислений

План

- **Основы облачных вычислений**
- **Три уровня облачных вычислений**
- **Проблемы миграции в облако**
- **Безопасность в облаке**

Введение

- Революция определяется как изменение способа мышления и поведения людей, существенное по природе и широкое по охвату.
- Если следовать этому определению, то облачные вычисления несомненно являются революцией. *Облачные вычисления фундаментально изменяют архитектуру компьютеров, разработку программного обеспечения и инструментальных средств и, конечно же, способы хранения, распределения и потребления информации.*
- Цель данной лекции - помочь вам примириться с реальностью революции и взять управление изменениями на себя и использовать их во благо себе и своей организации.

Новая парадигма в ИТ

- За последние несколько лет в отрасли информационных технологий (ИТ) получила развитие новая парадигма – облачные вычисления. Хотя облачные вычисления – это всего лишь *особый способ предоставления вычислительных ресурсов*, а не новая технология, они вызвали революцию в методах предоставления информации и услуг.
- Первоначально в ИТ доминировали мейнфреймы. Со временем эта жесткая конфигурация уступила дорогу клиент-серверной модели. Современная ИТ-отрасль становится все более мобильной, всепроникающей и, естественно, облачной. Но эта революция, как и любая другая, содержит старые компоненты, из которых она эволюционировала.
- Таким образом, чтобы правильно понимать облачные вычисления, нужно помнить, что они, по существу, генетически наследуют предшествующие системы. Во многих отношениях переломное изменение означает "назад в будущее", а не окончательное завершение прошлого.
- В этом новом мире облачных вычислений есть место для инновационного кооперирования **облачной технологии** и проверенной эффективности предшествующих систем, таких как мощные **мейнфреймы**.

Что такое облачные вычисления

- Облачные вычисления – это комплексное решение, предоставляющее ИТ-ресурсы в виде сервиса.
- Это основанное на интернет-технологиях решение, в котором ресурсы общего пользования предоставляются аналогично распределению электроэнергии по проводам.
- Компьютеры в облаке настроены на совместную работу, а различные приложения используют совокупную вычислительную мощность так, как будто выполняются на одиночной системе.
- **Гибкость облачных вычислений** зависит от возможности **распределения ресурсов по требованию**. Такое распределение позволяет использовать совокупные ресурсы системы без выделения конкретных аппаратных ресурсов определенной задаче.
- До облачных вычислений Web-сайты и серверные приложения выполнялись *на отдельно взятых системах*. С приходом облачных вычислений ресурсы используются как *объединенный виртуальный компьютер*. Такая объединенная конфигурация предоставляет среду, в которой приложения выполняются независимо без привязки к какой-либо конкретной конфигурации.

Зачем спешить в облако

Для перехода к парадигме облачных вычислений есть веские причины - как с точки зрения бизнеса, так и с точки зрения ИТ.

- **Уменьшение расходов.** Облачные вычисления могут снизить как капитальные затраты (CapEx), так и текущие расходы (OpEx), поскольку ресурсы приобретаются только по необходимости и оплачиваются только по использованию.
- **Оптимальное использование персонала.** Использование облачных вычислений высвобождает ценных сотрудников, позволяя им сконцентрироваться на увеличении прибыли, а не на поддержке аппаратного и программного обеспечения.
- **Надежная масштабируемость.** Облачные вычисления обеспечивают мгновенное масштабирование вверх или вниз в любое время без долгосрочных обязательств.

Компоненты облачных вычислений

- Модель облачных вычислений состоит из **внешней** (front end) и **внутренней** (back end) частей.
- Эти два элемента соединены по сети, в большинстве случаев через Интернет. Посредством внешней части пользователь взаимодействует с системой; внутренняя часть – это собственно само облако.
- Внешняя часть состоит из клиентского компьютера или сети компьютеров предприятия и приложений, используемых для доступа к облаку.
- Внутренняя часть предоставляет приложения, компьютеры, серверы и хранилища данных, создающие облако сервисов.

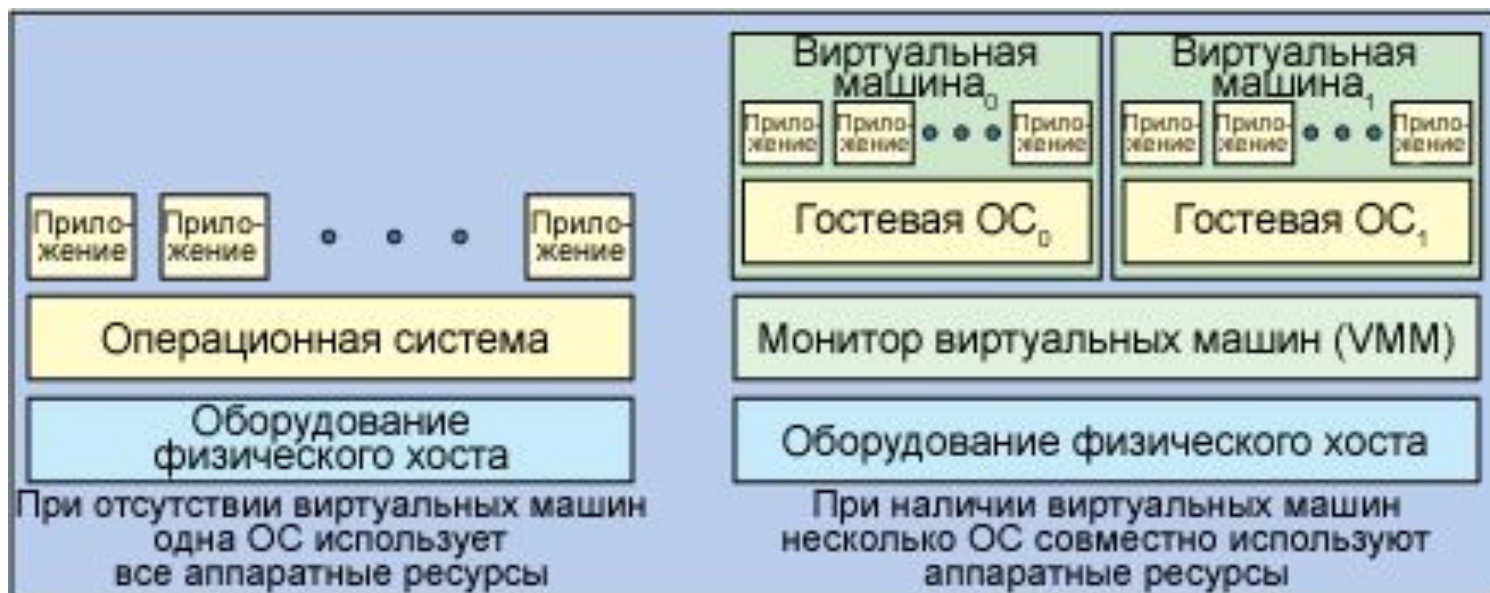
Уровни: вычисления как коммунальный ресурс

- Концепция облака основана на **уровнях**, каждый из которых предоставляет определенную функциональность.
- Такая стратификация компонентов облака позволяет сделать уровни облачных вычислений коммунальным ресурсом, аналогичным электричеству, услугам телефонии или природному газу.
- Товар "облачные вычисления" - это более дешевые и менее затратные для пользователя вычислительные ресурсы. У облачных вычислений есть все шансы стать еще одним коммунальным ресурсом.

Монитор виртуальных машин (virtual machine monitor - VMM)

- или *гипервизор (hypervisor)* предоставляет средства для одновременного использования функциональных возможностей облака (см. [рисунок 1](#)).
- VMM – это программа, выполняющаяся на хост-системе и позволяющая одному компьютеру поддерживать несколько сред исполнения программ. С точки зрения пользователя система представляет собой автономный компьютер, изолированный от других пользователей. В действительности все пользователи обслуживаются одним и тем же компьютером.
- Виртуальная машина – это одна операционная система (ОС), управляемая основной контролирующей программой, которая представляет ее в виде нескольких операционных систем.
- При облачных вычислениях VMM предоставляет пользователям возможность отслеживать и, следовательно, управлять такими аспектами процесса, как *доступ к данным, хранение данных, шифрование, адресация, топология и перемещение рабочей нагрузки*.

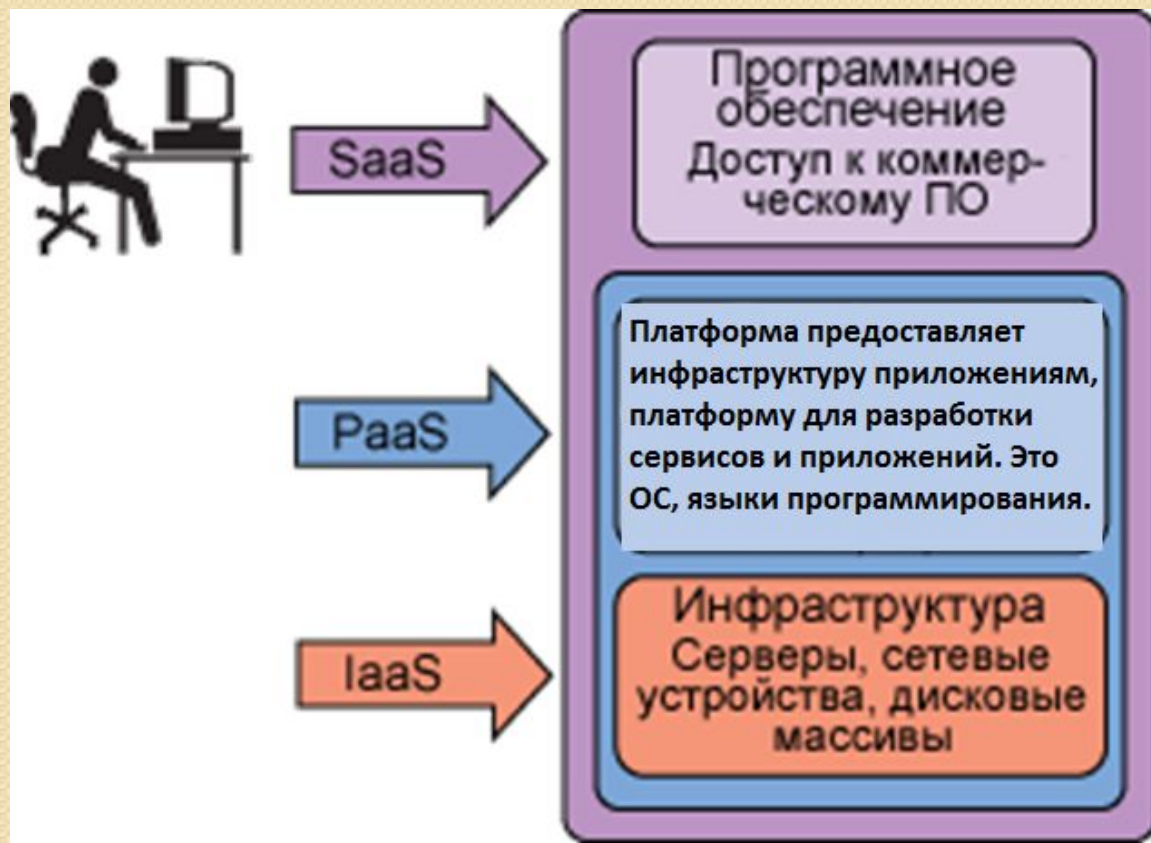
Рисунок 1. Как работает монитор виртуальных машин



Облако предоставляет следующие уровни обслуживания:

- Уровень *инфраструктуры* (Infrastructure as a Service – IaaS)
- Уровень *платформы* (Platform as a Service - PaaS)
- Уровень *приложений* (Software as a Service - SaaS)

РИСУНОК 2. УРОВНИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ, ВСТРОЕННЫЕ В КОМПОНЕНТЫ "КАК СЕРВИС"



Уровень инфраструктуры –

- это основа облака. Он состоит из физических активов – серверов, сетевых устройств, дисков и т.д. Существуют поставщики *инфраструктуры как сервиса* (Infrastructure as a Service - IaaS), например IBM.
- При взаимодействии с IaaS вы в действительности *не управляете базовой инфраструктурой, однако управляете операционными системами, хранилищами данных, развертываемыми приложениями и, до определенной степени, выбранными сетевыми компонентами.*
- Примером организаций, которые могут получить выгоды от IaaS, являются сервисы печати по требованию (Print On Demand - POD). Модель POD основана на продаже товаров, дизайн которых задается в соответствии с требованиями клиента. POD позволяет физическим лицам открывать магазины и продавать дизайны товаров. Владельцы магазинов могут загрузить столько дизайнов, сколько будут в состоянии создать. Благодаря возможностям облачной системы хранения POD может предоставлять неограниченный объем дискового пространства.

Платформа

- является Промежуточным уровнем. Она предоставляет инфраструктуру приложениям. Платформа как сервис (Platform as a Service - PaaS) предоставляет доступ к операционным системам и соответствующим сервисам. Она дает способ создания и развертывания приложений в облаке при помощи языков программирования и инструментальных средств, поддерживаемых поставщиком. Вам не нужно управлять используемой инфраструктурой или контролировать ее, но у вас есть возможность управлять развернутыми приложениями и, до определенной степени, конфигурациями среды хостинга приложений. Существуют поставщики PaaS, например Elastic Compute Cloud (EC2) от Amazon.
- Идеальный пользователь PaaS – это небольшая компания по созданию программного обеспечения. Имея в своем распоряжении такую платформу, можно создавать продукты мирового класса без накладных расходов, свойственных разработке на собственных ресурсах.

Уровень приложений

- Это – верхний уровень, который обычно и изображают в виде облака. Приложения, выполняющиеся в нем, предоставляются пользователям по требованию.

Существуют поставщики программного обеспечения как сервиса (Software as a Service - SaaS), например, Google Pack.

Google Pack содержит доступные через Интернет приложения - Calendar, Gmail, Google Talk, Docs и многие другие.

Облачные структуры

- По характеру владения облачные структуры делятся на три типа: *закрытые (private)*, *открытые (public)* и *гибридные (hybrid)*.
- Открытые облака доступны широкой общественности или большой промышленной группе; они принадлежат и поддерживаются организацией, продающей облачные сервисы. Под "облаком" обычно подразумевается именно открытое облако; посредством Web-приложений **сторонний поставщик** динамически предоставляет через Интернет ресурсы совместного использования и выставляет счета в зависимости от их использования.
- Закрытые облака расположены за сетевым экраном предприятия и управляются этим предприятием. Это облачные сервисы, создаваемые и управляемые внутри предприятия. Закрытые облака предлагают в основном те же преимущества, что и открытые; основное их отличие состоит в том, что **ответственность** за настройку и поддержку закрытого облака несет **предприятие**.
- Гибридные облака – это комбинация открытого и закрытого облака, в которой используются **сервисы**, расположенные как в **открытом**, так и в **закрытом** пространстве. Ответственность за управление такими сервисами распределяется между **поставщиком** открытого облака и **предприятием**. Используя гибридное облако, организации могут определить цели и требования к создаваемым сервисам и получить их, основываясь на выборе наиболее подходящего варианта.

Быть или не быть облаку: оценка рисков

- Важнейшими вопросами для тех, кто собирается использовать облачные вычисления, являются безопасность и конфиденциальность. Компании, предоставляющие сервисы облачных вычислений, знают об этом и понимают, что без надежной защиты их бизнес разрушится. Поэтому защита и конфиденциальность имеют наивысший приоритет для всех субъектов облачных вычислений.

Общее руководство

- *Руководство (governance)* является важнейшей ответственностью владельца *закрытого* облака и совместной ответственностью поставщика и потребителя сервиса в *открытом* облаке.
- Однако, учитывая такие факторы, как международный терроризм, DOS-атаки, вирусы, черви и т.д., которые могут быть неподвластны владельцу закрытого облака или поставщику и потребителю сервиса открытого облака, существует необходимость в более широкой координации совместной деятельности, особенно на глобальном, региональном и национальном уровнях. Естественно, подобная координация совместной деятельности должна быть организована так, чтобы не ослаблять и не нарушать каким-либо образом контроль со стороны владельца процесса или абонентов открытого облака.

Требования к пропускной способности

- Если вы собираетесь внедрить инфраструктуру облачных вычислений, при разработке стратегии необходимо оценить пропускную способность и ее потенциальное узкое место.
- Проектировщики систем виртуализации в прошлом знали, что основным *узким местом* при увеличении плотности виртуальных машин является *объем памяти*; однако сейчас появилось множество серверов, с намного большим объемом памяти, что устраняет это узкое место. Облачные вычисления нивелируют это узкое место, устраняя проблему плотности машин – ее решение становится ответственностью *поставщика облака*, что освобождает пользователей облака от необходимости заботиться об этом.
- Для *облачных вычислений узким местом является пропускная способность канала к поставщику облака и обратно.*

Финансовый вопрос

- Поскольку существенную долю стоимости ИТ-операций составляют администрирование и управление, неявная автоматизация части этих функций в среде облачных вычислений сама по себе снизит расходы. Автоматизация может значительно снизить количество ошибок и уменьшить расходы на дублирование рутинных ручных операций.
- Есть и другие источники экономии финансовых затрат - стоимость обслуживания оборудования, электрической энергии, систем охлаждения и, естественно, администрирования и управления и т.д.

Снижение рисков

Рассмотрите следующие возможные риски:

- Неблагоприятные последствия неправильного управления данными.
- Неоправданные расходы на обслуживание.
- Финансовые или юридические проблемы поставщика.
- Эксплуатационные проблемы или простои поставщика.
- Проблемы восстановления данных и конфиденциальности.
- Общие проблемы безопасности.
- Атаки на систему извне.

При использовании облачных систем всегда существует проблема безопасности данных, их доступности и злонамеренных действий, затрудняющими вычислительные процессы.

Однако при тщательном продумывании плана, методологии выбора поставщика сервиса и при трезвом подходе к общему управлению рисками большинство компаний может благополучно использовать преимущества данной технологии.

Заключение

- В наше время облачные вычисления могут предоставить организациям средства и методы, необходимые для обеспечения финансовой стабильности и высокого уровня обслуживания.
- Но для достижения оптимальной защищенности облачных вычислений и выработки общих эксплуатационных стандартов необходима глобальная кооперация.



ВЕДЕНИЕ В ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ И РИСКИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

План

- сущность облачных вычислений,
- основные характеристики облачных вычислений,
- уровни облачных вычислений, от инфраструктурного до клиентского, и сервисы одного или нескольких уровней,
- некоторые риски облачных вычислений.

Введение

- Крупные компании, такие как IBM®, Microsoft® и Amazon, объявили свои стратегии в области облачных вычислений. Представили свои услуги и более мелкие компании, даже начинающие. В этой лекции дается общий обзор облачных вычислений и предоставляемых ими функциональных возможностей – с точки зрения как поставщиков, так и потребителей.
- Хотя облачные вычисления обычно именуют "программное обеспечение как сервис" (Software as a Service - SaaS), иногда их называют "все как сервис" (everything as a service - EaaS). Под "все" подразумеваются серверы, базы данных, программные продукты и любые их комбинации. Мы рассмотрим различные уровни (сервисы), предлагаемые поставщиками облачных вычислений. Мы также обсудим риски облачных сервисов.

Облачные вычисления

- И так, идея облачных вычислений состоит в том, что компании, использующие ИТ-услуги, могут покупать эти услуги как сервисы. Вместо того, чтобы покупать серверы для поддержки внутренних или внешних сервисов и приобретать лицензии на программное обеспечение, компания может купить их как сервис.
- Здесь нет ничего нового. Уже довольно давно существуют сервисы хостинга, позволяющие сконцентрироваться на бизнесе, а не на ИТ-инфраструктуре. Однако облачные вычисления могут предложить нечто большее.

Преимущества облачных вычислений

- Представьте, что крупный поставщик продает свои товары через Web-портал. Ему требуется доступ к серверам и инфраструктуре, чтобы поддерживать самые высокие **пиковые нагрузки** (дневные часы пик, предпраздничные часы пик), но большую часть времени можно справляться с нагрузками, используя меньше ресурсов. Возникает вопрос, нужна ли такая большая инфраструктура, которая не используется полностью большую часть времени?
- Используя услуги хостинга, крупный производитель покупает необходимую **инфраструктуру как сервис**, вносит ежемесячную или ежегодную плату и больше не беспокоится об инфраструктуре. Он покупает именно столько ресурсов, сколько нужно в часы пиковых нагрузок. *Используя облачные вычисления*, можно платить только за *фактически используемые ресурсы*. Пользователи получают необходимые сервисы и платят только за их реальное использование. Если посещаемость Web-сайта компании в рабочие дни велика, а в выходные дни почти отсутствует, оплачивается только необходимая мощность (и использование) в рабочие дни. Выгоды для потребителя совершенно очевидны.

Ключевые характеристики облачных вычислений:

- Пользователь не должен знать (и покупать) полный объем ресурсов, который может понадобиться во время пиковых нагрузок. Облачные вычисления предоставляют возможность масштабировать ресурсы, доступные приложению. Фирма, только начавшая бизнес, не должна беспокоиться о том, что ее рекламная кампания сработает слишком хорошо и серверы обвалятся от трафика.
- Потребители платят только за то, что используют. Им не нужно покупать серверы или ресурсы в объеме, соответствующем их максимальным потребностям. Часто это экономит средства.
- Облако автоматически (или в некоторых случаях в полуавтоматическом режиме) выделяет и освобождает по требованию ресурсы процессоров, дискового пространства и пропускной способности сети. При малом числе пользователей на сайте облако использует для поддержки его работы очень мало ресурсов, и наоборот.
- Поскольку центры обработки данных, где выполняются сервисы, имеют гигантские размеры и распределяют ресурсы среди больших групп пользователей, затраты на инфраструктуру (электроэнергия, здания и т. д.) уменьшаются. В результате затраты на отдельного пользователя оказываются ниже.

Уровни

- Облачные вычисления имеют несколько уровней. Поставщики этих уровней предлагают очень разные сервисы и рабочие режимы. Некоторые поставщики концентрируются на создании и обслуживании огромного центра обработки данных, другие – на создании дружественных и богатых функциональными возможностями приложений.
- Такими уровнями (снизу вверх) являются: *инфраструктура, системы хранения, платформа, приложение, сервисы и клиент.*

Инфраструктура

- На нижнем уровне находится инфраструктура услуги, или *виртуализация платформ*. Вы получаете серверную среду такого типа, какой захотите. Это базовое предложение; пользователям все равно приходится управлять сервером, заниматься установкой всего программного обеспечения и осуществлять поддержку самостоятельно.

Инфраструктура облачных вычислений отличается от сервисов традиционного хостинга *масштабируемостью* и оплатой только по *факту* использования. Компанию, только начинающую свою деятельность, может очень заинтересовать масштабируемость и отсутствие оплаты периода, когда сервис не используется. Это удобно, особенно когда вы пытаетесь поднять трафик своего Web-приложения, но не знаете, когда это произойдет и на сколько.

Системы хранения

- На уровне систем хранения предлагается база данных или что-то подобное, а оплата взимается за количество использованных гигабайт в месяц. Этот уровень не является чем-то новым или особенным, за исключением полного набора сервисов. Конечно же, это очень важный уровень.

Существует много вариантов хранения данных. Некоторые из них являются традиционными базами данных, другие - например, Bigtable от Google или SimpleDB от Amazon - представляют собой фирменные решения с закрытой архитектурой.

Платформа

- Уровень платформы – это, например, такие решения, как Ruby on Rails, LAMP (*Linux - Apache Web-Server - MySQL - PHP*) или Python. Это уже более интересный уровень.
- Воображаемой начинающей фирме не нужно иметь дело с установкой серверного программного обеспечения или заниматься обновлением его версий, поскольку это входит в предлагаемые услуги. Фирма может сконцентрироваться на разработке и маркетинге своего приложения.

Приложение

- Уровень приложений содержит приложения, предлагаемые в качестве сервисов. Наиболее известными примерами, наверное, являются Salesforce.com и Google Docs, хотя имеются сотни, если не тысячи реальных приложений, которые можно приобрести в качестве сервисов.

Облачными сервисами являются такие популярные Web-приложения, как Facebook, Flickr и LinkedIn. Пользователь, возможно, даже не знает, где работает приложение: в масштабируемом центре обработки данных, на обычном хостинге или в помещении поставщика сервисов. Но это не является заботой или проблемой пользователя, использующего приложение. Наверное, этот уровень является самой заметной частью облачных вычислений. Он наглядно демонстрирует преимущества, которые могут видеть пользователи.

Сервисы

- Уровень сервисов содержит взаимодействующие по сети операции компьютер-компьютер. Самым распространенным примером этого уровня являются Web-сервисы.
- К другим примерам относятся платежные системы, такие как PayPal, и картографические сервисы, такие как Google Maps и Yahoo Maps.

Клиент

- На вершине стека уровней находится клиентский уровень, к которому относятся пользователи облачных систем. Клиентами, к примеру, являются пользователи настольных систем (тонких или толстых клиентов) и мобильные пользователи (Symbian, Android, iPhone).
- Как видите, существует огромное количество возможностей как для поставщиков – в части *предложения существующих или новых сервисов*, так и для пользователей – в части поиска сервисов и приложений, решающих их проблемы. Однако имеются и определенные риски, которые должны понимать пользователи.

Риски

- Есть несколько вопросов, о которых стоит подумать до покупки сервисов, основанных на облачных вычислениях. Следует рассматривать указанные ниже риски не как преграды или кардинальные недостатки, а только как проблемы, которые необходимо учитывать при принятии решений. Необходимо проанализировать, удовлетворяют ли облачные вычисления вашим требованиям, и какие из доступных сервисов вам подходят.

Зависимость от поставщика

- Убедитесь в том, что легко сможете забрать ваши данные из сервиса. Если вы используете инфраструктурный сервис, резервное копирование файлов и данных должно осуществляться относительно просто.
- Если вы используете Web-приложение, подготовьте план получения своих данных при возникновении необходимости перехода к другому поставщику.
- Не всегда нужно переносить все данные в новое приложение, если имеется возможность как-то их просматривать. Например, не нужно переносить все старые данные приложения отслеживания поведения пользователей, если у вас имеется доступ для их просмотра.

Надежность

- Если с поставщиком сервиса что-то случается (например, останавливаются серверы), пользователь ничего не может с этим поделать. Для подобных ситуаций лучше всего выбирать поставщика сервисов, предлагающего зеркалирование. Хотя иногда даже этого недостаточно. Даже крупные поставщики не застрахованы от проблем.

Безопасность данных

- Безопасность данных - это не всегда риск. Процедуры защиты и опыт поставщика могут быть намного лучшими, чем у маленькой начинающей фирмы.

Проблема заключается в том, кто может просматривать данные и какова политика поставщика в этом плане. Например, если ваши данные не должны видеть конкуренты, проверьте политику поставщика.

Уход из бизнеса


- Проанализируйте, что может случиться с вашими данными или приложением, если ваш поставщик вынужден будет прекратить деятельность. Этот негативный аспект редко упоминается в маркетинговых материалах. Если экспорт ваших данных выполняется просто, то возможное прекращение деятельности поставщика не должно представлять опасности. Хотя вы все равно столкнетесь с задачей поиска нового приложения (или поставщика), подходящего под ваши бизнес-требования.

Заключение

- В данном разделе вы познакомились с сущностью облачных вычислений, основными характеристиками, уровнями и некоторыми рисками, которые нужно учитывать. *Для обычного пользователя облачные вычисления означают получение сервисов по запросу и их оплату по фактическому использованию.* Облачные вычисления имеют множество достоинств, таких как масштабируемость, распределение ресурсов по требованию, экономия на масштабе. Облачные вычисления имеют шесть уровней, от инфраструктурного до клиентского. Поставщики предлагают сервисы одного или нескольких уровней. Также вы узнали о рисках, которые должны учитывать пользователи и поставщики.
- Облачные вычисления – это не тенденция будущего. Это уже реальность; обычные Web-пользователи сталкиваются с облачными вычислениями ежедневно, даже не зная об этом. В ближайшем будущем мы, возможно, увидим более четкую тенденцию к переносу в облака корпоративных сервисов и приложений. Естественно, есть ряд приложений, которые организации предпочитают выполнять на собственных серверах, но большинство корпоративных приложений вполне могут подходить для облачных вычислений.

- 
- Смотреть: Введение в облачные вычисления

Инфраструктура как сервис (Infrastructure as a Service) платформа как сервис (Platform as a Service) и программное обеспечение как сервис (Software as a Service) являются примерами современных концепций, которые помогут сориентироваться в новых моделях.



МОДЕЛИ СЕРВИСОВ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Часть I. Инфраструктура как сервис

- Ознакомимся с ключевыми концепциями инфраструктуры как сервиса (Infrastructure as a Service – IaaS).

IaaS

- предоставляет базовые сервисы, такие как виртуальные серверы, хранилища данных и базы данных, на одной платформе для разработки и выполнения приложений.

Некоторые основные концепции IaaS:

- Выгрузка в облако (Cloudbursting).
- Вычисления с множественной арендой (Multi-tenant computing).
- Организация пулов ресурсов (Resource pooling)
- Гипервизор (Hypervisor).

Значение IaaS

- Главной ценностью IaaS с точки зрения бизнеса является концепция, называемая *cloudbursting* – процесс выгрузки задач в облако в период, когда необходимо максимальное количество вычислительных ресурсов. Потенциал экономии при этом очень велик, поскольку компании не нужно вкладывать средства в приобретение дополнительных серверов, загруженных на 70% мощности дважды или трижды в год, а в остальное время работающих с нагрузкой всего лишь 7-10%.
- Но чтобы использовать эти преимущества IaaS, ИТ-отделы должны уметь создавать и внедрять программное обеспечение, способное перераспределять процессы в облако IaaS.

Четыре фактора перераспределения процессов в IaaS

При создании и внедрении программного обеспечения, управляющего такими процессами перераспределения, нужно учитывать четыре важных фактора:

- Разработка приложений для закрытого IaaS-облака *конкретного* поставщика может стать дорогостоящей ошибкой, если этот поставщик прекратит свою деятельность.
- Сложность хорошо написанного программного обеспечения для перераспределения ресурсов высока и обычно требует наличия первоклассных разработчиков, которые стоят недешево.
- Что вы будете отправлять в облако для обработки? Отправка персональных данных, финансовой информации и других конфиденциальных данных может быть рискованной для организации с точки зрения соблюдения требований нормативных документов.
- Важно понимать опасность переноса в облако процессов, являющихся критически важными для повседневной работы компании. Рекомендуем создать таблицу, в одном столбце которой будут размещены процессы, затрагивающие данные, критичные в плане соответствия нормативным документам, во втором – задачи, критичные для ведения бизнеса, а в третьем – некритичные задачи. Перенос следует планировать только для тех элементов программного обеспечения, которые размещены в третьем столбце.

Форматы для переноса VM

- Кроме того, необходимо учитывать текущее состояние рынка облачных вычислений с точки зрения привязки к поставщику. Использование виртуальных машин (VM), которые можно переносить из центров обработки данных в облако и между облаками разных поставщиков, может быть ценным активом компании, но такая возможность требует поддержки поставщиками стандартизированного формата файлов, что они делают с большой неохотой.
- В настоящее время не существует открытой спецификации, разработанной каким-либо органом по стандартизации. Другими словами, **отсутствует стандартизированный формат**, что больше всего усложняет ситуацию, поскольку нет никаких гарантий какой-либо поддержки выбранного вами формата.
- Однако стоит отметить, что зачастую существует возможность перенести виртуальную систему в другой формат при условии открытости спецификации этого формата и наличия доступа к ней.
- Обнадеживает недавно достигнутый значительный прогресс в поддержке формата Open Virtualization Format (OVF), который является прекрасным кандидатом на то, чтобы стать стандартом. Еще одним таким кандидатом является формат Virtual Machine Disk (VMDK). VMDK первоначально был собственностью VMware, но в настоящее время он является **открытым и поддерживается многими поставщиками**.

Инфраструктура как актив

- Чтобы проиллюстрировать эволюцию облачных вычислений, рассмотрим развитие автомобильной промышленности за последние пять десятилетий. Конкурентным преимуществом для производителей автомобилей в 1960-1970 годы чаще всего было количество лошадиных сил и *величина* крутящего момента, которые можно было втиснуть в автомобиль. Однако в 1980 годы эта парадигма утратила актуальность, и на рынке произошел переход от парадигмы "инфраструктура как актив" к парадигме "инфраструктура как сервис".
- Точно так же подавляющее большинство успешных компаний за последние 50 лет потратили много драгоценного времени и ресурсов на построение инфраструктуры, нацеленной на получение дополнительных конкурентных преимуществ, создавая более крупные, быстрые и надежные сети, чем у конкурентов. Парадигма "инфраструктура как актив" в ИТ-области имеет ряд таких же негативных с точки зрения эффективности аспектов, как и мощные автомобили 60-х и 70-х годов.

Инфраструктура как актив

Что касается корпоративных вычислений, то среди таких аспектов следует отметить:

- Длительные периоды недоиспользования вычислительной мощности и производительности и расходы, связанные с выделением большого пространства для размещения аппаратного обеспечения в больших и дорогих центрах обработки данных.
- Потребность в дорогостоящих трудовых ресурсах, включая круглосуточное дежурство сетевых администраторов в центрах обработки данных, где размещается инфраструктура (серверы, маршрутизаторы, коммутаторы и т.д.).
- Высокий барьер для "зеленых вычислений" (Green Computing) как результат высокого уровня неэкономного потребления энергии.

IaaS

- IaaS означает переход от парадигмы (модели) "инфраструктура как актив" к парадигме "инфраструктура как сервис". Две другие категории облачных вычислений, также означают изменение парадигмы. PaaS (платформа как сервис) – это замена парадигмы "платформа как актив", требующей приобретения большого количества лицензий. То же можно сказать и о SaaS (программное обеспечение как сервис), которая является заменой парадигмы "программное обеспечение как актив" в форме лицензий на программное обеспечение, предоставляемых как сервис.
- Более подробная информация о PaaS и SaaS приведена во второй и третьей частях.
- Обратимся к матрице концепций, приведенной в таблице I. Начнем с IaaS.

Таблица I. Матрица концепций трех категорий облачных вычислений (часть I)

IaaS	
Заменяемая парадигма	Инфраструктура как актив
Характеристики	Обычно не зависит от платформы; расходы на инфраструктуру разделяются и, следовательно, снижаются; соглашения SLA; оплата по факту использования; автоматическое масштабирование.
Основные понятия	Распределенные вычисления (grid computing), вычисления как коммунальная услуга (utility computing), вычислительный экземпляр (compute instance), гипервизор (hypervisor), выгрузка в облако (cloudbursting), вычисления с множественной арендой (multitenant computing), организация пулов ресурсов (resource pooling).
Преимущества	Снижение капиталовложений в аппаратное обеспечение и трудовые ресурсы; снижение риска потери инвестиций; низкий порог внедрения; плавное автоматическое масштабирование.
Недостатки и риски	Бизнесэффективность и производительность очень зависят от возможностей поставщика; потенциально большие долгосрочные расходы; централизация требует новых/ других подходов к мерам безопасности.
Когда не стоит использовать	Когда капиталовложения превышают текущие расходы.

Основные аспекты IaaS

- Возможно, более правильно было бы представлять Интернет не как единое глобальное облако, а как систему, состоящую из множества облаков, похожую на грозу. Исходя из этой метафоры, логично утверждать, что молния – это погодный эквивалент обмена данными между облаками. Данная метафора, возможно, является более точной в том смысле, что облака систематически взаимодействуют между собой, создавая единый результат – Интернет.

Основные аспекты IaaS

- Маловероятно, что Интернет станет единым облаком (по крайней мере в ближайшем будущем), поскольку стандарты облачных вычислений отсутствуют, а поставщики предпринимают попытки долгосрочной капитализации путем привязки пользователей.
- Возможно, когда-нибудь Интернет действительно станет единым взаимосвязанным облаком, в котором виртуальные машины смогут без усилий переноситься в облако независимо от формата файлов, а взаимосвязанные кластеры виртуальных машин будут управляться поставщиками сервисов через единый интерфейс. Но этот день настанет не скоро. А пока мы будем рассматривать Интернет как систему, состоящую из множества облаков.

Эластичная инфраструктура

- Эластичность (гибкость) – это первый важнейший аспект IaaS.
- Чтобы продемонстрировать концепцию эластичности, представьте, что облака сделаны из зефира, слепленного вместе так, чтобы на них можно было сидеть и ездить. Каждое зефирное облако может удерживать определенное количество людей в зависимости от числа зефирных кластеров, составляющих облако, а также от числа отдельных зефирин, содержащихся в этих кластерах. По мере увеличения количества людей, едущих на облаке, мы можем расширять зефирные кластеры, добавляя зефирины и тем самым увеличивая площадь. Как вы уже догадались, люди представляют собой приложения, которым нужны вычислительные ресурсы (например, для размещения Web сайтов и выполнения программных сервисов). Зефирные кластеры представляют собой кластеры виртуальных машин, а каждая зефирина – это виртуальная машина.
- Такая метафора облегчает понимание концепции *эластичной кластеризации*.
- *Кластеризация физических серверов с целью формирования виртуального облака известна как облачная кластеризация (cloud clustering).*

Эластичная инфраструктура

- Давайте рассмотрим пример.
- Представьте, что вы выполняете статистические исследования по заказу правительства.
- Перед вами поставлена задача обработать все данные последней переписи населения. Вы должны представить необходимые статистические данные в течение трех дней, чтобы на их основании можно было принять важные решения по распределению взыскиваемых сборов и налогов. Излишне говорить, что это весьма значительная работа, и вы испытываете некоторую нехватку времени. Более того, вам нужно обработать огромный объем данных, и вы осознаете, что для его обработки понадобится трехнедельная работа всего ИТ-отдела.

Эластичная инфраструктура

- Это именно тот тип проблем, которые можно легко решить при помощи IaaS. Фактически, используя IaaS, можно выполнить анализ всей переписи населения за час. Можно начать с создания одного экземпляра сервера, содержащего систему управления базами данных для выполнения запросов к данным. Этот экземпляр называется образом (image).
- После развертывания образа и импорта данных в базу можно продублировать этот образ необходимое число раз и запустить задания обработки данных. В процессе выполнения заданий можно вручную или автоматически добавлять или удалять ресурсы. Например, если вычисления выполняются недостаточно быстро, можно просто добавить в кластер дополнительные дубликаты образа.
- Теперь, когда вы познакомились с концепцией эластичности, рассмотрим второй важный аспект IaaS – виртуализацию.

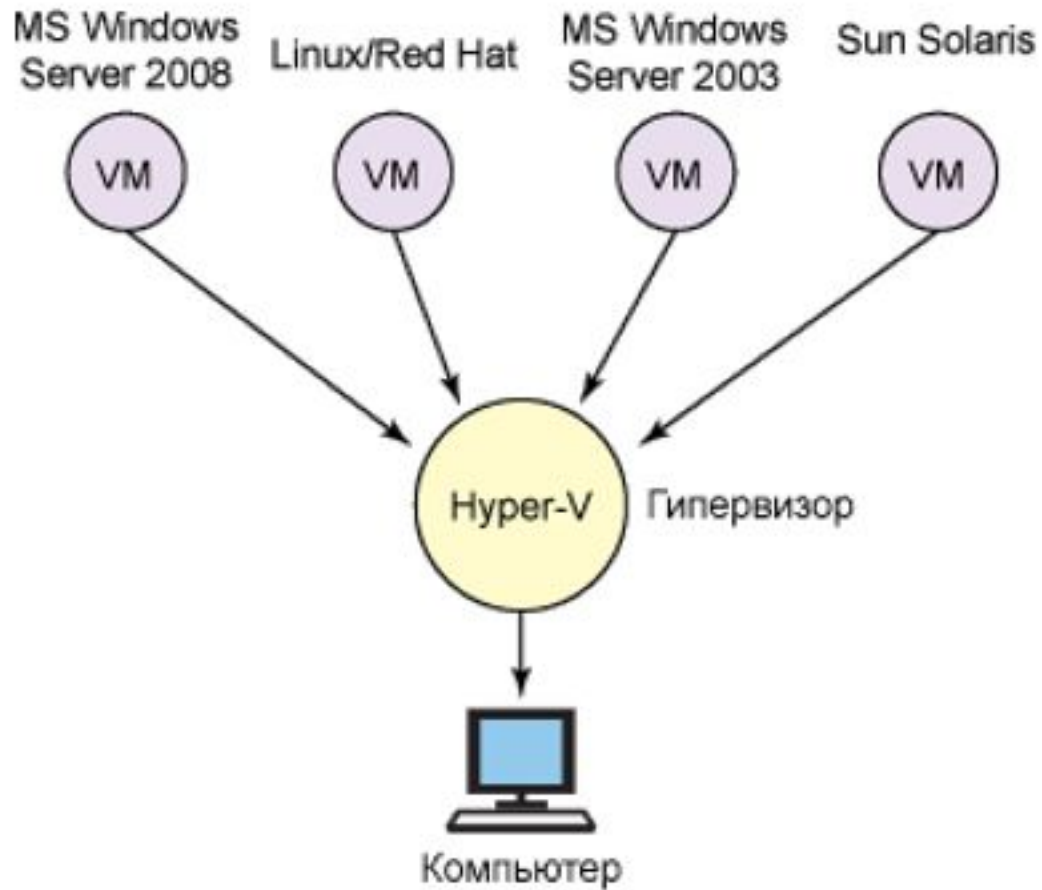
Виртуализация машин

- У Сергея Брина (Sergey Brin) и Ларри Пейджа (Larry Page), создателей Google, в 1995 году родилась правильная идея, когда они вечерами рылись в мусорных баках за факультетом компьютерных наук Стэнфордского университета, извлекая из них выброшенные компьютерные запчасти. Они приносили эти запчасти, основанные на архитектуре x86, в свою комнату студенческого общежития, и подключали к машине "Франкенштейн", на которой был размещен легендарный Web-паук, индексировавший всю сеть компьютеров Стэнфорда.
- В настоящее время, по некоторым оценкам, Google имеет более 1 миллиона x86-серверов в 12 крупных центрах обработки данных и примерно 20 центрах меньшего размера, расположенных на различных континентах. Это очень большое облако. Масштабирование существа из студенческой комнаты, рожденного в 1995 году, до миллиона серверов современной сети Google стало возможным благодаря двум ключевым факторам дизайна системы.
- Во-первых, Google по сей день продолжает использовать недорогие x86-компоненты вместо намного более дорогих компонентов корпоративных серверов, применяемых в центрах обработки данных многих компаний.
- Во-вторых, *восстановление после сбоев, резервирование, мониторинг, кластеризация и другие задачи управления инфраструктурой* обрабатываются *системой виртуализации, работающей под уровнем операционной системы*, а не отдельным специализированным оборудованием, таким как распределители нагрузок.

Виртуализация машин

- IaaS легко определить, поскольку она обычно не зависит от платформы. IaaS состоит из комбинации ресурсов аппаратного и программного обеспечения. Программное обеспечение IaaS представляет собой **низкоуровневый код**, выполняющийся независимо от операционной системы и называемый **гипервизором**, который отвечает за инвентаризацию аппаратных ресурсов и распределение их по требованию (см. рисунок 1). Этот процесс называется организацией *пулов ресурсов* (resource pooling).
- Организация пулов ресурсов гипервизором делает возможной *виртуализацию*, а *виртуализация* делает возможными *вычисления со множественной арендой* (multi-tenant) – совместное использование инфраструктуры несколькими организациями со сходными интересами в области требований к защите и соблюдения нормативов.

Рисунок 1. Взаимоотношения между виртуальными машинами, гипервизором и компьютером



IaaS

- Благодаря IaaS появляется возможность предоставлять *ресурсы* процессора, системы хранения данных, сетевые и другие вычислительные ресурсы, на которых можно размещать произвольное *программное обеспечение*, такое как операционные системы и приложения.
- Большинство вариантов использования облачных вычислений следует единой фундаментальной многоуровневой структуре, к которой мы уже привыкли: *стек программных решений или платформа развернуты на сетевой инфраструктуре, а приложения выполняются на верхнем уровне.*
- Однако виртуализация делает парадигму облака уникальной.

Заключение

- В этой части лекции приведена информация об основных принципах облачных вычислений, а также об анатомии IaaS и о том, как ее можно использовать в реальных ситуациях.
- В следующей части будет рассмотрена вторая основная категория облачных вычислений – PaaS.

Вы узнаете, какие возможности предлагает концепция "Платформа как сервис" (Platform as a Service – PaaS), для того чтобы разработчики получили выгоду от использования облачных вычислений, а владельцы и руководители предприятий смогли заранее оценить поставщиков PaaS, сведя к минимуму потенциальную возможность ошибок.



МОДЕЛИ СЕРВИСОВ

ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ:

ЧАСТЬ 2. ПЛАТФОРМА КАК

СЕРВИС

Концепция "Платформа как сервис"

- вызывает больше всего разночтений, поскольку ее трудно идентифицировать и отличить от концепций "Инфраструктура как сервис" и "Программное обеспечение как сервис". В этой второй части рассказывается о том, что делает PaaS уникальной и как ее можно эффективно использовать в бизнесе.
- Определяющим фактором уникальности PaaS является то, что она позволяет **разработчикам создавать и развертывать Web-приложения** на предлагаемой инфраструктуре. Другими словами, PaaS позволяет воспользоваться практически безграничными вычислительными ресурсами облачной инфраструктуры.
- Естественно, безграничные вычислительные ресурсы – это иллюзия, поскольку они ограничены размером инфраструктуры. Однако, как говорилось в первой части, инфраструктура Google, по оценкам, содержит более миллиона компьютеров архитектуры x86. Кроме того, поскольку инфраструктура для PaaS является эластичной, облако может расширяться, предоставляя, при необходимости, еще большие вычислительные ресурсы, так что представление о безграничности не так уж далеко от реальности.

Paas для разработчиков

- Общее заблуждение разработчиков состоит в том, что облачные вычисления касаются только сетевых администраторов. При этом игнорируются многие возможности, которые облачные вычисления предоставляют *разработчикам и отделам обеспечения качества.*
- Рассмотрим некоторые вопросы, которые часто приводят к проблемам в период разработки программного обеспечения. Большой проблемой может быть *процесс настройки серверной среды, в которой будет размещаться Web-приложение, создаваемое группой разработчиков.* Даже на самых крупных предприятиях обычно имеется только один сетевой администратор, обслуживающий все группы разработчиков.

Если не использовать PaaS,

настройка среды разработки или тестирования обычно требует выполнения следующих действий:

- Приобрести и развернуть сервер.
- Установить операционную систему, библиотеки среды разработки и выполнения программ, репозиторий управления исходным кодом и все остальное необходимое программное обеспечение промежуточного уровня.
- Настроить операционную систему, модули среды разработки и выполнения программ и дополнительное программное обеспечение промежуточного уровня.
- Перенести или скопировать существующий код.
- Протестировать и запустить код, чтобы убедиться в корректности работы всей системы.

Сетевой администратор

- Сетевой администратор скорее уже загружен работой до предела, поэтому развертывание новой среды может представлять собой болезненный процесс. Еще одной большой проблемой для разработчиков Web-приложений (как клиентской, так и серверной части) является дублирование среды исполнения на локальном рабочем месте для собственного тестирования.
- В ситуации, когда вы работаете в группе разработчиков, использующей PaaS, у вас будет виртуальная машина, содержащая полную серверную среду, которую буквально можно переносить на USB-флэшке.
- Обратимся к матрице концепций.

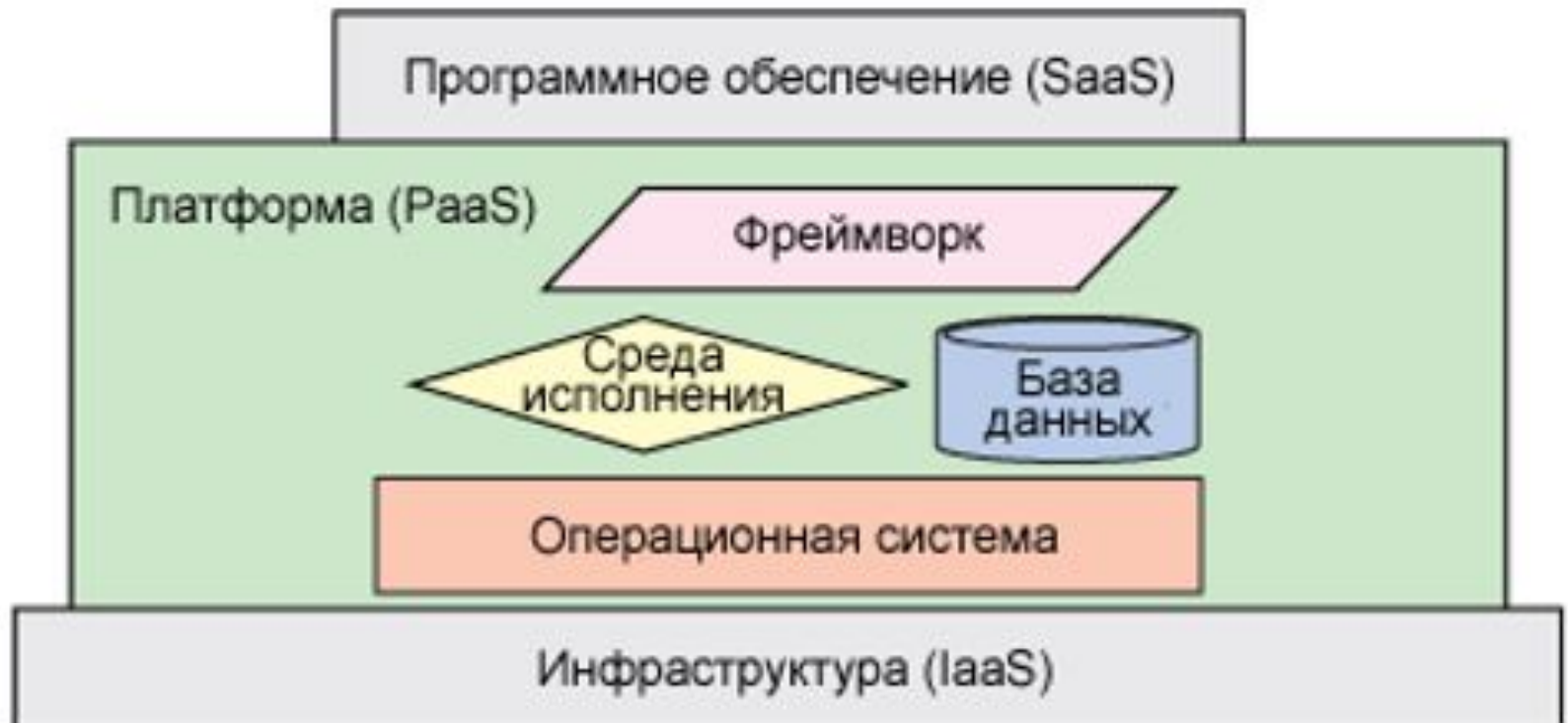
Таблица I (продолжение). Матрица концепций трех категорий облачных вычислений (часть 2)

Paas	
Заменяемая парадигма	Приобретение лицензий
Характеристики	Потребляет инфраструктуру облака; обеспечивает методы динамического (agile) управления проектами.
Основные понятия	Стек решений (solution stack).
Преимущества	Плавное развертывание версий.
Недостатки и риски	Централизация требует новых/ других мер безопасности.
Когда не стоит использовать	Отсутствует.

Основные компоненты PaaS

- Возможно, лучший способ разобраться в PaaS - это разбиение на главные компоненты: **платформу и сервис**. Будем рассматривать предоставляемый **сервис как стек решений**. Тогда двумя главными компонентами PaaS являются *вычислительная платформа и стек решений*.
- Для лучшего понимания этих двух компонентов рассмотрим их определения.
- *Вычислительная платформа* в своем простейшем виде представляет собой место, где может без проблем работать программное обеспечение, если оно отвечает стандартам этой платформы. Типичными примерами платформ являются: для операционных систем: Windows™, Apple Mac OS X и Linux®; для мобильных вычислений: Google Android, Windows Mobile® и Apple iOS; Adobe® AIR™ или для программных инфраструктур: Microsoft® .NET Framework. Важно помнить, что мы не говорим о самом программном обеспечении, а только о платформе, на которой оно должно работать. Рисунок 1 поможет понять эти взаимоотношения.

Рисунок 1. Графическая интерпретация взаимоотношений между категориями облачных вычислений и элементами PaaS



Стек решений

- Стек решений состоит из **приложений**, которые будут помогать в процессе **разработки и развертывания программ**.
- Такими приложениями являются операционная система, среда разработки и исполнения, репозиторий управления исходными кодами и любое другое необходимое программное обеспечение промежуточного уровня.

Выбор поставщика

Стек решений – это именно то, чем различаются между собой компании-поставщики PaaS. Его нужно очень тщательно исследовать, прежде чем принимать окончательное решение и прыгать на подножку поезда PaaS.

Вот несколько основных вопросов, которые, возможно, следует задать конкретному поставщику PaaS до подписания договора:

- Какие инфраструктуры и языки поддерживаются? В идеале услуга PaaS должна поддерживать *все инфраструктуры, основанные на предпочтительном для этой платформы языке.*
- Сколько приложений можно создать? Большинство поставщиков PaaS ограничивает количество создаваемых приложений на основании плана или пакета, на который вы подписались. Убедитесь в наличии плана или пакета, отвечающего вашим требованиям.

Выбор поставщика

- Какого рода содержимое разрешено? Инфраструктуры, поддерживающие PaaS, обычно оперируют концепцией под названием "вычисления со множественной арендой" (multitenant computing), когда много "арендаторов" "живут" на одном сервере, разделенные посредством экземпляров виртуальных машин, управляемых гипервизором. Поставщик PaaS может устанавливать определенные *ограничения на виды приложений и содержимого*, которые вы планируете разместить.
- Базы данных какого типа поддерживаются? Ответ на этот вопрос очень важен, если имеются данные, которые вы будете переносить как часть приложения. Необходимо убедиться в том, что предлагаемая поставщиком база данных совместима с форматом, который вы намереваетесь использовать для импорта своих данных.
- Поддерживается ли протокол SSL (HTTPS)? Это еще один важный фактор с точки зрения безопасности. Вы можете столкнуться с большими проблемами, если, запланировав ведение финансовых операций посредством ваших приложений, обнаружите, что протокол SSL не поддерживается.

Анатомия PaaS

Теперь давайте рассмотрим *функциональные возможности*, которые следует учитывать при сравнении поставщиков PaaS:

- **Инфраструктура разработки приложений.** Надежная инфраструктура разработки приложений, построенная на широко распространенной технологии. В идеале нужно остерегаться потенциальной возможности попасть в зависимость от поставщика. В этом отношении обычно безопасным является использование технологии с открытыми исходными кодами, такой как Java™.
- **Простота использования.** PaaS должна иметь простые в использовании WYSIWYG программы с предустановленными виджетами, готовые компоненты пользовательского интерфейса и инструменты перетаскивания объектов (drag-and-drop), а также поддерживать несколько стандартных интегрированных сред разработки. Все это будет способствовать быстрой итеративной разработке приложений.
- **Инструменты моделирования бизнес-процессов (Business Process Modeling - BPM).** Вам необходима надежная BPM-инфраструктура, позволяющая выполнять моделирование бизнес-процессов и создавать приложения на основе этой модели.
- **Доступность.** Выбранная платформа должна быть доступна всегда и везде.
- **Масштабируемость.** Платформа должна быть достаточно интеллектуальна, чтобы использовать эластичность инфраструктуры для обработки нагрузок, которые может создавать приложение.

Анатомия PaaS

- **Безопасность.** Платформа должна иметь эффективные средства борьбы с угрозами, такими как межсайтовые скрипты, внедрение SQL, отказ в обслуживании (Denial of Service), и средства шифрования трафика, а также предоставлять возможность встраивать эти средства в разрабатываемые приложения. Кроме того, платформа должна поддерживать возможности единого входа (*single sign-on*), чтобы ее можно было интегрировать в другие локальные или облачные приложения.
- **Инклюзивность.** Платформа должна обеспечивать возможность включать, встраивать и интегрировать другие приложения, созданные на этой или другой платформе.
- **Переносимость.** Платформа должна быть независимой от используемой инфраструктуры и позволять компаниям переносить приложения с одной IaaS на другую.
- **Инструменты переноса.** Чтобы упростить и ускорить процесс миграции данных из традиционных локальных приложений в приложения, основанные на новой платформе, в составе инструментария платформы должны быть программы пакетного преобразования и импорта данных.
- **API.** Для выполнения задач аутентификации пользователей, сохранения и извлечения файлов (например, файлы и активы Web-приложения), а иногда и прямых вызовов базы данных, платформа должна иметь хорошо документированный API. Это позволит повысить гибкость создания и настройки программного обеспечения на совместную работу с платформой в соответствии с конкретными требованиями компании.

Управление привязкой к поставщику

- Привязка к поставщику означает, что пользователь зависит от него и не может перейти к другому поставщику без существенных затрат. Возможность создать среду, поддерживающую привязку к поставщику, возникает в относительно новых, с растущей популярностью технологиях, таких как облачные вычисления. Последователи новых технологий должны все тщательно обдумать, прежде чем подписывать какие-либо долгосрочные соглашения с поставщиками IaaS и PaaS.
- Одним из способов избежать привязки к поставщику является стандартизация API и технологий платформы. Такие организации, как Simple Cloud, начали работать с различными поставщиками, вовлекая их в проект с открытыми исходными кодами, преследующий цель внедрить PaaS в облако. Инициатива Simple Cloud, объединившая компании Zend Technologies, Microsoft, IBM и Rackspace, имеет целью разработку уровня абстракции между различными платформами.

Управление привязкой к поставщику

- Simple Cloud API предназначен для создания *общих интерфейсов* для сервисов хранения файлов, сервисов хранения документов и сервисов простых очередей. Это позволило бы писать приложения, переносимые между основными поставщиками облачных вычислений.
- При выборе поставщика PaaS-сервисов рекомендуется обратить внимание на поставщиков, поддерживающих стандартизацию. Стандарты облегчают жизнь всем, кто работает в ИТ-отрасли, и, что самое важное, они экономят деньги.
- Чтобы освободить рынок PaaS от привязки к поставщику, необходимы поставщики сервисов, поддерживающие *общий базовый API*.

Заключение

- В этой второй части вы узнали, как можно охарактеризовать и идентифицировать PaaS. Вы также узнали, какие вопросы следует задавать и чего следует избегать при выборе поставщика PaaS (например, привязки к поставщику). В последней части данной лекции будет подробно рассмотрена анатомия SaaS, ее характеристики и идентификация. Вы также узнаете о том, на что необходимо обращать внимание при выборе поставщика SaaS.

Вы узнаете о четырех основных факторах, способствующих окупаемости инвестиций разработчиков, работающих в рамках концепции "Программное обеспечение как сервис", и о том, как использовать эти факторы для повышения рентабельности SaaS. В этой части рассматриваются возможности бизнес-модели, предоставляемые SaaS, а также раскрывается роль проектирования взаимодействия с пользователем для SaaS-приложений.



МОДЕЛИ СЕРВИСОВ

ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ:

ЧАСТЬ 3. ПРОГРАММНОЕ

ОБЕСПЕЧЕНИЕ КАК СЕРВИС

Концепция "программное обеспечение как сервис"

- Концепция "программное обеспечение как сервис" предусматривает предоставление доступа по сети к коммерческому программному обеспечению.
- Возможно, вы уже используете SaaS, даже не подозревая об этом. Примерами SaaS являются Netflix, Photoshop.com, Acrobat.com, Intuit QuickBooks Online, Gmail и Google Docs. Другой, возможно, менее очевидной, реализацией SaaS является значительная часть растущего рынка мобильных приложений.

SaaS

- SaaS предоставляет дешевый способ использования программного обеспечения – использование по требованию вместо покупки лицензии на каждый компьютер, особенно когда большинство компьютеров не используется почти 70% времени. В отличие от покупки нескольких лицензий для одного пользователя, чем ближе время использования лицензии компанией к 100%, тем больше денег компания экономит.
- Для удобства приводим матрицу концепций.

Таблица I (продолжение). Матрица концепций трех категорий облачных вычислений (часть 3)

SaaS	
Заменяемая парадигма	Программное обеспечение как актив (бизнеса и потребителя)
Характеристики	Соглашения SLA; пользовательский интерфейс, предоставляемый приложениями тонких клиентов; компоненты облака; взаимодействие посредством API; не сохраняющий состояние (stateless); слабосвязанный (loosely coupled); модульный; семантическая совместимость.
Основные понятия	Тонкий клиент; клиент-серверное приложение.
Преимущества	Снижение капиталовложений в аппаратное обеспечение и трудовые ресурсы; снижение риска потери инвестиций; плавное итеративное обновление.
Недостатки и риски	Централизация данных требует новых/других мер безопасности.
Когда не стоит использовать	Отсутствует

Четыре фактора повышения окупаемости инвестиций для SaaS

- SaaS открывает новые возможности поставщикам программного обеспечения. В частности, поставщики SaaS-приложений называют четыре основных фактора повышения окупаемости инвестиций (ROI):
 - Увеличение скорости развертывания.
 - Улучшение принятия пользователями.
 - Уменьшение требований к поддержке.
 - Снижение стоимости реализации и обновлений.

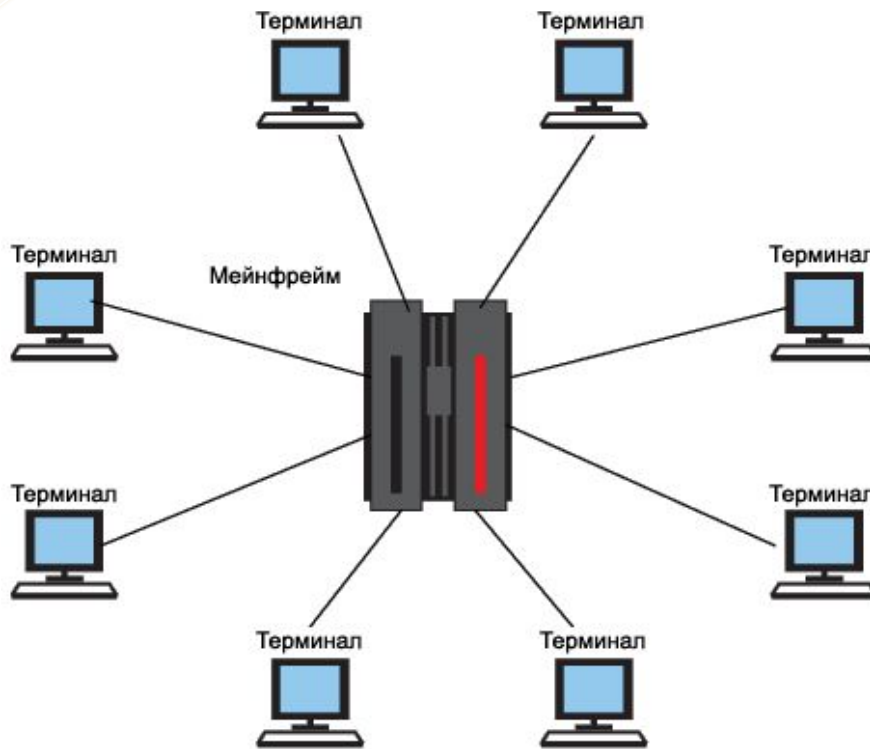
Скорость развертывания

- При развертывании традиционных настольных приложений традиционно возникают значительные трудности. Разработчики настольных приложений называли обновление своих приложений "кошмаром развертывания".
- "Трудности логистики при попытке заставить тысячи, если не сотни тысяч, клиентов перейти на строго определенную версию вашей программы в строго определенное время поистине безграничны" (Tariq Ahmed "*Flex 4 в действии*", Manning Press)
- Далее автор говорит, что эти трудности настолько огромны, что большинство компаний, разрабатывающих настольное программное обеспечение, вообще не считают такое обновление рациональным и даже осуществимым. Разработчики, столкнувшиеся с этим в прошлом, являются хорошими кандидатами для развертывания SaaS-версий своего программного обеспечения.

Скорость развертывания

- Однако самый большой барьер для выхода на рынок SaaS, который нужно преодолеть разработчикам традиционного программного обеспечения, – возможность выполнения настольных приложений как SaaS-приложений.
- Во многих случаях сделать это можно только путем переписывания программы на определенном уровне, что некоторые компании находят очень дорогостоящим.
- Это одна из основных причин того, что переход к облачным вычислениям – медленный и постепенный процесс. В большинстве случаев наиболее логичным решением является поэтапный перенос программного обеспечения в облако, начиная с предоставления на условиях SaaS *очень упрощенной версии оригинального приложения*. Это имеет очевидный смысл, учитывая ограниченность возможностей разработчика с точки зрения управления версиями.
- Можно заметить много общего между облачными вычислениями и вычислениями в локальной сети. Типичная архитектура локальной сети состояла из ряда рабочих станций, часто называемых *клиентскими терминалами*, выполняющих приложения путем подключения к мощному мейнфрейму, как показано на [рисунке 1](#).

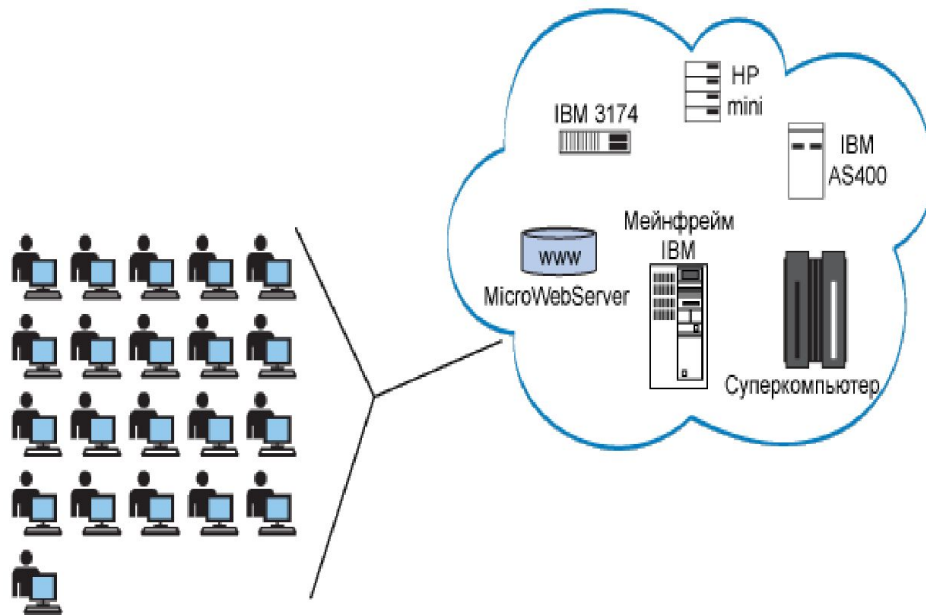
**Рисунок 1. Простая
схема
взаимоотношений
клиентских
терминалов и
мейнфрейма в
простой локальной
сети**



На [рисунке 1](#) показан один мейнфрейм, способный поддерживать достаточное количество экземпляров программного обеспечения для обслуживания всех клиентов, подключающихся в пределах локальной сети.

Скорость развертывания

- Такой тип архитектуры прекрасно работал на предприятиях, поскольку ИТ-отделы имели неограниченный уровень контроля над управлением версиями, а обновления можно было разворачивать итеративно и плавно. Точно так же и в облаке устраняются логистические барьеры, мешавшие разработчикам настольных приложений в плане управления версиями, поскольку программное обеспечение выполняется в инфраструктуре, к которой компания-разработчик имеет мгновенный доступ.
- Разумеется, размер и масштаб инфраструктуры SaaS по сравнению с локальной сетью огромны, учитывая количество клиентов, которых нужно обслужить. Но основополагающие концепции остаются теми же.



На [рисунке 2](#) показано облако, состоящее из разных вычислительных ресурсов, которые создают общую вычислительную мощность, способную выполнять большое количество экземпляров программного обеспечения, необходимого для обслуживания клиентов по всему миру.

**Рисунок 2. Простая
схема
взаимоотношений
клиентских
устройств и облака
в SaaS**

Улучшение освоения

ПОЛЬЗОВАТЕЛЯМИ

- SaaS делает доступным программное обеспечение, лицензии на использование которого прежде были слишком дороги для обычного потребителя. Хорошим примером являются инициативы Adobe по предоставлению в виде SaaS доступа к Adobe® Photoshop®.
- Все больше людей предпочитают использовать для базового редактирования фотографий сайт Photoshop.com, а не загружать полную версию, когда необходимо выполнить только элементарные задачи редактирования. Здесь важно то, что люди, которым не нужна вся функциональность полной версии, теперь будут экономить деньги. В то же время Photoshop.com используют те пользователи, которые иначе и не пытались бы это делать, т.е. у Adobe появляется возможность получить в долгосрочной перспективе клиентов, к которым ранее не было доступа.

Улучшение освоения

ПОЛЬЗОВАТЕЛЯМИ

- Наличие различных бизнес-моделей является особенно интересным аспектом SaaS. Например, Intuit предлагает сервисы QuickBooks Online в виде SaaS с ежемесячной оплатой. В свою очередь, Adobe использует возможности SaaS в Photoshop.com и Acrobat.com, предлагая программное обеспечение как *freemium*-сервис – термин, придуманный для описания специфичной бизнес-модели, основанной на предложении в виде SaaS упрощенной версии лицензируемого программного продукта.
- Freemium SaaS основана на модели получения дохода, которая предполагает, что определенный процент бесплатно обслуживаемых пользователей в конечном итоге посчитает полезным либо перейти на платную SaaS-версию, имеющую дополнительные функциональные возможности, либо приобрести лицензированную копию настольного приложения, содержащую все доступные дополнительные функции.
- Такой метод пробного использования программного обеспечения в демонстрационном режиме становится все более предпочтительным по сравнению с установкой приложения на настольный компьютер, которая может и не завершиться покупкой. Кроме того, такая модель может быть дополнена рекламой, если процент бесплатно обслуживаемых пользователей, переходящих на платные версии, меньше ожидаемого. Это общепринятый метод, который используют традиционные поставщики настольного программного обеспечения для адаптации к меняющемуся рынку по мере развития облачных вычислений.

Уменьшение требований к поддержке

- Стоимость больших систем технической поддержки пользователей и сложность проблем поддержки, возникающих в результате необходимости работы с несколькими платформами, значительно уменьшаются при использовании SaaS. *Простота развертывания* позволяет разработчикам начинающих компаний исправлять ошибки сразу после их обнаружения, что означает возможность исправления ошибок до того, как с ними столкнется подавляющее большинство пользователей. В результате уменьшается количество звонков в службу поддержки, а также значительно повышается вероятность удовлетворения и удержания клиентов.
- Кроме того, производители традиционного настольного программного обеспечения часто вынуждены поддерживать более одной платформы. Например, для разработчика, который должен поддерживать *операционные системы* Windows® и Linux, стоимость разработки практически удваивается, и это еще без учета проблем, которые возникают при поддержке *нескольких версий* операционной системы. Кроме того, поддержка нескольких версий операционной системы накладывает дополнительные ограничения.
- Например, если вы создаете программу для Windows 10, которая должна быть совместима с Windows 7, необходимо быть очень внимательным, чтобы все функциональные возможности могли работать на обеих версиях; в противном случае придется делать ветку исходного кода для каждой версии, что неминуемо снизит производительность и эффективность, а также увеличит время разработки. Если добавить сюда увеличение бюджета на поддержку различных операционных систем и различных их версий, то становится ясно, почему в настоящее время так высока доля неудачных проектов разработки программного обеспечения.

Снижение стоимости реализации и обновлений

- Источником существенной экономии средств разработчика является *отсутствие дополнительных расходов на тестирование и развертывание пакетов исправлений ошибок и новых функциональных возможностей на нескольких платформах*, поскольку разработчик может управлять платформой, на которой работает программное обеспечение. Это уменьшает стоимость обновления SaaS-приложений.
- Дополнительный косвенный финансовый результат связан с *удовлетворенностью и удержанием клиентов*, поскольку существенная экономия времени и средств позволяет разработчику лучше реагировать на запросы по созданию новых функций и повышению удобства эксплуатации.

SaaS и проектирование

взаимодействия с пользователем

- SaaS-приложения представляют *новый подход к дизайну приложений*. В SaaS-программах применяется современный подход к дизайну *пользовательского интерфейса*, больше согласующийся с процессом проектирования изделия в большинстве других отраслей. Такой подход включает в себя процесс, называемый *проектированием взаимодействия с пользователем* (User Experience Design – UXD), в котором графический пользовательский интерфейс разрабатывается отделом разработки продукта, а не отделом разработки ПО.
- *Главная задача UXD* – определить, что сделает приложение наиболее полезным для целевой группы пользователей, и включить это знание в дизайн. Хотя логически понятно, что это необходимо учитывать при разработке любого программного обеспечения, больше всего, по-видимому, этот принцип применяется к разработке SaaS-приложений. Причина, возможно, кроется в необходимости работы с бизнес-моделями SaaS, отличающимися от бизнес-моделей для традиционного программного обеспечения, а также в значительной экономии, получаемой от разработки на принципах SaaS.

SaaS с точки зрения разработчиков

- Итак, полнофункциональные облачные вычисления являются большим шагом вперед как для бизнеса, так и для потребителей, но предстоит преодолеть еще много препятствий. В результате процесс потребует времени и будет проходить через несколько этапов постепенных изменений. Во время этой эволюции вычислений **разработчикам** программного обеспечения очень важно уметь **адаптироваться** к меняющейся среде, чтобы соответствовать требованиям бизнеса и потребителей.
- Так же, как коммерческие компании должны быть способны развиваться вместе с меняющейся средой по мере развития облачных вычислений, программистам придется адаптировать свои знания и осваивать программные модели SaaS, чтобы оставаться востребованными. Облачные вычисления – это не только масштабируемые инфраструктуры и переносимость платформ посредством виртуализации. Они переводят программное обеспечение на совершенно новый уровень и открывают этап, который можно рассматривать как *следующее поколение в компьютерном программировании*.
- Например, экономичность этой модели означает более широкую доступность, что влечет за собой расширение клиентской базы. Добавьте сюда экономию средств, возникающую как прямое следствие наличия контроля над платформой, инфраструктурой и управлением версиями программного обеспечения, и вы быстро поймете, что SaaS привносит определенный уровень "демократизации", позволяющий малым и средним компаниям-разработчикам играть на равных с большими.

Заключение

- В данной разделе рассматривалась анатомия SaaS, третьей категории в нашей классификации облачных вычислений. Но главное - вы познакомились с возможностями и проблемами, которые предоставляет растущая популярность SaaS-приложений.
- Вы можете практически мгновенно регулировать производительность, создавать готовые корпоративные приложения в надежной, высокопроизводительной среде, в которой вы платите только за время и объем используемых ресурсов.



IBM и Web-сервисы Amazon

- IBM и AWS реализуют совместный проект по предоставлению доступа к программному обеспечению IBM в виртуальной вычислительной среде. Сервис Amazon EC2 позволяет оценивать и использовать программное обеспечение без его установки на собственной системе. Вы можете практически мгновенно регулировать производительность, создавая готовые корпоративные приложения в надежной, высокопроизводительной среде, в которой вы платите только за время и объем используемых ресурсов. В среде EC2 предлагается следующее программное обеспечение:
 - DB2 Express-C 9.5
 - Informix Dynamic Server Developer Edition 11.5
 - WebSphere Portal Server и Lotus Web Content Management Standard Edition
 - WebSphere sMash
- Предоставляются приложения производственного уровня со всеми функциональными возможностями и опциями. Ссылки на дополнительную информацию и файлы загрузки образов Amazon Machine Images для этих продуктов приведены в [developerWorks Cloud Computing Resource Center](#).
- Дополнительные ресурсы по облачным вычислениям можно найти в [разделе Cloud computing](#) на developerWorks.