Лекция 5:

**Веб-службы в Облаке**

Целью данной лекции является обзор веб-служб, предоставляемые концепцией облачных вычислений. Особое внимание уделяется типу "Инфраструктура как Сервис".

Поскольку технологии мигрируют от традиционной локальной модели в новую модель облака, сервисные предложения развиваются практически ежедневно. Предложения веб-служб часто имеют много общих характеристик. Часто от клиента требуются лишь минимальные затраты для получения услуги. Масштабируемость предполагается для каждого из типов предложений, но это не всегда необходимо. Многие из "облачных" вендоров еще работают над использованием масштабируемости, потому что их пользователи пока не нуждаются в данном виде услуг. Наконец, устройство и независимость местоположения позволяет пользователям получить доступ к системам независимо от того, где они находятся или какие устройства используют.

#### Инфраструктура как Сервис (IaaS)

*Инфраструктура как Сервис (Infrastructure-as-a-Service, IaaS)* - предоставление компьютерной инфраструктуры (как правило, это платформы виртуализации) как сервиса. IaaS существено усиливает технологию, услуги и вложения в центры обработки данных, чтобы предоставить это как услугу клиентам. В отличие от традиционного аутсорсинга, который требует должного усердия, бесконечных переговоров и сложных, длинных контрактов, IaaS сосредоточенна вокруг модели предоставления услуг, которая обеспечивает предопределенную, стандартизированную инфраструктуру, определенно оптимизированную под потребности клиента. Упрощенные предложения работы и выбор уровня сервисного обслуживания облегчает клиенту выбор решения с определенным набором основных эксплуатационных характеристик. Как правило, поставщики предоставляют компоненты следующих уровней:

* Аппаратное обеспечение (как правило, Грид с массивной горизонтальной масштабируемостью);
* Компьютерная сеть (включая маршрутизаторы, брандмауэры, балансировку нагрузки и т.д.);
* Подключение Интернет;
* Платформа виртуализации для того, чтобы запускать виртуальные машины;
* Соглашения сервисного обслуживания;
* Инструменты учета вычислений.

Вместо покупки пространства в центрах обработки данных, серверов, программного обеспечения, сетевого оборудования, и т.д., клиенты IaaS по существу арендуют ресурсы, которые находятся на стороне обслуживающих поставщиков услуг IaaS. Оплата за предоставление услуг обычно производится ежемесячно. Клиент платит только за потребленные ресурсы. Основные преимущества данного типа услуг включает:

* Свободный доступ к предварительно сконфигурированной окружающей среде;
* Использование инфраструктуры последнего поколения;
* Защищенные и изолированные вычислительные платформы;
* Уменьшение риска, за счет использования сторонних ресурсов, поддерживаемых третьими лицами;
* Способность управлять пиковыми нагрузками;
* Более низкие затраты;
* Меньшее время, стоимость и сложность при добавлении или расширении функциональности.

Вычисления по требованию приобретают все большую популярность среди предприятий. Вычислительные ресурсы, которые обслуживают пользовательские веб сайты, становятся все меньше и меньше, в то время, как доступные ресурсы поставщиков услуг постоянно возрастают. Модель по требованию развилась, чтобы преодолеть проблему того, как эффективно удовлетворить колеблющимся требованиям системы к ресурсам. Спрос на вычислительные ресурсы может существенно меняться за достаточно короткие промежутки времени, и поддержка ресурсов достаточных, чтобы удовлетворить пиковым требованиям может быть дорогостоящей. Технически переусложненное решение может быть столь же неблагоприятной, как ситуация, когда предприятие сокращает издержки, поддерживая только минимальные вычислительные ресурсы. Такие понятия как кластерные вычисления, Грид вычисления и т.д., могут казаться очень подобными понятию вычислений по требованию, но лучше их понять можно, если думать о них, как стандартных блоках, которые развивались в течение долгого времени, чтобы достигнуть современной модели облачных вычислений, которую мы используем сегодня.



**Рис. 4.1.** Грид вычисления

##### Amazon

Рассмотрим один из примеров – Amazon’s Elastic Compute Cloud (Amazon EC2). *Amazon EC2* – веб-служба, которая обеспечивает вычислительные мощности порядочного размера в облаке. Это разработано, чтобы сделать веб-вычисления доступнее для разработчиков и чтобы предложить множество преимуществ для клиентов:

* Интерфейс веб-службы, позволяет клиентам получать и формировать пространство с минимальным усилием;
* Предоставляет пользователям полный контроль над их (арендованными) вычислительными ресурсами и позволяют им работать в проверенной вычислительной окружающей среде;
* Уменьшает время, требуемое для получения и загрузки нового сервера до минут, разрешая клиентам быстро изменять конфигурацию, согласно их вычислительным требованиям;
* Изменяет экономику вычислений, позволяя клиентам платить только за используемые ресурсы;
* Предоставляет разработчикам инструменты, которые необходимы для построения отказоустойчивых приложений и изолирования себя от общих сценариев отказа.

*Amazon EC2* представляет вычислительную окружающую среду, разрешая клиентам использовать веб интерфейс, для получения и управления услугами, необходимыми для запуска одного или более экземпляров операционной системы. Клиенты могут загрузить окружающую среду с их настроенными приложениями. Они могут управлять сетевыми правами доступа и так много систем, сколько им нужно. Для использования *Amazon EC2,* клиентам сначала необходимо создать Amazon Machine Image (AMI). Этот образ содержит приложения, библиотеки, данные и связанные параметры конфигурации, используемые в виртуальной вычислительной среде. *Amazon EC2* предлагает использование предварительно сконфигурированные образы, созданные с шаблонами, необходимыми для немедленного запуска. Когда пользователи определили и сформировали их AMI, они используют инструменты *Amazon EC2* для загрузки образа в Amazon S3. *Amazon S3* – склад, который обеспечивает безопасный, надежный и быстрый доступ к клиенту AMI. Прежде, чем клиенты смогут использовать AMI, они должны использовать веб интерфейс *Amazon EC2* для настройки безопасности и сетевой доступ.

Во время конфигурации пользователи выбирают, какой тип категории и какую операционную систему они хотят использовать. Доступные типы категорий составляют две различные категории: Стандартный процессор или High-CPU процессор. Большинству приложений лучше всего удовлетворяет Стандартный случай, в который входят маленький, большой и очень большой экземпляры платформы. В случае High-CPU используется пропорционально больше ресурсов центрального процессора, чем памяти произвольного доступа (RAM), что более подходит высоконагруженным приложениям. В случае High-CPU процессора для выбора есть средняя и очень большая платформы. После определения, какую сущность использовать, клиенты могут запустить, завершить, и контролировать так много экземпляров их AMI, сколько необходимо при использовании интерфейсов прикладного программирования веб-службы (API) или большое разнообразие других инструментов управления, которыми производится обслуживание. Пользователи могут выбрать, хотят ли они запускать приложения в разных центрах обработки данных, использовать статические IP адреса конечных точек, при этом они платят только за фактически потребляемые ресурсы. Пользователи также могут выбрать доступные AMI из библиотеки. Например, если необходим обычный Linux сервер, то клиентами может быть один из стандартных Linux сборок AMIs.

Есть довольно много особенностей сервиса EC2, которые обеспечивают существенные льготы для предприятия. Прежде всего, *Amazon EC2* обеспечивает финансовую выгоду. Из-за крупного масштаба компании Amazon и большой клиентской базы, это недорогая альтернатива многим другим возможным решениям. Затраты понесенные на запуск и управление разделены между большим количеством клиентов, делая стоимость для любого клиента намного ниже, чем любая другая альтернатива. Клиенты платят очень низкий процент за вычислительные мощности, которые они фактически потребляют. Безопасность также обеспечена через интерфейсы веб-службы *Amazon EC2.* Эти интерфейсы позволяют пользователям формировать параметры настройки брандмауэра, которые контролируют сетевой доступ к и между группами экземпляров служб. *Amazon EC2* предлагает очень надежную среду, где случаи замены могут быть быстро обеспечены.

* Динамическая Масштабируемость.

*Amazon EC2* позволяет пользователям увеличивать или уменьшать производительность за несколько минут. Пользователи могут запускать единственный экземпляр, сотни экземпляров или даже тысячи экземпляров служб одновременно. Всем этим управляют с помощью API веб-службы, приложение может автоматически масштабировать себя вверх или вниз, в зависимости от его потребностей. Данный тип динамической масштабируемости очень привлекателен для клиентов предприятий, потому что это позволяет им соответствовать требованиям своих клиентов, не имея необходимость достраивать их инфраструктуру.

* Полный контроль над экземплярами.

У пользователей есть полный контроль над их экземплярами. У них есть полный доступ к каждому экземпляру, и они могут взаимодействовать с ними с любой машины. Экземпляры могут быть перезагружены, удаленно используя API веб-службы. Пользователи также имеют доступ к консоли своих экземпляров. Как только пользователи настроили их аккаунт и загрузили их AMI в сервис *Amazon S3,* им необходимо только запустить экземпляр. Возможно запустить AMI на любом числе экземпляров (или для любого типа), вызвав RunInstances API, который поддерживается Amazon.

* Гибкость конфигурации.

Параметры настройки конфигурации могут значительно различаться среди пользователей. У них есть выбор из разных типов экземпляров, операционных систем, и пакетов программного обеспечения. *Amazon EC2* позволяет им выбирать конфигурацию памяти, центрального процессора, и системы хранения, которая оптимально подходит для их выбора операционной системы и приложения. Например, выбор пользователя операционной системы может также включать многочисленные сборки Linux, Microsoft Windows Server, и даже OpenSolaris, все запущенные на действительных серверах.

* Интеграция с Другими Веб-службами Amazon.

*Amazon EC2* работает в соединении со множеством других веб-служб Amazon. Например, *Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB, Amazon Simple Queue Service (Amazon SQS)* и *Amazon CloudFront* все интегрированы, чтобы обеспечить полное решение для вычислений, обработки запросов и хранение между широким диапазоном приложений.

*Amazon S3* обеспечивает интерфейс веб-служб, который позволяет пользователям хранить и восстанавливать любой объем данных через Интернет в любое время, где угодно. Это предоставляет разработчикам прямой доступ к тому же самому, хорошо масштабируемому, надежному, быстрому, недорогому использованию инфраструктуры хранения данных Amazon, чтобы управлять их собственной глобальной сетью веб сайтов. Служба S3 стремится максимизировать преимущества масштабируемости и передать эти преимущества разработчикам.

*Amazon SimpleDB* - другой веб сервис, разработанный для того, чтобы выполнять запросы на структурированных данных *Amazon Simple Storage Service (Amazon S3)* в режиме реального времени. Этот сервис работает в соединении с *Amazon EC2,* чтобы предоставить пользователям возможность хранения, обработки и запросов наборов данных в пределах окружающей среды облака. Эти сервисы разработаны, чтобы сделать веб масштабируемые вычисления легче и более прибыльными для разработчиков. Традиционно данный тип функциональности был обеспечен использованием кластерной реляционной базы данных, которая требует значительных инвестиций. Внедрение данных технологий вызывало больше сложности и часто требовало услуг администрирования и поддержки базы данных.

В сравнении с традиционным подходом, *Amazon SimpleDB* легка в использовании и обеспечивает основную функциональность баз данных (например, поиск в реальном времени и запросы структурированных данных), не наследуя сложности эксплуатации, возникающие при традиционном выполнении. *Amazon SimpleDB* не требует схемы, данные индексируются автоматически, обеспечивает простой интерфейс API для хранения и доступа к данным. Это избавляет клиентов от необходимости выполнить задачи, такие как моделирование данных, обслуживание индексов и настройка производительности.

*Amazon Simple Queue Service (Amazon SQS)* – сервис принимает очереди сообщений для хранения. При использовании Amazon SQS, разработчики могут просто переместить данные, распределённые между компонентами своих приложений, которые выполняют различные задачи, не теряя при этом сообщения. При этом достигается высокая масштабируемость и надёжность. Amazon SQS работает как демонстрация масштабируемой передачи сообщений Amazon, инфраструктуры как сервиса. Любой компьютер, связанный с Интернетом, может добавить или прочитать сообщения без необходимости в установке какого-либо программного обеспечения или специальный конфигурации брандмауэра. Компоненты приложений, использующие Amazon SQS, могут запускаться независимо и не должны обязательно размещаться в той же самой сети, использующей те же самые технологии или работающие в то же самое время.

*Amazon CloudFront* – веб-сервис для доставки контента (содержания). *Amazon CloudFront* интегрируется с другими Amazon Web Services. Цель сервиса — дать разработчикам и предприятиям простой способ распространять контент для конечных пользователей с низкой задержкой, высокой скоростью передачи данных, при этом не требуя никаких обязательств. Запросы объектов автоматически маршрутизируются на самый близкий граничный сервер. Таким образом, содержание доставлено с лучшей возможной производительностью. Граничный сервер получает запрос от компьютера пользователя и соединяется с другим компьютером, вызывая оригинальный сервер, где расположено приложение. Когда оригинальный сервер выполняет запрос, он отправляет данные приложения назад на граничный сервер, который передает данные обратно компьютеру клиента, который выполнял запрос. Сервис не является свободным для пользования.

#### Платформа как Сервис (PaaS)

Развитие "облачных" вычислений привело к появлению платформ, которые позволяют создавать и запускать веб-приложения. *Платформа как сервис (Platform as a Service, PaaS)* — это предоставление интегрированной платформы для разработки, тестирования, развертывания и поддержки веб-приложений как услуги, организованная на основе концепции облачных вычислений.

Модель PaaS создает все условия требуемые для поддержки полного жизненного цикла создания и доставки веб-приложений и услуг доступных из сети Интернет, не требующих загрузки или установки программного обеспечения для разработчиков, ИТ менеджеров или конечных пользователей. В отличие от модели IaaS, где разработчики могут создавать определенные экземпляры операционных систем с доморощенными приложениями, разработчики PaaS заинтересованы только веб разработкой и не заботятся о том, какая операционная система используется. PaaS сервисы позволяют пользователям сосредотачиваться на инновациях, а не на сложной инфраструктуре. Организации могут направить существенную часть их бюджета на создание приложений, которые обеспечивают реальную ценность, вместо затрат на поддержку инфраструктуры. Модель PaaS таким образом открывает новую эру массовых инноваций. Теперь разработчики во всем мире могут получить доступ к неограниченной вычислительной мощности. Любой человек, имеющий доступ в Интернет, может создавать приложения и легко разворачивать.

Традиционный подход создания и запуска локальных (On-Premises) приложений всегда был сложен, дорог и рискован. Строительство Вашего собственного решения никогда не предоставляло гарантии успеха. Каждое приложение было разработано, чтобы удовлетворить определенным деловым требованиям. Каждое решение потребовало определенной конфигурации аппаратных средств, операционной системы, базы данных, электронную почту, веб-серверы, и т.д. Когда была создана окружающая среда аппаратного и программного обеспечения, команда разработчиков должна была выбрать комплекс платформ для разработки, чтобы создавать приложения. Неизбежно бизнес требует от разработчиков производить изменения в приложении. Измененное приложение требует новых циклов испытательных работ, прежде чем быть распространенным. Крупные компании часто нуждаются в специализированных средствах, чтобы разместить их в центрах обработки данных. Огромное количество электричества необходимо для работы серверов и поддержки системы кондиционирования. Наконец, все это требует использования отказоустойчивых площадок для центров обработки данных так, чтобы информация могла копироваться в случае сбоя.

PaaS предлагает более быструю, более экономически выгодную модель для разработки и доставки приложений. PaaS обеспечивает всю инфраструктуру для запуска приложений через Интернет. Аналогичные сервисы предоставляют большое количество компаний, таких как Microsoft, Amazon.com, Google. PaaS основан на модели учета лицензий или модели подписки, таким образом, пользователи платят только за то, что они используют. Предложения PaaS включают рабочие процессы для создания приложений, разработки приложений, тестирования, развертывания и размещения. Также сервисы приложений, виртуальные офисы, командное сотрудничество, интеграцию баз данных, безопасность, масштабируемость, хранение, работоспособность, управление состоянием, инструментарий приборных панелей и много другое.

Главные особенности PaaS включают сервисы для разработки, тестирования, развертывания, размещения и управления приложениями для поддержки жизненного цикла разработки приложений. Веб интерфейсы инструментов создания, как правило, обеспечивают некоторый уровень поддержки чтобы упростить создание пользовательских интерфейсов, основанных на таких технологиях как HTML, JavaScript и других технологиях. Поддержка многопользовательской архитектуры помогает избежать проблем при разработке относительно использования приложений многими пользователями одновременно. Провайдеры PaaS часто включают услуги для управления параллельной обработкой, масштабируемостью, отказоустойчивостью и безопасностью. Другая особенность – это интеграция с веб-службами и базами данных. Поддержка протокола обмена структурированными сообщениями в распределённой вычислительной среде (Simple Object Access Protocol, SOAP) и других интерфейсов позволяют приложениям PaaS создавать комбинации веб-сервисов (которые называют mashup) так же легко, как наличие доступа к базам данных и повторному использованию услуг внутри частных сетей. Способность формировать и распространять код между специализированными, предопределенными или распределенными командами очень увеличивают производительность предложений вендоров PaaS. Интегрированные предложения PaaS обеспечивают возможность для разработчиков, чтобы наиболее хорошо понимать внутреннюю работу их приложений и поведение пользователей при использовании инструментов, подобных приборной панели, чтобы рассмотреть внутренние параметры, основанные на измерениях количества параллельных соединений и т.д. Некоторые предложения PaaS расширяют этот инструментарий, что позволяет составлять счета оплаты за использование.

##### Microsoft Azure

*Платформа корпорации Майкрософт Windows Azure* (первоначально известная под названием Azure Services Platform) — это группа "облачных" технологий, каждая из которых предоставляет определенный набор служб для разработчиков приложений. На [рисунке 4.2](http://www.intuit.ru/studies/courses/673/529/lecture/11918?page=2#image.4.2) показано, что платформа Windows Azure может быть использована как приложениями, выполняющимися в "облаке", так приложениями, работающими на локальных компьютерах

 **Рис. 4.3.** Windows Azure предоставляет "облачные" приложениям службы для вычисления и хранения на базе Windows

Как показано на рисунке, Windows Azure выполняется на большом количестве компьютеров, расположенных в центрах обработки данных корпорации Майкрософт, и доступна через Интернет. Общая структура подключения Fabric Windows Azure соединяет множество вычислительных мощностей в единое целое. Службы Windows Azure для вычисления и хранения построены на основе этой структуры.

Вычислительная служба Windows Azure, естественно, работает на базе Windows. Для обеспечения первоначальной доступности этой службы осенью 2008 г. была открыта для широкой публики CTP-версия. Корпорация Майкрософт разрешила выполнять на Windows Azure только приложения, разработанные на платформе .NET Framework. Сегодня, однако, Windows Azure также поддерживает неуправляемый код, позволяя разработчикам выполнять приложения, которые разработаны не на базе .NET Framework. В любом случае такие приложения написаны на обычных языках Windows — C#, Visual Basic, C++ и других — с помощью Visual Studio 2008 или других средств разработки. Разработчики могут создавать веб-приложения с помощью таких технологий, как ASP.NET и Windows Communication Foundation (WCF), приложения, которые выполняются как независимые фоновые процессы, или приложения, сочетающие и то и другое.

Как приложения Windows Azure, так и локальные приложения могут получать доступ к службе хранилища Windows Azure, делая это одним и тем же способом: с помощью подхода RESTful. Однако Microsoft SQL Server не является базовым хранилищем данных. Фактически хранилище Windows Azure не относится к реляционным системам, и язык его запросов не SQL. Поскольку оно изначально предназначено для поддержки приложений на базе Windows Azure, то обеспечивает более простые и масштабируемые способы хранения. Следовательно, оно позволяет хранить большие двоичные объекты (binary large object — blob), обеспечивает создание очередей для взаимодействия между компонентами приложений и даже что-то вроде таблиц с простым языком запросов. (Для тех приложений Windows Azure, которым требуется обычное реляционное хранилище, платформа Windows Azure предоставляет базу данных SQL Azure, описанную далее.)

Выполнение приложений и хранение их данных в Интернете имеет очевидные преимущества. Например, вместо того, чтобы покупать, устанавливать и эксплуатировать собственные компьютеры, организация может доверить все это поставщику услуг Интернета. При этом заказчики платят только за вычислительные мощности и хранилище, которое они используют, и не связаны с обслуживанием большого количества серверов, предназначенных только для пиковых нагрузок. Если приложения правильно написаны, их можно легко масштабировать, воспользовавшись преимуществами огромных центров обработки данных, которые могут предложить поставщики.

И все же для получения этих преимуществ требуется эффективное управление. В Windows Azure каждое приложение имеет файл конфигурации, как показано на рис. 2. Изменяя информацию в этом файле вручную или с помощью программы, владелец приложения может контролировать различные аспекты его поведения, такие как настройка количества экземпляров, которые должны выполняться на платформе Windows Azure. Структура Fabric платформы Windows Azure наблюдает за тем, чтобы приложение поддерживалось в требуемом состоянии.

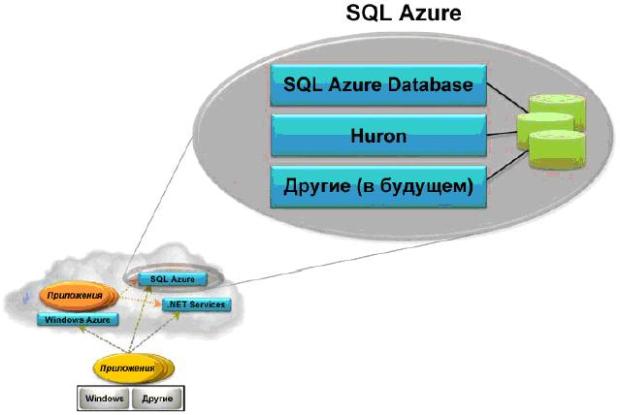
Чтобы позволить своим заказчикам создавать, настраивать приложения и наблюдать за ними, Windows Azure предоставляет портал, доступный с помощью браузера. Заказчик предоставляет Windows Live ID, а затем решает, создавать ему учетную запись размещения для выполнения приложений, учетную запись хранения для хранения данных или и ту и другую. Оплата использования приложения заказчиками может производиться любым удобным способом: с помощью подписки, повременно или как-нибудь иначе.

Windows Azure — это общая платформа, которую можно использовать в различных сценариях. Приведем несколько примеров, все они описываются с учетом возможностей CTP-версии.

* При создании нового веб-сайта, скажем, такого как Facebook, можно разрабатывать приложения на платформе Windows Azure. Благодаря тому, что данная платформа поддерживает как веб-службы, так и фоновые процессы, приложение может предоставлять интерактивный интерфейс пользователя и асинхронно выполнять работу для пользователей. Вместо того, чтобы тратить время и деньги, думая об инфраструктуре, пусковая группа может полностью сосредоточить свое внимание на разработке кода, который будет приносить пользу пользователям и инвесторам. Компания может также запустить небольшой веб-сайт, требующий незначительных затрат, если у ее приложения очень мало пользователей. Если приложение приобретает популярность и количество пользователей растет, Windows Azure при необходимости позволяет масштабировать его.
* Независимые поставщики программного обеспечения, создающие версию программы как службы имеющегося локального приложения Windows, могут разработать его на базе Windows Azure. Благодаря тому, что Windows Azure главным образом обеспечивает стандартную среду Windows, перемещение бизнес-логики приложения на эту "облачную" платформу не должно создавать особых проблем. Еще раз подчеркнем: разработка на базе имеющейся платформы позволяет независимым поставщикам ПО сосредоточить внимание на бизнес-логике, то есть на том, что позволяет им делать деньги, а не тратить время на инфраструктуру.
* Компания, создающая приложение для своих заказчиков, может выбрать для его разработки платформу Windows Azure. В силу того что Windows Azure поддерживает .NET, не представляет труда найти разработчиков с соответствующими навыками к тому же за разумную плату. Выполнение приложения в центрах обработки данных Microsoft освобождает предприятия от ответственности и расходов на поддержку собственных серверов, превращая капитальные затраты в эксплуатационные расходы. В особенности если у приложения имеются периоды пиковой нагрузки (если это, например, сетевой магазин цветов, которые необходимо вручить во время всеобщего ажиотажа 8 марта), предоставление корпорации Microsoft функции поддержки большой серверной базы, необходимой для этого, может оказаться экономически выгодным.

Выполнение приложений в "облаке" — один из самых важных аспектов "облачных" вычислений. С помощью Windows Azure корпорация Microsoft обеспечивает как платформу для выполнения приложений, так и способ хранения данных. По мере того, как растет интерес к "облачным" вычислениям, ожидается создание еще большего количества приложений Windows для этой новой области.

Один из наиболее привлекательных способов использования серверов, доступных через Интернет, — это обработка данных. Цель SQL Azure — решить эту проблему, предлагая набор веб-служб для хранения самой разной информации и работы с ней. В то время, как представители Microsoft заявляют, что постепенно SQL Azure будет содержать целый ряд возможностей, ориентированных на данные, включая создание отчетов, анализ данных и многое другое, первыми компонентами SQL Azure станут база данных SQL Azure Database и средство синхронизации данных Huron. Это наглядно продемонстрировано на [рисунке 4.42](http://www.intuit.ru/studies/courses/673/529/lecture/11918?page=2#image.4.4).



**Рис. 4.4.** SQL Azure обеспечивает средства в Интернете, ориентированные на работу с данными

База данных SQL Azure Database (ранее известная под названием SQL Data Services) обеспечивает систему управления базами данных (СУБД) в Интернете. Эта технология позволяет локальным и веб-приложениям хранить реляционные и другие типы данных на серверах Microsoft в центрах обработки данных Microsoft. Так же как при работе с другими веб-технологиями, компания платит только за то, что использует, увеличивая и уменьшая объем использования (и затраты) по мере возникновения необходимости в изменениях. Использование базы данных в "облаке" также меняет характер капитальных затрат: на место инвестиций в жесткие диски и ПО для СУБД приходят эксплуатационные затраты.

В отличие от службы хранилища Windows Azure база данных SQL Azure разработана на основе Microsoft SQL Server. Тем не менее в первоначальной CTP-версии 2008 г. база данных SQL Azure Database не предоставляла традиционный реляционный подход к данным. Учитывая отзывы заказчиков, корпорация Microsoft решила внести соответствующие изменения. В дальнейшем база данных SQL Azure Database будет поддерживать реляционные данные, обеспечивая среду SQL Server в "облаке" с индексами, представлениями, хранимыми процедурами, триггерами и многим другим. Доступ к этим данным можно получить с помощью ADO.NET и других интерфейсов доступа к данным Windows. Фактически приложения, которые сегодня получают доступ к SQL Server локально, будут работать почти точно так же с данными в SQL Azure Database. Для работы с этой информацией в "облаке" заказчики могут также использовать локальное ПО, такое как службы отчетов SQL Server.

В то время, как приложения могут использовать базу данных SQL Azure Database в значительной степени также, как локальную СУБД, требования к управлению существенно сокращены. Вместо того, чтобы беспокоиться о технике, например, обеспечивать мониторинг использования диска и обслуживание файлов журнала, заказчик SQL Azure Database может сосредоточить внимание на том, что действительно важно, на данных. Корпорация Microsoft будет отвечать за вопросы эксплуатации. Кроме того, так же как в случае с другими компонентами платформы Windows Azure, использование SQL Azure Database не составляет труда. Нужно просто зайти на веб-портал и предоставить необходимую информацию.

Второй компонент SQL Azure был заявлен под названием Huron Data Sync. Эта технология, разработанная на основе Microsoft Sync Framework и SQL Azure Database, позволяет синхронизировать реляционные данные в разных локальных СУБД. Владельцы данных могут определять, что именно должно синхронизироваться, как должны разрешаться конфликты и многое другое.

Приложения могут использовать SQL Azure самыми разными способами. Приведем несколько примеров.

* Приложение Windows Azure может хранить данные в SQL Azure Database. Несмотря на то, что Windows Azure обеспечивает собственное хранилище, реляционные таблицы не входят в число предлагаемых вариантов. Учитывая то, что многие имеющиеся приложения используют реляционное хранилище, а многие разработчики знают, как с ним работать, значительное количество приложений Windows Azure, скорее всего, будет работать с данными привычным способом, то есть с опорой на SQL Azure Database. Для повышения производительности заказчики могут указать, что определенное приложение Windows Azure должно выполняться в том же центре обработки данных, где SQL Azure Database хранит информацию этого приложения.
* В небольшой компании или подразделении большой организации приложение может использовать SQL Azure Database. Вместо того, чтобы хранить данные в базе данных SQL Server или Access, работающей на компьютере под чьим-то столом, приложение может использовать преимущества надежного и отказоустойчивого "облачного" хранилища.
* Предположим, производитель хочет сделать информацию о продукте доступной для своей дилерской сети и непосредственно для заказчиков. Размещение данных в SQL Azure Database позволит сделать их доступными для приложений, выполняемых на стороне дилеров, и для ориентированных на заказчиков веб-приложений, которые выполняются на стороне производителя.
* Компания с клиентской базой данных, реплицированной в географически удаленных местах, должна использовать компонент Huron для синхронизации этих реплик. Возможно, в каждой из географических точек требуется собственная копия данных для повышения производительности, обеспечения доступности или по каким-то иным причинам. Автоматическая синхронизация может сделать такое обязательное распределение менее проблематичным.

Идет ли речь о приложении Windows Azure, обеспечении большей доступности данных, синхронизации этих данных или о чем-то еще, службы данных в Интернете могут оказаться очень полезными. По мере появления новых технологий в рамках SQL Azure организации будут получать возможность использования Интернета для выполнения все большего количества задач, ориентированных на работу с данными.

Выполнение приложений и хранение данных в Интернете относятся к важным аспектам вычислительной сетевой среды. Однако они далеко не исчерпывают ее возможности. Другая возможность заключается в обеспечении инфраструктуры служб на базе "облака", которые могут использоваться локальными приложениями или веб-приложениями. Заполнить этот пробел и призваны службы .NET Services.

Первоначально известные как BizTalk Services, службы .NET Services предлагают функции для решения общих проблем инфраструктуры при создании распределенных приложений. На [рисунке 4.5](http://www.intuit.ru/studies/courses/673/529/lecture/11918?page=2#image.4.5) показаны их основные компоненты.

 **Рис. 4.5.** Службы .NET Services обеспечивают инфраструктуру в "облаке", которая может быть использована для веб-приложений и локальных приложений

Службы .NET Services состоят из следующих компонентов.

* Управление доступом. Получающий все большее распространение подход к удостоверениям заключается в том, что каждый пользователь должен предоставить приложению маркер, содержащий некоторый набор утверждений. На основании этих утверждений приложение решает, что разрешено делать пользователю. Эффективное осуществление этой процедуры в масштабах компании требует федерации удостоверений, которая позволяет принимать утверждения, сделанные в одной области удостоверений, в другой области. Может также потребоваться преобразование утверждений, изменяющее их при передаче из одной области удостоверений в другую. Служба управления доступом обеспечивает реализацию обеих функций на основе "облака".
* Шина служб. Предоставление служб приложений в Интернете гораздо труднее, чем может показаться. Задача шины служб — упростить эту процедуру, позволяя приложениям предоставлять конечные точки веб-служб, доступ к которым может быть получен другими приложениями — локальными или работающими в "облаке". Каждой предоставленной конечной точке присваивается URI, который клиенты могут использовать для поиска службы и получения доступа к ней. Шина служб также решает проблему преобразования сетевых адресов и прохождения через межсетевые экраны без открытия новых портов для предоставленных приложений.

Приведем несколько примеров использования службы .NET Services.

* Независимый поставщик ПО, который поставляет приложение, необходимое заказчикам из разных организаций, может использовать службу управления доступом для упрощения разработки и эксплуатации приложения. Например, этот компонент .NET Services может преобразовывать различные утверждения, применяемые в разных организациях с различными технологиями идентификации в согласованный набор, подходящий для приложения независимого поставщика ПО. Такое преобразование позволяет также разгрузить механизм федерации удостоверений за счет службы управления доступом, освобождая независимых поставщиков ПО от необходимости выполнения собственной локальной программы федерации.
* Предположим, предприятие хочет открыть доступ к одному из своих приложений торговым партнерам. Оно может распределить функции приложения с помощью веб-служб SOAP или RESTful и зарегистрировать их конечные точки с помощью шины служб. Затем торговые партнеры могут использовать шину для поиска конечных точек и доступа к службам. Это позволяет снизить риски, связанные с предоставлением приложения, поскольку не требует открытия новых портов в межсетевом экране компании. Организация может также использовать службу управления доступом, предназначенную для работы с шиной служб, для рационализации сведений о проверке подлинности, отправленной приложению ее партнерами.

Так же как в случае Windows Azure, предоставляется портал, доступный с помощью браузера, чтобы дать заказчикам возможность использовать службы .NET Services с помощью Windows Live ID. Цель корпорации Microsoft, достигаемая с помощью .NET Services, совершенно очевидна: обеспечить полезную "облачную" инфраструктуру для распределенных приложений.