МИНИСТЕРСТВО ОБРАЗОВАНИЯ АЗЕРБАЙДЖАНСКОЙ РЕСПУБЛИКИ

АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

«ЦЕНТР МАГИСТРАТУРЫ»

На правах рукописи

ГАСАНОВ ЭДГАР ХИКМЕТ

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

НА ТЕМУ:

«ИССЛЕДОВАНИЕ МЕТОДОВ И МОДЕЛЕЙ ОБЛАЧНЫХ СИСТЕМ ОБРАБОТКИ ДАННЫХ»

Наименование и шифр специальности:	060509 «Компьютерные науки»
Наименование специализации:	«Информационные технологии управления»

Научный руководитель:	**доц. Мамедова А.**
Руководитель магистерской программы:	**акад. Аббасов А.М.**
Заведующий кафедрой:	**акад. Аббасов А.М.**

БАКУ-2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность темы. Идея использования облачных вычислений в качестве утилиты заключается в том, чтобы привлечь новые организации к внедрению этой среды, чтобы справиться с энергично меняющейся бизнес-средой. ИТ-менеджеры видят облачные вычисления в качестве источника для поддержки масштабируемых ИТ-инфраструктур, обеспечивающих гибкость бизнеса.

Облачные вычисления начинались как средство для межличностных вычислений, но теперь они широко используются для доступа к программному обеспечению онлайн, онлайн-хранилищам, не беспокоясь о стоимости инфраструктуры и вычислительной мощности . Организации могут разгрузить свою ИТ-инфраструктуру в облаке и получить выгоду от быстрой масштабируемости. Важно понимать риски и угрозы в облачной среде, чтобы можно было подготовить эффективную политику безопасности для целей защиты. Для внедрения облачных вычислений важно, чтобы организации имели приемлемый уровень доверия к ним.

Необходимость решения некоторых проблем безопасности, связанных с облаком и виртуализацией, а также восприятия людей для анализа, необходим уровень осведомленности. Была проведена большая исследовательская работа, которая охватывает техническую сторону этих технологий, но предстоит проделать большую работу над восприятием людьми облачных вычислений и проблемами безопасности.

Предмет и объект исследования. В работе были исследованы облачные технологии, без которых современный мир IT невозможно представить, их классификация, модели и предоставляемые им услуги.

Основная цель и задачи исследования – состоит в обосновании понятия облачных вычислений , изучении проблем безопасности облак, их внедрение в организации.

Теоретико-методологической основа. В качестве методологических основ исследования были выбраны основные понятия облачных систем обработки данных, модели услуг облачных вычислений, принципы взаимодействия облачных провайдеров и клиентов с облачными системами.

Научная новизна. Исследование крупнейших современных облачных поставщиков услуг на примере Microsoft Azure.

Практическая значимость. Она заключается в том, чтобы побудить организации перейти к облачной инфраструктуре, поскольку эта технология будет иметь огромное значение и революционизировать модели предоставления услуг в мире ИТ. Организации, избегающие этого, будут отставать, но в то же время организациям следует рассмотреть возможность адаптации облачной стратегии, наиболее надежной и совместимой с их требованиями. Эта работа дает представление о различных технологиях и инструментах, реализованных специально для мониторинга и обеспечения безопасности в облаке. Большое внимание уделяется технологии виртуализации, потому что облачные вычисления сильно зависят от нее.

Структура и объём работы. Диссертация состоит из введения, трех глав, заключения, списка использованной литературы и реферата.

В первой главе были описаны основные аспекты облачных вычислений, его сравнение с Greed-вычислениями и результаты внедрения облачных систем в предприятия.

Во второй главе обсуждается основные модели услуг облачных систем обработки данных. Тут детально анализируются три основные модели облачных вычислений.

Третья глава посвящена обзору основных провайдеров облачных вычислений. Особое внимание уделено Microsoft Azure – одному из крупнейших поставщиков облачных услуг.

ГЛАВА I. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ОБЛАЧНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

1.1 Основные понятия облачных систем обработки данных

Информационные технологии всегда считались одной из основных проблем корпоративных организаций с точки зрения затрат и управления. Однако в отрасли информационных технологий за последнее десятилетие произошел существенный сдвиг - такие факторы, как коммерциализация аппаратного обеспечения, программное обеспечение с открытым исходным кодом, виртуализация, глобализация рабочей силы и гибкие ИТ-процессы, способствовали разработке новых технологий и бизнес-моделей.

Облачные вычисления в настоящее время предоставляют организациям больше возможностей для управления инфраструктурой, экономии затрат и передачи обязательств сторонним поставщикам. Он стал неотъемлемой частью технологий и бизнес-моделей и заставил предприятия адаптироваться к новым технологическим стратегиям [4].

Особености облачных вычислений:

• Облачные вычисления - это новая компьютерная парадигма.

• Ресурсы инфраструктуры (аппаратное обеспечение, хранилище и системное программное обеспечение) и приложения предоставляются в виде X-as-a-Service. Когда эти услуги предлагаются независимым поставщиком или

внешним клиентам, облачные вычисления основываются на бизнес-моделях с оплатой за использование.

• Основными функциями облаков являются виртуализация и динамическая масштабируемость по требованию.

• Utility computing и SaaS предоставляются интегрированным образом, даже если служебные вычисления могут использоваться отдельно.

• Облачные сервисы используются либо через веб-браузер, либо через определенный API.

NIST (Национальный институт стандартов и технологий) является признанным учреждением во всем мире для их работы в области информационных технологий. NIST определяет архитектуру облачных вычислений, описывая пять основных характеристик, три модели облачных сервисов и четыре модели облачного развертывания [5].

Есть 5 основных характеристик облачных вычислений, которые объясняют их взаимосвязь и отличие от традиционных вычислений.

1. Самообслуживание по требованию.

Потребитель может предоставлять или отменять предоставление услуг, когда это необходимо, без участия человека с поставщиком услуг.

2. Широкий доступ к сети.

Он имеет возможности по сети и доступен через стандартный механизм.

3. Объединение ресурсов.

Вычислительные ресурсы провайдера объединяются для обслуживания нескольких потребителей, которые используют мультитенантную модель с динамическим назначением различных физических и виртуальных ресурсов в зависимости от потребительского спроса [5].

4. Быстрая эластичность.

Услуги могут быть быстро и эластично предоставлены.

5. Измеренная служба.

Системы облачных вычислений автоматически контролируют и оптимизируют использование ресурсов, предоставляя возможность измерения для типа услуг (например, хранение, обработка, пропускная способность или активные учетные записи пользователей).

Существует 3 модели облачных сервисов, и эти 3 фундаментальные классификации часто называют «моделью SPI», то есть программным обеспечением, платформой или инфраструктурой как услугой [6].

1. SaaS – программное обеспечение как сервис.

Это возможность, с помощью которой потребитель может использовать приложения провайдера, работающие в облаке.

2. PaaS – платформа как сервис.

В этом типе услуг потребитель может развернуть, облачные или созданные пользователем приложения, созданные с использованием языков программирования или инструментов, предоставленных провайдером, в облачной инфраструктуре.

3. IaaS – инфраструктура как сервис

Это возможность, предоставляемая потребителю, с помощью которой он может предоставлять обработку, хранение, сети и другие основные вычислительные ресурсы, где потребители могут развертывать и запускать программное обеспечение [12].

Модели облачного развертывания

• Публичное облако - Облачная инфраструктура доступна для широкой публики.

• Частное облако - Тип облака, который доступен только для одной организации.

• Облако сообщества - В этом типе модели облачного развертывания инфраструктура облака совместно используется несколькими организациями и поддерживает определенное сообщество с общими проблемами.

• Гибридное облако - Это облачная инфраструктура, состоящая из двух или более облаков, то есть частных, общественных или общедоступных [14].

1.2. Сравнение облачных и Greed – вычислений.

Всегда были споры об эволюции облачных вычислений, и самым важным моментом в этом является Grid вычисления. Некоторые люди называют Облачные и Grid вычисления одним и тем же явлением, в то время как другие называют облачные вычисления расширением Grid-вычислений.

Grid вычисления - сложное явление, которое развилось благодаря более ранним параллельным, распределенным и высокопроизводительным вычислениям. Одно из наиболее цитируемых определений Grid-вычислений в начале было от Фостера и Кессельмана: «Grid вычисления - это аппаратная и программная инфраструктура, которая обеспечивает надежный, согласованный, всеобъемлющий и недорогой доступ к высокопроизводительным вычислительным возможностям [17]. Настоящая и конкретная проблема, которая лежит в основе концепции Grid, заключается в скоординированном распределении ресурсов и решении проблем в динамичных многоинституциональных виртуальных организациях. Обмен, который нас интересует, - это в первую очередь не обмен файлами, а прямой доступ к компьютерам, программному обеспечению, данным и другим ресурсам, как того требует ряд совместных стратегий решения проблем и посредничества в ресурсах, появляющихся в промышленности, науке и технике.».

Виртуальные организации в этом определении могут быть определены как динамическая группа людей, групп или организаций, которые определяют условия и правила для совместного использования ресурсов. Некоторые организации также определили Grid-вычисления в отношении их функций.

По данным IBM: «Grid-вычисления позволяют объединять пулы серверов, систем хранения и сетей в одну большую систему, чтобы вы могли передавать ресурсы нескольких систем одной точке пользователя для определенной цели. Для пользователя, файла данных или приложения система представляется единой огромной виртуальной вычислительной системой».

Описание облачных вычислений ранее и грид-вычислений здесь показывает, что облачные вычисления и грид-вычисления имеют много общего. Это приводит к обсуждению различий в этих двух технологиях. В приведенной ниже таблице показаны технические различия между облачными и Grid вычислениями [19].

	Grid вычисления	**Облачные вычисления**
Средства утилизации	Выделение нескольких серверов на одну задачу или работу	Виртуализация серверов; один сервер для вычисления нескольких задач одновременно
Типичная модель использования	Как правило, используются для выполнения задания, т.е. выполнения программы в течение ограниченного времени	Чаще используется для поддержки долгосрочных сервисов
Уровень абстракции	Обеспечивает высокий уровень детализации	Обеспечивает высокий уровень абстракции

Как показано в таблице, чем облачные вычисления отличаются от Grid-вычислений, так это «виртуализацией». Облачные вычисления используют

виртуализацию для максимизации вычислительной мощности. Виртуализация, отделяя логическое от физического, решает некоторые проблемы, с которыми сталкиваются Grid-вычисления. В то время как Grid-вычисления достигают высокой степени использования за счет распределения нескольких серверов для одной задачи или задания, виртуализация серверов в облачных вычислениях обеспечивает высокую степень использования, позволяя одному серверу одновременно выполнять несколько задач [22].

Наряду с различиями в технологиях между Grid-вычислениями и облачных вычислений, модели использования также различаются между ними. Сетка обычно используется для выполнения заданий, в то время как облака чаще используются для поддержки долгосрочных служб.

Как упоминалось выше, в мире технологий ведутся дебаты о том, что облачные вычисления развились из Grid-вычислений и что Grid-вычисления является основой для облачных вычислений. Таким образом, мы можем подвести итог, что Grid-вычисления является отправной точкой и основой для облачных вычислений. Облачные вычисления, по сути, представляют собой растущую тенденцию к внешнему развертыванию ИТ-ресурсов, таких как вычислительная мощность, хранилище или бизнес-приложения, и их получение в качестве услуг [11].

1.3. Внедрение облачных систем в организации.

В последнее время облачные вычисления стали одной из самых обсуждаемых технологий и привлекли большое внимание СМИ, а также аналитиков из-за возможностей, которые они предлагают. Компания IDC, занимающаяся исследованиями и анализом рынка, предполагает, что рынок услуг облачных вычислений в 2008 году составлял 16 миллиардов долларов, а к 2019 году вырос до 157 миллиардов долларов в год. Было подсчитано, что стоимостные преимущества облачных вычислений должны быть в три-пять раз больше для бизнес-приложений и более чем в пять раз для потребительских приложений [3].

Организации стремятся сократить вычислительные затраты, и по этой причине большинство из них начинают укреплять свои ИТ-операции, а затем используют технологии виртуализации. Для блага организаций существует новая технология, которая помогает им в этом, то есть облачные вычисления. Облачные вычисления претендуют на то, чтобы вывести организации на новый уровень и позволяют им еще больше сократить расходы за счет улучшения использования, снижения затрат на администрирование и инфраструктуру и ускорения циклов развертывания.

Облачные вычисления - это термин, используемый для описания как платформы, так и типа приложения. В качестве платформы она предоставляет,

настраивает и перенастраивает серверы, в то время как серверами могут быть физические или виртуальные машины [8]. С другой стороны, облачные вычисления описывают приложения, к которым предоставлен доступа через Интернет, и для этого используются большие центры обработки данных и мощные серверы для размещения веб-приложений и веб-сервисов.

Облако - это метафора для Интернета и абстракция для сложной инфраструктуры, которую оно скрывает. В определении есть несколько важных моментов, касающихся облачных вычислений. Облачные вычисления отличаются от традиционных вычислительных парадигм тем, что они масштабируемы, их можно инкапсулировать как абстрактную сущность, которая предоставляет клиентам различный уровень услуг, обусловленный эффектом масштаба, с динамически настраиваемыми службами.

Различные исследователи заявляют о преимуществах Cloud Computed, которые делают его более предпочтительным для организаций. Инфраструктура облачных вычислений позволяет организациям более эффективно использовать свои инвестиции в оборудование и программное обеспечение. Это достигается путем разрушения физического барьера, присущего изолированным системам, автоматизируя управление группой систем как единым целым. Облачные вычисления также можно описать как полностью виртуализированную систему и естественную эволюцию для центров обработки данных, которые предлагают автоматизированное управление системами [6].

Организации должны учитывать преимущества, недостатки и последствия облачных вычислений для своих организаций и практики использования, чтобы принять решение о принятии и использовании. В организации «принятие облачных вычислений зависит как от зрелости организационных и культурных процессов, так и от технологий [2].

Многие компании инвестировали в технологии облачных вычислений, создавая свои общедоступные облака, в том числе Amazon, Google и Microsoft. Эти компании часто выпускают новые функции и обновления своих услуг. Например, Amazon Web Services (AWS) разместили на своем веб-сайте центр Security2 и Economics3, чтобы получить академические и общественные рекомендации по этим вопросам. Это показывает, что по-прежнему существует множество сомнений относительно затрат и безопасности для организаций, использующих облачные вычисления. Следовательно, должны быть исследованы вопросы экономики и безопасности в облачных вычислениях для организаций [15].

Поскольку крупные организации по своей природе являются сложными, для облачных вычислений очень важно обеспечить реальную ценность, а не просто быть платформой для простых задач, таких как тестирование приложений или демонстрация продуктов. По этой причине следует изучить проблемы, связанные с переносом систем приложений в облако и удовлетворением потребностей ключевых заинтересованных сторон. Заинтересованные стороны включают технических, проектных, операционных и финансовых менеджеров, а также инженеров, которые собираются разрабатывать и поддерживать отдельные системы. Для организаций важен экономический фактор или фактор затрат, но в то же время отношения с клиентами, имидж общественности, гибкость, непрерывность бизнеса и соответствие имеют одинаковое значение [12]. Следовательно, организации должны понимать, как облачные вычисления влияют на все это.

Облачные вычисления - это также способ обеспечения и использования ИТ, а не только технологические усовершенствования центров обработки данных. Организации должны учитывать преимущества, недостатки и другие эффекты облачных вычислений для своих предприятий и практики их использования до принятия и использования облачных вычислений. В

организациях внедрение облачных вычислений во многом зависит от зрелости организационных и культурных процессов как технологии как таковой [9].

Решение о принятии облачных вычислений является сложной задачей из-за ряда практических и социально политическим причинам. Невозможно, чтобы все организации передали свои требования к вычислительным ресурсам для облачных провайдеров; они создадут гетерогенную вычислительную среду, основанную на выделенных серверах, организационных облаках и, возможно, более чем одном поставщике общедоступного облака. То, как осуществляется переход на облачные вычисления, зависит не только от технических проблем, но и от социально-технических факторов (т. е. стоимости, конфиденциальности и контроля), влияния на методы работы и ограничений, вытекающих из существующих бизнес-моделей. Следовательно, перед внедрением облачных вычислений организации должны решить следующие задачи: 1) предоставить точную информацию о затратах на внедрение облачных вычислений; 2) для поддержки управления рисками; и 3) обеспечить, чтобы лица, принимающие решения, могли сделать обоснованный компромисс между выгодами и рисками [15].

ГЛАВА II. АНАЛИЗ СУЩЕСТВУЮЩИХ МОДЕЛЕЙ УСЛУГ ОБЛАЧНЫХ СИСТЕМ ОБРАБОТКИ ДАННЫХ.

Облачные вычисления начали крупнейшую трансформацию ИТ в истории, и эта трансформация открыла много новых бизнес-возможностей. Ожидается, что публичные облака предоставят большинство возможностей поставщикам облачных услуг.

Такое быстрое внедрение общедоступного облака можно объяснить несколькими факторами:

• Не требуется никаких предварительных инвестиций или обязательств.

• Вы платите только за то, что используете.

• Компании могут протестировать любые услуги до покупки.

• Для обслуживания инфраструктуры требуется меньше человеческих ресурсов.

• Предложения по обслуживанию легко сравнивать.

• Обновления программного обеспечения могут быть автоматизированы.

Ожидается, что миграция в публичное облако ускорится. В то время как другие модели развертывания - такие как управляемый хостинг, гибридный хостинг и частные облака будут расти, публичные облака, несомненно, станут областью наиболее существенного роста. Компании, которые имеют больше возможностей для предоставления общедоступных облачных услуг, скорее всего, выиграют больше всего; поэтому большинство новых облачных сервисов предназначены для публичных облаков [14].

Здесь мы опишем и проанализируем таксономию облачного сервиса. Несколько компаний выборочно проверяются в каждом таксономическом отделении, и мы уделяем особое внимание таксономиям с общедоступными облаками, потому что именно там большинство предложений облачных услуг. Целью этого облачного таксономического анализа является классификация рыночных предложений и сравнение различных моделей ценообразования для нескольких поставщиков.

Поскольку все предложения облачных услуг могут быть классифицированы в одну или несколько моделей облачных услуг, мы используем три фундаментальные модели (SaaS, PaaS и Iaas) в качестве верхних уровней таксономии. Некоторые могут утверждать, что можно идентифицировать много дополнительных моделей, таких как «база данных как услуга» или «безопасность как услуга», и такие аргументы действительны, поскольку официальной классификации облачных сервисов не существует. Тем не менее, все текущие модели могут быть с пользой классифицированы по трем исходным [2].

2.1 Инфраструктура как услуга (Iaas)

Инфраструктура как услуга помогает компаниям переместить свою физическую инфраструктуру в облако с уровнем контроля, аналогичным уровню, который они имели бы в традиционном локальном центре обработки данных. IaaS обеспечивает самое близкое сходство с внутренним центром обработки данных по сравнению с другими типами услуг.

Основными компонентами инфраструктуры центра обработки данных являются хранилище, серверы (вычислительные устройства), сама сеть и инструменты управления для обслуживания и мониторинга инфраструктуры. Каждый из этих компонентов создал отдельную рыночную нишу [6].

В то время как некоторые небольшие компании специализируются только на одной из этих облачных ниш IaaS, у крупных облачных провайдеров, таких как Amazon или RightScale, есть предложения во всех областях IaaS. Технически, рынок IaaS имеет относительно низкий входной барьер, но для создания и поддержки облачной инфраструктуры могут потребоваться значительные финансовые вложения. Зрелые фреймворки для управления облаком с открытым исходным кодом, такие как OpenStack, доступны для всех и обеспечивают надежную программную основу для компаний, которые хотят создать свое частное облако или стать поставщиком общедоступного облака [22].

Хранилище İaaS. Услуги хранения позволяют компаниям хранить данные на устройствах хранения сторонних провайдеров. Облачное хранилище доступно онлайн и представляется клиентам в виде набора пулов или сегментов памяти, доступных с помощью богатых интерфейсов, таких как API-интерфейсы программирования, веб-интерфейсы или инструменты командной строки. Сложность архитектуры облачного хранилища скрыта от клиентов, но на внутреннем уровне она довольно сложна - обычно состоит из распределенных устройств хранения, которые управляются централизованным программным обеспечением. Сложное программное обеспечение для управления хранением использует алгоритмы, которые управляют данными, распределенными по нескольким устройствам хранения[20].

Однако облачное хранилище может не соответствовать потребностям каждой организации. Потенциальные недостатки включают задержку в сети, зависимость от доступности Интернета, соображения безопасности и ограниченный контроль. Задержка в сети выше по сравнению с внутренним хранилищем, поскольку центр обработки данных облачного провайдера расположен в другом географическом месте, по крайней мере, в нескольких сетевых скачках от местоположения клиента. Клиент, который хранит все данные в публичном облаке и не имеет локальной копии, полностью зависит

от подключения к Интернету. Облачный провайдер должен обеспечивать высокий уровень безопасности, чтобы избежать потери или компрометации информации, а передача данных должна быть зашифрована.

Характеристики типичного облачного хранилища:

• Высокая надежность и избыточность

• Автоматически масштабируемый

• Предоставление самообслуживания для клиентов

• Доступ через богатые интерфейсы (веб-консоль, API, CLI)

• Платежная модель с оплатой по факту

Цены на облачное хранилище. Большинство провайдеров облачных хранилищ определяют цены, используя следующие факторы:

• Общее количество хранимых данных

• Количество данных, переданных в облачное хранилище

• Количество данных, переданных из облачного хранилища

• Количество конкретных http запросов

Сеть İaaS. Существует два основных сетевых сервиса, предлагаемых провайдерами общедоступного облака: Балансировка нагрузки и DNS (domain name systems).

Балансировка нагрузки обеспечивает единую точку доступа для нескольких серверов, которые работают за ней. Балансировщик нагрузки - это сетевое устройство, которое распределяет сетевой трафик между серверами, используя специальные алгоритмы балансировки нагрузки [9]. Существует множество различных алгоритмов распределения нагрузки, хотя наиболее популярными являются следующие:

1. Циклический перебор: равномерное распределение соединения по всем

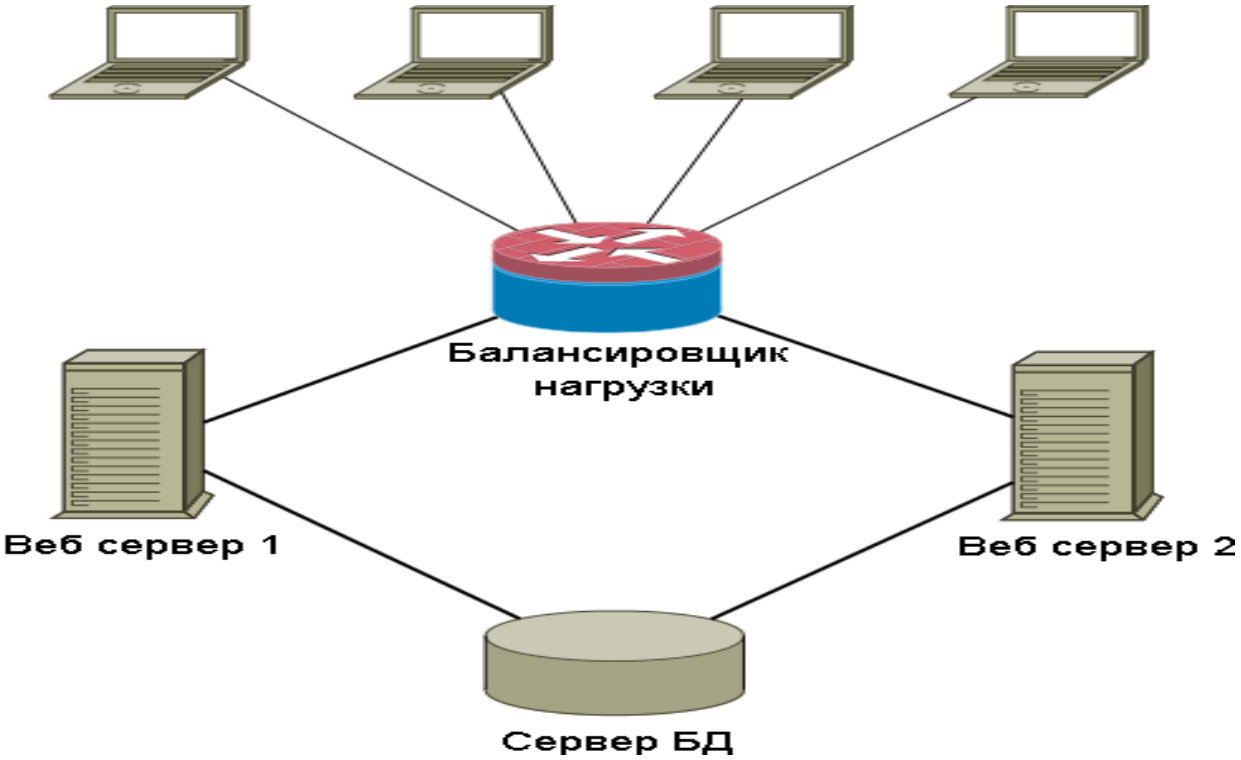
серверам

2. Взвешенный циклический перебор: распределение соединения пропорционально весу, назначенному для каждого сервера

3. Динамический циклический перебор: аналогично взвешенному циклическому перебора, но вес сервера определяется динамически на основе непрерывного мониторинга сервера

4. Минимальное соединение: соединение отправляется на сервер с наименьшим количеством текущих соединений

5. Быстрота: распределяет новые подключения к серверу на основе самого быстрого времени ответа сервера



Балансировка нагрузки имеет несколько преимуществ: отработка отказа - в случае сбоя конкретного сервера балансировщик нагрузки автоматически перенаправляет сетевой трафик на другие серверы; производительность - поскольку нагрузка трафика распределяется между несколькими серверами,

время отклика сети обычно быстрее; масштабируемость - клиенты могут быстро добавлять серверы под балансировщик нагрузки для увеличения вычислительной мощности без влияния на другие компоненты сети / системы.

Облачные сервисы с балансировкой нагрузки обеспечивают распределение нагрузки только для внутренних экземпляров, развернутых в облаке провайдера, причем цена услуги определяется часами обслуживания плюс объем используемой полосы пропускания [15]. Различные поставщики услуг балансировки нагрузки могут предлагать разные технические функции.

Ниже приведена таблица, где мы сравниваем балансировщики нагрузки Amazon и Rackspace.

Свойство	**Amazon**	**Rackspace**
Публичный IP-адрес	доступен	доступен
Список контроля доступа	Не доступен	доступен
Регистрация соединения	Не доступен	доступен
Терминации SSL	доступен	Не доступен
Дросселирование соединения	Не доступен	доступен
LB алгоритмы	только циклический	циклический,случайный,взвешенный rr, наименьший conn.
Внутренние виртуальные IP-адреса	Не доступен (только в VPC)	доступен

Поддержка IPv6	доступен	доступен

Система доменных имен (DNS)

DNS - это иерархическая система именования компьютеров или любых других устройств именования, которые используют IP-адресацию для идентификации сети - система DNS связывает доменные имена с IP-адресами. Службы DNS облачного провайдера предлагают хранить сопоставления IP / доменных имен как для внутренних, так и для внешних серверов. Модели ценообразования могут отличаться у разных поставщиков; некоторые (например, Amazon) взимают плату за зону, размещенную на DNS, и количество просмотров DNS, в то время как другие (например, Rackspace) предлагают услуги DNS бесплатно для своих существующих клиентов [7].

Существует несколько различий между функциями DNS, предлагаемыми облачными провайдерами, и внутренними службами DNS:

• Облачные провайдеры предлагают API управления DNS для разработчиков. Эти API позволяют разработчикам автоматизировать операции DNS, такие как аутентификация и создание доменной зоны.

• Высокая доступность и автоматическая масштабируемость

• Расширенная безопасность и контроль доступа

• Геолокационная DNS-маршрутизация

IaaS: Управление облаками

Службы управления облаком (CMS), как правило, заполняют некоторые пробелы в инструментах управления облаком крупных провайдеров облачных услуг. Крупные провайдеры облачных услуг концентрируются на ядре своего бизнеса, предоставляя клиентам масштабируемую и надежную облачную

инфраструктуру, но в некоторых случаях поставщики могут не иметь ресурсов (или склонности) для разработки дополнительных служб управления облаком. Крупные облачные провайдеры обычно используют «достаточно хороший» подход к дополнительным инструментам управления облаком, поскольку эти инструменты не вносят непосредственного вклада в доход. Поставщики CMS работают напрямую с конкретными облачными провайдерами, чтобы интегрировать услуги провайдера с существующим облаком клиента или создавать дополнительные услуги на основе существующей инфраструктуры провайдера [7].

Заказчики получают преимущества от дополнительных возможностей, предоставляемых службами управления облаком, которые могут включать в себя службы интеграции, службы безопасности и управления доступом, функции высокой доступности и службы репликации баз данных. CMS можно разделить на три основные ветви: облачные услуги управления добавленной стоимостью, услуги облачной интеграции, и брокерские услуги облачного сервиса.

Облачные услуги управления добавленной стоимостью.

Поставщики услуг управления облачной добавленной стоимостью (CMS) расширяют существующие сервисы облачных поставщиков, создавая дополнительные функции на их основе. CMS обычно используются для публичных облачных сервисов; однако некоторые поддерживают частные облачные решения, такие как VMWare или Citrix. Системы управления контентом приносят пользу клиентам, поскольку они экономят время и ресурсы, автоматизируя регулярные задачи, которые в противном случае заказчики должны были бы выполнять самостоятельно [13].

Scalr - это популярный поставщик CMS, который работает напрямую с публичными облаками Amazon и Rackspace и предоставляет дополнительные услуги, такие как отказоустойчивость базы данных и автоматическое

масштабирование, резервное копирование, безопасность, управление DNS и распределенное планирование задач.

Чтобы продемонстрировать ценность, предоставляемую CMS, мы сравниваем сервисы Scalr с сервисами Amazon AWS и анализируем функции AWS, улучшенные Scalr [5].

Свойства	Amazon	Scalr
Масштабируемость и высокая доступность баз данных (MySQL)	База данных master и slave с автоматической репликацией.	Автоматическое обнаружение сбоев мастера БД. Автоматическое масштабирование базы данных создает дополнительные ведомые базы данных на основе условий повышенной нагрузки. Повышение производительности БД за счет разделения запросов на чтение и запись по разным узлам.

Автоматическое масштабирование виртуальной машины EC2	Автоматическое масштабирование доступно через инструменты командной строки	Простой в использовании инструмент веб-интерфейса для определения политик автоматического масштабирования Гибкие политики автоматического масштабирования на основе компонентов Пользовательские метрики автоматического масштабирования Интеллектуальная обработка событий понижения: постепенное отключение, запуск условий предварительного понижения
Резервные копии	Нет доступных сторонних или межоблачных резервных копий	Доступны автономные инструменты резервного копирования WebUI интерфейс для управления резервным копированием
Планировщик	Только стандартный хост-планировщик cron	Централизованный планировщик на основе кластера и хоста с простым управлением через веб-интерфейс.
Логическая группировка серверов	Не доступен	Логическая группировка серверов помогает целостно управлять серверами. Клиенты могут работать с группой серверов вместо того,

		чтобы повторять одну и ту же операцию на отдельных хостах.
DNS управление	Веб-интерфейс доступен для управления записями DNS	Веб-интерфейс на основе политики управления DNS Автоматическое добавление / прекращение записи DNS при создании / удалении компьютерного экземпляра
Безопасность	Веб-интерфейс доступен для управления брандмауэром	Веб-интерфейс для управления брандмауэром. Автоматически изменяет брандмауэр, чтобы разрешить подключения только с других серверов в том же облаке.

Облачные услуги по управлению добавленной стоимостью имеют смысл, поскольку они экономят время и ресурсы для компаний. Задача для поставщиков CMS состоит в том, чтобы всегда быть на шаг впереди, чтобы убедиться, что предоставляемые ими услуги актуальны. Если крупный облачный провайдер реализует некоторые из этих дополнительных услуг, он может легко вывести провайдера CMS из бизнеса. Облачные провайдеры, такие как Amazon и Rackspace, обладают обширным опытом управления облаками и знают, как управлять своими облаками; следовательно, они могут легко реплицировать любые сервисы, созданные на их платформе, если обнаружат, что данный сервис может приносить разумную прибыль [14].

Другая проблема заключается в том, что существует низкий барьер для входа в бизнес CMS, поскольку поставщики CMS обычно используют существующие технологии и API, предлагаемые поставщиками облачных услуг. Компании CMS не создают много собственной интеллектуальной собственности, и их функции программного обеспечения легко копируются.

Услуги облачной интеграции.

Облачные службы интеграции (CIS) обеспечивают интеграцию между частными и общедоступными облаками. Поставщики CIS предлагают уровень абстракции управления, который обеспечивает прозрачную поддержку нескольких облаков - клиенты могут управлять различными облаками как в общедоступных, так и в частных средах, используя централизованные инструменты администрирования и API. CIS может автоматизировать рутинные административные задачи, такие как обеспечение, масштабирование, мониторинг и управление конфигурацией, с помощью простых в использовании инструментов централизованного администрирования [2].

CIS предоставляют инструменты управления несколькими облаками, которые позволяют клиентам отслеживать все журналы, события безопасности и использование ресурсов компьютера с единой панели управления. Кроме того, такие инструменты централизованного управления помогают контролировать доступ пользователей, поддерживать безопасность сервера и проводить аудит безопасности. Пользователи могут устанавливать условные триггеры для автоматической отправки уведомлений в случае достижения определенного предварительно заданного условия.

Административные задачи управления облаком требуют времени и ресурсов, и, хотя есть некоторые сходства, каждая технология управления облаком отличается. Изучение новых инструментов управления облаком аналогично овладению новым языком программирования: требуются усилия для развития значительного уровня знаний. Когда компании используют гибридные облака или облачные технологии от разных поставщиков, эта кривая обучения становится еще круче, потому что есть больше технологий для изучения. Основным преимуществом CIS является то, что они предоставляют прозрачный уровень абстракции управления, который предлагает точно такие же элементы управления для управления несколькими

облаками. Это помогает минимизировать кривую обучения, упростить задачи управления и снизить административные издержки [18].

Rightscale - это крупная компания, предоставляющая услуги облачного управления, предлагающая CIS, одна из первых компаний по управлению облаком на рынке в 2006 году. Клиентами Rightcale являются Associated Press, CBS и Zynga. В настоящее время Rightscale обеспечивает поддержку управления большинством крупнейших общедоступных облаков в мире, таких как Amazon Web Services, Rackspace, SoftLayer, Google Cloud, Windows Azure, Tata и Korea Telecom. Rightscale также поддерживает большинство популярных технологий частного облака, включая CloudStack, OpenStack и Eucalyptus.

Брокерские услуги облачного сервиса.

Брокеры облачных сервисов (CSB) объединяют сервисы, предлагаемые несколькими поставщиками облачных услуг, и организуют эти сервисы в каталог сервисов. Этот каталог услуг предоставляет информацию о продукте с легким поиском для сервисов IaaS, SaaS и PaaS, доступных для покупки через распределительную сеть поставщика CSB. Услуги CSB можно рассматривать как рынок облачных услуг, где клиенты могут быстро определить и приобрести необходимые услуги. Помимо брокерских услуг, эти провайдеры также обеспечивают централизованное управление облачными сервисами аналогично тому, как это делают провайдеры CMS и CIS. Кроме того, CSB могут помочь как поставщикам услуг, так и их клиентам управлять административными задачами, такими как выставление счетов и арбитраж.

Основное различие между брокерами облачных сервисов(CSB), облачными управляемыми сервисами(CMS) и сервисами облачной интеграции(CIS) заключается в том, что любой поставщик облачных сервисов может продавать сервисы через облачные брокерские компании, тогда как компании CMS и CIS работают только с выбранными ими поставщиками.

По мнению многих аналитиков, CSB считается одним из наиболее динамично развивающихся сегментов в отрасли облачных вычислений. «Брокеры облачных сервисов позволяют компаниям легче, безопаснее и продуктивнее ориентироваться, интегрировать, использовать, расширять и обслуживать облачные сервисы, особенно когда они охватывают несколько разных поставщиков облачных сервисов», - Gartner Group [17].

CSB предоставляют множество преимуществ как для облачных провайдеров, так и для клиентов, предоставляя каналы распределения для облачных провайдеров, что может быть особенно полезно для небольших провайдеров, которые не имеют большого числа клиентов. Используя такие каналы распространения, облачные провайдеры могут перепродавать не только свои собственные услуги, но и соответствующие сторонние решения

2.2. Платформа как услуга (Paas)

PaaS (Платформа как услуга) - это модель облачного сервиса, которая предоставляет заказчикам настраиваемую платформу приложений, включая предустановленный программный стек. PaaS можно рассматривать как еще один уровень абстракции над аппаратным обеспечением, операционной системой и стеком виртуализации. Модель PaaS приносит значительную пользу компаниям, поскольку снижает сложность обслуживания инфраструктуры и приложений и позволяет сконцентрироваться на основных аспектах разработки программного обеспечения.

Стоимость разработки программного обеспечения в крупных организациях обычно ниже стоимости программного обеспечения и обслуживания инфраструктуры. Неудивительно, что компании все больше заинтересованы в упрощении своих инфраструктур приложений и промежуточного программного обеспечения для повышения производительности и минимизации связанных с этим операционных расходов.

Характеристики услуг PaaS

- Масштабируемость и автоматическое предоставление базовой инфраструктуры.
- Безопасность и резервирование
- Инструменты сборки и развертывания для быстрого управления приложениями и их развертывания.
- Интеграция с другими компонентами инфраструктуры, такими как веб-сервисы, базы данных и LDAP.
- Многопользовательский, платформенный сервис, который может использоваться многими одновременными пользователями
- регистрация, введение отчетов и инструментарий кода
- Интерфейсы управления и / или API

Сегмент рынка PaaS менее развит, чем IaaS - есть только несколько крупных провайдеров PaaS, а также несколько более мелких нишевых игроков. Хотя для разработки PaaS и SaaS может потребоваться более глубокая техническая экспертиза по сравнению с IaaS, им также не требуется столько первоначальных инвестиций в инфраструктуру. Построить инфраструктуру PaaS или SaaS в существующих облаках IaaS крупных поставщиков, таких как Amazon или Rackspace, достаточно просто и доступно. По этому пути обычно идут небольшие компании, выходящие на рынок PaaS и SaaS.

PaaS (SaaS): Аналитика данных и бизнес-аналитика.

Программное обеспечение для анализа данных и бизнес-аналитики помогает компаниям упростить операции с данными и применять методы статистического анализа для поддержки принятия бизнес-решений и поиска информации.

Аналитика данных и бизнес-аналитика считаются одной из наиболее неиспользованных возможностей в облаке. BI-решения очень дороги, требуют высокого уровня технической сложности и требуют значительного опыта для внедрения и обслуживания. Более того, данные, вероятно, являются наиболее ценным активом любой организации, и безопасность всегда остается в центре внимания других проблем. Не каждая компания готова рисковать и передавать данные в общедоступное облако, даже если данные зашифрованы и облачный провайдер полностью соответствует требованиям. Безопасность является одной из основных причин того, что внедрение облачных вычислений было намного медленнее в аналитике данных и в пространстве BI по сравнению с другими сегментами ИТ. Однако преимущества общедоступного облака часто слишком привлекательны, чтобы их можно было упустить, и они могут перевесить даже деликатные проблемы безопасности [8].

Эластичность. Как обсуждалось ранее, эластичность является основным преимуществом облака, и, как и в любом другом ИТ-сегменте, BI может сильно выиграть от эластичности облачных вычислений. Задачи анализа данных и статистического моделирования требуют огромных вычислительных мощностей - эти типы вычислительных рабочих нагрузок очень сложно предсказать, и поэтому компании всегда должны поддерживать дополнительное аппаратное обеспечение и чрезмерно обеспечивать свою инфраструктуру заранее. Избыточного планирования можно избежать, переместив часть или всю аналитическую инфраструктуру в общедоступное облако. Некоторые компании предпочитают гибридный облачный подход, при котором они выполняют свои вычисления в локальном облаке, но используют публичные облака, когда требуется дополнительная емкость; другие предпочитают перенести свой полный стек аналитической инфраструктуры в облако.

Тонкая линия между BI PaaS и SaaS

Основываясь на нашем определении в начале этого документа, основное различие между PaaS и SaaS заключается в том, что PaaS обычно представляет собой настраиваемую облачную платформу для разработки приложений, в то время как SaaS предоставляет уже разработанное онлайн-приложение. В случае облачных платформ BI (а также некоторых других облачных программ) существует четкая грань между классификацией PaaS и SaaS. Например, предложение Google BigQuery BI можно рассматривать как решение SaaS и PaaS - это SaaS для пользователей, выполняющих запросы к нему, но PaaS для разработчиков, использующих API для его программирования.

PaaS: Интеграция

Платформы облачной интеграции помогают компаниям интегрировать данные из разных источников. Эти источники могут включать локальные базы данных частного облака, базы данных открытого облака, базы данных сторонних поставщиков, системы CRM и системы обмена сообщениями. Интеграция данных является очень сложной и утомительной задачей, и задачи интеграции данных включают поддержание качества данных, совместимости и стандартов.

Dell Boomi является лидером в области интеграции облачных данных. Boomi начиналась как независимая компания, но была приобретена Dell в 2010 году. Boomi объединяет более 40 различных технологий и протоколов, включая Oracle, SAP, Siebel, Salesforce, Hadoop и QuickBooks.

Пользовательский код	**Dell Boomi**
Требуются большие внутренние ресурсы	Плавное обновление

Труд 50% от общей стоимости владения в течение 3 лет	Экономически эффективный
Построен для решения текущих проблем	Автоматические обновления
Нехватка гибкости бизнеса	Масштабируемый, гибкий и простой в использовании
Высокая стоимость обслуживания	Автоматические обновления
Пользовательская безопасность	Встроенная централизованная безопасность
Ручная настройка и резервное копирование	Лучшие практики для настройки бесшовного резервного копирования

2.3. Программное обеспечение как услуга (Saas)

Эта облачная модель — самая распространенная. Как и в других моделях облачной услуги, SaaS предлагает компаниям возможность сократить внутренние расходы на ИТ-поддержку и передать ответственность за обслуживание поставщику SaaS. SaaS, безусловно, является наиболее широко используемой моделью облачной услуги, поскольку почти каждый поставщик программного обеспечения стремится разместить свое предложение на

рельсах SaaS - в каждой категории программных продуктов есть предложения SaaS, и, возможно, потребуется несколько дней, чтобы перечислить все программное обеспечение SaaS [12]. Поэтому в нашей таксономии SaaS мы перечисляем только отдельные группы поставщиков в нескольких категориях.

SaaS - это большой рынок с сильным потенциалом роста. По оценкам Gartner Group, к 2017 году мировой доход от SaaS достигнет 23,5 млрд долларов. Продолжение модели ASP (поставщик сервисов приложений), основные сервисы SaaS появились в конце 1990-х годов, когда такие компании, как Salesforce, предлагали клиентам хостинг и управление своим программным обеспечением. Модель ASP также предлагала централизованный хостинг сторонних приложений, но отличалась от SaaS, потому что хостинг и операции все еще требовали ручного участия. Кроме того, ASP иногда требовал установки толстого настольного клиента (локально работающее программное обеспечение, которое выполняет большинство вычислительных задач на компьютере пользователя, а не на стороне сервера), в то время как приложения SaaS обычно доступны через веб-браузеры или мобильные приложения. ,

Платформы SaaS используют мультитенантные архитектуры, в которых одно и то же оборудование и программное обеспечение платформы совместно используется несколькими клиентами. Многопользовательский хостинг и автоматизация сервисов позволяют провайдерам SaaS поддерживать низкие затраты на обслуживание. В зависимости от поставщика SaaS и типа приложения часто возможна некоторая настройка приложения, но обычно она более ограничена, чем в облачных решениях PaaS и IaaS [12].

Первоначальные идеи SaaS вызывали серьезную озабоченность в отношении безопасности, производительности и доступности услуг. Однако за прошедшие годы технологии и бизнес-модели SaaS значительно окрепли и преодолели многие из этих начальных проблем. Компании сейчас активно используют SaaS, и это вряд ли изменится

Бизнес-модель SaaS может показаться идеальной - она оставляет весь контроль над управлением приложениями в руках поставщика SaaS. Таким образом, поставщик решает, когда обновлять программное обеспечение, когда добавлять новые функции и когда повышать цены.

ГЛАВА III. ОБЗОР ОСНОВНЫХ ПОСТАВЩИКОВ УСЛУГ ОБЛАЧНЫХ СИСТЕМ ОБРАБОТКИ ДАННЫХ

3.1. Основные поставщики услуг облачных вычислений

Мир облачных вычислений постоянно развивается, и новые концепции и технологии создаются быстрыми темпами. Каждый из существующих и новых поставщиков услуг предлагает различные облачные решения, предназначенные для разных категорий клиентов, т.е. приложения SaaS для отдельных пользователей или услуги IaaS для предприятий. Разнообразие создает конкуренцию. Множество доступных облачных решений и поставщиков облачных услуг ускоряет скорость, с которой сервисы облачных вычислений развиваются в функциональном отношении. Кроме того, усиление конкуренции приводит к удешевлению услуг, что приводит к увеличению выгод для клиентов. Несмотря на то, что они ориентированы на разные типы клиентов и используют разные технологии, у поставщиков

облачных услуг есть одна общая цель: предлагать стабильные, безопасные и масштабируемые облачные вычисления, которые облегчают разработку приложений и снижают затраты на управление локальной средой.Примерами основных поставщиков услуг облачных вычислений являются Amazon, Google и Microsoft [19].

Решение Amazon для облачных вычислений - EC2. EC2 следует сервисной модели IaaS. Сервисы EC2 ориентированы на предоставление простой в использовании и масштабируемой среды, предназначенной для разработчиков приложений. Как следует из названия, платформа IaaS достаточно гибкая, чтобы позволить клиентам масштабировать емкость в течение нескольких минут. Гибкость услуг дополнительно подтверждается и расширяется благодаря другим функциям, которые Amazon предлагает через платформу EC2: гибкие IP-адреса, хранилище гибких блоков и гибкий балансировщик нагрузки. Помимо повышения масштабируемости Amazon повышает безопасность платформы с помощью таких функций, как виртуальное частное облако Amazon, Amazon CloudWatch и различные другие методы обеспечения безопасности. С клиентов EC2 взимается плата за почасовое использование услуг и за объем передаваемых данных. Другим важным аспектом является то, что сервисы EC2 предназначены для использования с другими сервисами Amazon, такими как S3 и Amazon SimpleDB. Сервисы Amazon EC2 предоставляют виртуализированные экземпляры множества баз данных и операционных систем, включая Microsoft Windows и несколько дистрибутивов Linux, что позволяет разработчикам создавать приложения с использованием любого языка программирования, поддерживаемого доступными операционными системами.

Google предлагает Google App Engine (GAE). Он принимает модель PaaS и ориентирован на разработку веб-приложений. Грубая разбивка сервисов GAE от Google объясняется тем, что клиенты могут запускать свои собственные веб-приложения в инфраструктуре Google. GAE предоставляет

интегрированные инструменты, обеспечивающие поддержку на протяжении всего жизненного цикла разработки приложений. Эти инструменты позволяют легко управлять хранением, распределением, балансировкой нагрузки и мониторингом. В то время как Amazon фокусируется на рекламировании своих сервисов как на легко масштабируемых, Google подчеркивает легкость внедрения их облачного решения. GAE имеет такие функции, как автоматическая масштабируемость, и, удобно для разработчиков, несколько других облачных приложений Google, таких как Gmail и Google Docs, интегрированы в платформу. Поначалу размещение и запуск приложения на платформе GAE ничего не стоит. Google придерживается своей политики предоставления доступа к бесплатным ресурсам. Неоплачиваемые учетные записи пользователей могут хранить до 1 гигабайта данных и могут использовать достаточную пропускную способность для поддержки до 5 миллионов посетителей в день и различных других бесплатных ресурсов. В отличие от биллинговой системы, представленной в EC2 и MWA, в которой с клиентов взимается почасовая оплата, клиентам выставляются счета только за фактически используемые ресурсы, то есть вычисляют часы и объем передачи данных. GAE поддерживает только две среды программирования, для Java и Python. Ограниченное количество доступных языков программирования может подтолкнуть потенциальных клиентов к принятию других облачных решений, таких как предлагаемые Amazon или Microsoft.

MWA - это прикладная платформа Microsoft, доступная для широкой публики, которая предоставляет ресурсы в общедоступной облачной инфраструктуре. MWA предлагается в виде двух моделей предоставления услуг, PaaS и IaaS. Microsoft не фокусируется только на одной характеристике платформы MWA, вместо этого выделены несколько основных функций. Для разработчиков важнее всего то, что MWA - это открытая платформа, а это означает, что клиенты могут использовать любую операционную систему и могут развертывать приложения, разработанные с использованием любого

языка программирования. Кроме того, MWA поддерживает три различных типа хранилищ: базы данных на языке структурированных запросов (далее SQL), таблицы noSQL и двоичное хранилище. Политика выставления счетов позволяет клиентам бесплатно зарегистрировать пробную учетную запись сроком на три месяца. Пробная учетная запись включает в себя одну полностью работающую базу данных SQL и хранилище данных до 35 ГБ, а также другие бесплатные ресурсы. MWA принимает открытый подход к разработке приложений. Это дает клиентам возможность адаптировать свои приложения на основе их спецификаций, а не ограничений, налагаемых поставщиком облачных услуг. Кроме того, MWA имеет несколько функций, которые недоступны на других облачных платформах, например, Access Control и Service Bus. Платформа MWA находится в центре внимания этого исследования и будет обсуждаться в данной главе, включая технические детали и последствия для бизнеса [21].

В таблице 1 представлено сравнение между тремя платформами облачных сервисов, рассмотренными выше. Данные, представленные в таблице, взяты из официальной документации, предоставленной тремя поставщиками облачных услуг, по состоянию на апрель 2020 года. Сравнение проводится с технической и функциональной точек зрения.

Таблица 1. Сравнение между различными платформами облачных сервисов

	Amazon EC2	**MWA**	**GAE**
Тип развертывания	IaaS	PaaS/IaaS	PaaS
Поддерживаемые ОС	Red Hat Enterprise Linux, SUSE Linux Enterprise, Fedora, Oracle	Windows Server, CentOS, Ubuntu, SUSE Linux Enterprise, openSUSE, Guest OS	Linux

	Enterprise Linux, various Linux distributions, Windows Server, Amazon Linux AMI		
Поддерживаемые языки	.NET, C++, PHP, Ruby, Python, Java, Android, iOS	.NET, C++, PHP, Ruby, Python, Java, Windows Phone, iOS	Java, Python, Go, PHP
Хранилище	MySQL, MongoDB, Microsoft SQL Server, Postgre SQL, CouchDB + others	Azure SQL, нереляционные таблицы, очереди, blobs	Google Cloud SQL, Cloud Storage, App Engine Datastore
Приложения запускаются в виртуальных машинах	Да	Да	Нет
Интерфейс управления	API, Command Line, User Interface (UI)	API, Command Line, Management Portal	API
Балансировщик нагрузки	Да	Да	Да
Подходит для развития малых размеров приложений	Да	Нет	Да
Подходит для развития больших размеров приложений	Да	Да	Да

Средства мониторинга	CloudWatch	Интернет-портал управления, 3rd Party	Appstats for Java, Appstats for Python
Модель ценообразования	почасовой, планы подписки	почасовой, планы подписки	на основе использования
Бесплатные ресурсы	30 ГБ для хранения, 15 GB исходящего трафика ежемесячно, в течение одного года	3-месячная пробная версия с доступом ко всем функциям, но с ограниченными ресурсами	1 Гб для хранения, 1 Гб исходящего трафика ежемесячно

Чтобы понять концепцию безопасности в среде облачных вычислений и ее последствия, необходимо определить три основных принципа информационной безопасности. Основные принципы определены достаточно широко, чтобы охватить все политики безопасности, которые может использовать организация. Кроме того, все возможные типы нарушений безопасности, которым может подвергаться организация, включены в определения, представленные ниже. Эти три основополагающих принципа это - конфиденциальность, целостность, доступность. Во-первых, конфиденциальность представляет собой возможность защиты информации от раскрытия, включая средства, с помощью которых достигается конфиденциальность. Утрата конфиденциальности представляет собой преднамеренное или непреднамеренное разглашение конфиденциальной информации. Второй принцип безопасности - это целостность. Целостность представляет возможность сохранения информации в оригинальном формате, гарантируя подлинность информации. Потеря целостности может возникнуть в результате преднамеренного или случайного изменения информации. Последний принцип, включенный в определение концепции безопасности, - это доступность. Доступность гарантирует, что информация доступна уполномоченным сторонам из указанных мест и с помощью указанных

средств. Потеря доступности приводит к временному или окончательному нарушению доступа к информации.

Также необходимо обсудить и другие принципы, чтобы детально определить безопасность. Идентификация и аутентификация - это процессы, обеспечивающие доступ к конкретной информации только авторизованным, законным пользователям. Авторизация представляет разрешения, которые устанавливают диапазон пользователя с точки зрения доступа к информации, обычно достигаемой путем развертывания списков контроля доступа (ACL). Доверие и надежность также являются понятиями, которые трудно определить самим, но их необходимо учитывать при определении концепции безопасности. Провайдеры облачных услуг должны создать прочные доверительные отношения с клиентами, чтобы поддерживать предпосылку облачных вычислений о том, что облачные ресурсы надежны и доступны в любой момент. Другим важным аспектом безопасности является конфиденциальность. Конфиденциальность может быть сложно определить из-за различных глобальных организаций, представляющих разные определения. Конфиденциальность представляет собой уровень конфиденциальности и защиты частной информации и обеспечивается в соответствии с региональными законами.

За безопасность сред облачных вычислений и информации, связанной с такими средами, отвечает как поставщик услуг, так и заказчик. В большинстве случаев поставщик услуг отвечает за поддержание безопасности облачной инфраструктуры, включая физические, логические и сетевые ресурсы, тогда как заказчик несет ответственность за безопасность приложений и информации, хранящейся в облачной среде. Безопасность жизненно важна для сред облачных вычислений, и все меры безопасности должны применяться постоянно, независимо от того, получает ли клиент доступ к ресурсам. Среды облачных вычислений доступны через общие интернет-протоколы, поэтому

они подвержены распространенным атакам, уязвимостям и угрозам, влияющим на интернет-коммуникации [21].

Нарушения безопасности могут быть определены с точки зрения потери любого из принципов, как обсуждалось выше. Развертывание приложений в облачной среде подразумевает предоставление третьей стороне данных, критически важных для благополучия бизнеса. Таким образом, нарушения безопасности могут иметь разрушительные последствия для ресурсов организации. Воздействие нарушений безопасности может быть низким, средним или высоким в зависимости от ущерба, нанесенного физическим ресурсам организации. Кроме того, нарушения безопасности могут оказать негативное влияние на организацию с точки зрения доверия или потери доверия со стороны общественности. В попытке обеспечить безопасность облачной среды, поставщик облачных услуг, клиент или обе стороны рекомендуют принять ряд мер безопасности. Общие меры безопасности должны включать поддержание в актуальном состоянии протоколов сетевой безопасности, внедрение служб сетевой аутентификации и шифрования данных, а также внедрение брандмауэров и систем обнаружения вторжений. Принципы безопасности, которые рекомендуется учитывать при разработке приложений для облака, включают принцип наименьших привилегий, разделение обязанностей, принцип отказоустойчивости и принцип самого слабого звена.

В то время как среды облачных вычислений создают различные преимущества и возможности для разработчиков благодаря успешному решению таких проблем, как высокая стоимость высокопроизводительного оборудования, риски безопасности, связанные с развертыванием приложения в облачной среде, значительно возрастают по сравнению с локальным запуском приложения. Однако, приняв несколько известных, рекомендуемых политик и практик, разработчики могут снизить риски безопасности и

минимизировать влияние в случае нарушения безопасности приложения или облачной среды.

3.2 Microsoft Azure как облачная платформа

Microsoft Azure (ранее известный как Windows Azure) - это общедоступная облачная платформа, созданная и размещаемая Microsoft. Он был выпущен 1 февраля 2010 года и с самого начала должен был конкурировать на относительно молодом рынке, где доминируют Amazon Web Services. За прошедшие годы Azure повзрослела, и по состоянию на май 2014 года она стала одним из лидеров в облачных вычислениях, все еще далекая от достижения Amazon, но уже опережая другие конкуренты. Одним из факторов, способствующих такому прогрессу, была политика относительной технологической открытости Microsoft. Компания не ограничивала доступную операционную систему семейством Windows, но поставляла машины с различными дистрибутивами Linux, часто с предустановленным популярным программным обеспечением. Более того, разработчики не ограничиваются поддерживаемыми Microsoft языками, такими как C #, Vision Basic или C ++, но и другими популярными, например, Java, PHP, Python или Ruby. Управление учетной записью Azure возможно через веб-приложение под названием Microsoft Azure Management Portal или через различные API-интерфейсы.

Тем не менее, благодаря платформе MWA Microsoft предоставляет клиентам не только услуги вычислений и хранения. В отличие от решений, предоставляемых конкурентами Microsoft, которые предлагают либо сырые ресурсы, либо очень ограниченные средства разработки, платформа MWA обеспечивает более управляемый интерфейс. Сервисы и инструменты, предоставляемые MWA, призваны помочь разработчикам в разработке легко масштабируемых и легко управляемых приложений. Услуги и поддержка, предлагаемые MWA, подходят для разработки широкого спектра приложений, включая SaaS или корпоративные приложения.

MWA принимает модели IaaS и PaaS. Это набор облачных сервисов, которые предоставляют разработчикам средства для создания масштабируемых приложений. Три основных компонента инфраструктуры MWA - это вычисления, хранилище и матричный контроллер [24].

Разработка приложений с использованием MWA требует другого подхода в зависимости от способа разработки сервисов MWA. Технологии и возможности платформы MWA можно обобщить следующим образом. Во-первых, вычислительная служба обеспечивает вычислительную мощность для запуска приложений, которые размещены на отдельных виртуальных машинах, работающих под управлением ОС, выбранной разработчиком, то есть Windows Server или Ubuntu. Однако сами виртуальные машины работают в ОС Windows Azure. Затем служба хранения предоставляет место для хранения на серверах в дата-центрах, принадлежащих Microsoft. Наконец, контроллер фабрики представляет приложение, которое управляет связью между вычислительными и сервисами хранения.

MWA - облачная платформа, созданная и поддерживаемая Microsoft. По сути, облачные платформы предоставляют услуги вычислений и хранения. Сервисы, предоставляемые платформой MWA, предоставляются в виде сервисов PaaS и IaaS и предоставляют разработчикам средства для создания, развертывания и управления приложениями. В отличие от традиционной разработки приложений, использование услуг, предлагаемых MWA, может быть полезным для разработчиков программного обеспечения. Интегрированная в MWA технология обеспечивает быстрый процесс развертывания и обеспечивает эффективный метод разработки облачных приложений за счет внедрения автоматизированных сервисов [22].

В отличие от ограниченных облачных сервисов, предлагаемых двумя наиболее важными конкурентами против Microsoft, а именно Amazon и Google, платформа MWA позволяет разработчикам использовать большой выбор языков программирования и операционных систем. Кроме того,

инструменты разработки и поддержка отличают платформу MWA от других облачных платформ и создают более управляемые возможности. Важные аспекты разработки облачных приложений подчеркиваются в модели программирования, принятой MWA. В MWA приложения создаются исходя из того, что любой части приложения может быть назначена одна из трех доступных ролей. Приложения должны быть оптимизированы для запуска нескольких экземпляров каждой роли, чтобы другие экземпляры успешно обрабатывали оставшуюся часть задач, в конечном счете, сбои одного экземпляра. Таким образом, разработчики программного обеспечения могут создавать легко масштабируемые, доступные и легко управляемые приложения.

Хотя платформа MWA предлагает множество преимуществ для разработчиков программного обеспечения, важным выводом является то, что платформа MWA не подходит для разработки небольших приложений. Услуги, предоставляемые платформой MWA, поддерживают разработку приложений любого размера и удовлетворяют потребности многих разработчиков программного обеспечения с технической и функциональной точек зрения. Однако модель ценообразования на услуги MWA не отражает намерения Microsoft привлечь ISD. Ежемесячные затраты на запуск приложения небольшого размера в MWA могут достигать или превышать значения, близкие к 100 €, тогда как Google предлагает разработчикам достаточно бесплатных ресурсов для запуска приложения в течение неограниченных периодов времени. Однако, могут возникнуть проблемы с низкой производительностью, если приложения не оптимизированы в соответствии с рекомендуемой моделью программирования MWA [21].

Кроме того MWA является безопасной облачной средой, выделяя важнейшие функции безопасности. Центры обработки данных, управляемые Microsoft, обеспечивают стабильную и безопасную облачную среду, поскольку Microsoft использует различные всемирно признанные

сертификаты безопасности и лучшие практики безопасности. В связи с этим Microsoft оказывает поддержку разработчикам в том, как повысить безопасность и производительность облачных приложений. Платформа MWA представляет собой решение Microsoft для облачных сервисов. Поскольку платформа часто обновляется, появляются дополнительные возможности для исследования новых или будущих технологий, интегрированных в платформу.

Модели выполнения. Модели выполнения - это способы, с помощью которых выполнение приложений обрабатывается Azure. Они отличаются простотой использования, универсальностью, свободой управления и предполагаемым типом приложений. В Microsoft Azure доступно четыре модели выполнения:

- Облачные сервисы
- виртуальные машины
- веб-сайты
- Мобильные услуги

Облачные сервисы. Облачный сервис обеспечивает относительно простой способ размещения приложения в облаке. Это решение PaaS, поэтому оно имеет удобные функции, но в то же время предъявляет ряд требований. Многие задачи автоматизированы или упрощены, например, автоматическая установка системных исправлений и выкатывание новых версий образов. Кроме того, вам не нужно беспокоиться о конфигурации системы - вам нужно только предоставить пакет с вашим приложением и конфигурацией. Конфигурация должна, среди прочего, содержать список «ролей», используемых приложениями. Проще говоря, они являются предварительно сконфигурированными виртуальными машинами и существуют в двух вариантах:

1) Веб-роль - предназначена для размещения веб-приложений,

с установленным IIS

2) Рабочая роль - лучше всего подходит для серверных услуг

Уровень облачной службы представляет ресурсы, которые обеспечивают вычислительную мощность для приложения, развернутого в среде Windows Azure. Сам облачный сервис представляет собой сервис, который обеспечивает вычислительную мощность и содержит фактический код приложения. В MWA приложения делятся на три компонента, называемые ролями, а именно веб-ролями, рабочими ролями или ролями виртуальных машин. Веб-роли представляют собой внешний интерфейс приложения, с которым пользователи могут взаимодействовать, тогда как рабочие роли - это внутренний компонент, часть приложения, которая работает в фоновом режиме и не видна с точки зрения пользователей. Любое приложение MWA должно быть реализовано как одна роль, или комбинация ролей, и одновременно могут выполняться несколько экземпляров любой роли приложения. Кроме того, веб-роли и рабочие роли функционируют независимо, и каждая роль выполняется в своей отдельной виртуальной машине. Несмотря на то, что они работают в разных виртуальных машинах, разные роли могут взаимодействовать друг с другом, но пользователи приложений не могут напрямую взаимодействовать с рабочей ролью [20].

Веб-роли. Веб-роль представляет собой интерфейс приложения, размещенного в облачной среде. Веб-роли похожи на веб-сайты и могут также включать веб-сервисы. Они могут инициировать или получать запросы связи через IIS. Веб-роль представляет собой сервис, который обеспечивает взаимодействие с пользователем и служит точкой соединения между пользовательской и рабочей ролями, обычно графическим интерфейсом веб-сайта. Кроме того, веб-роли можно использовать для размещения приложений, разработанных на любом из языков, поддерживаемых MWA, будь то ASP.NET, PHP или другие.

Рабочие роли. Рабочая роль представляет собой часть облачной службы, которая работает в фоновом режиме. Он не имеет пользовательского интерфейса, и его целью является выполнение фоновых действий, таких как обработка данных или видео. Рабочие роли похожи на службы Windows, которые работают в фоновом режиме локальной системы. Этот тип роли может использоваться для размещения сервера для любого языка программирования, поддерживаемого MWA, однако рабочие роли могут быть разработаны только с использованием .NET.

Примером приложения, которое использует по крайней мере одну веб-роль и одну рабочую роль, является веб-сайт с поисковой системой. Веб-роль представляет собой интерфейс веб-сайта, который является интерфейсом, через который пользователь может взаимодействовать с самим веб-сайтом. Рабочая роль представляет функцию поисковой системы. Когда пользователь инициирует поисковый запрос, рабочая роль обрабатывает данные и возвращает результаты в веб-роль. Если одна из ролей не может обработать все пользовательские запросы, разработчик может легко увеличить роль, увеличив число экземпляров, в которых выполняется соответствующая роль. Чтобы дополнительно подчеркнуть свойства масштабируемости MWA, приложение может назначить две разные рабочие роли для двух действий, даже если действия идентичны. При увеличении количества экземпляров для роли учитывайте влияние затрат, поскольку каждый экземпляр роли потребляет дополнительные ресурсы [19].

Виртуальные машины. Сервисы VM выделяют возможности IaaS, предоставляемые платформой MWA. Виртуальные машины предоставляют мощные ресурсы обработки для разработки новых приложений или миграции новых. Эта вычислительная служба может запускать ОС, отличные от Windows Azure, такие как Windows Server 2012 или различные дистрибутивы Linux. Можно создать несколько виртуальных машин одновременно, и

каждая виртуальная машина может быть настроена для работы с другими ролями.

Виртуальные машины - это решение IaaS, доступное в Microsoft Azure. Они представляют собой концепцию, отличную от облачных сервисов, но на самом деле это просто другой тип «ролей» (PersistentVMRole). Как и в случае с другими ролями, каждая виртуальная машина должна быть размещена в развертывании, которое, в свою очередь, размещено в облачной службе. Виртуальные машины обеспечивают высочайшую степень гибкости, но имеют очень ограниченные возможности размещения приложений. Это постоянные виртуальные хосты, которые можно создавать, запускать, останавливать и удалять по требованию. Пользователь может создавать виртуальные машины, выбрав один из нескольких доступных образов ОС или загрузить свой собственный образ. Затем необходимо выбрать размер экземпляра виртуальной машины. С пользователя взимается плата за каждую минуту работы экземпляра. После выбора размера экземпляра необходимо ввести имя пользователя и пароль администратора. Затем необходимо выбрать уникальное имя хоста в домене .cloudapp.net. Машина будет видна под этим именем хоста из интернета. Следующим шагом является выбор региона, Affinity Group или виртуальной сети, в которой будет размещена виртуальная машина.

Услуги хранения. Microsoft Azure предоставляет способ хранения данных, которые могут совместно использоваться развернутыми приложениями, виртуальными машинами или доступны извне, которые обычно используются или создаются развернутыми приложениями. С пользователя взимается плата за количество хранимых данных, выбранные параметры репликации, количество запросов на чтение и запись и передачу данных за пределы региона службы хранения. Существует 4 так называемых службы хранения:

1. Blob хранилище

2. Хранилище очереди

3. Таблица хранения

4. Файловое хранилище

Blob хранилище предназначен для хранения больших объемов неструктурированных двоичных данных. Это делает его наиболее универсальной службой хранения, доступной в Windows Azure. Данные хранятся в виде капель. Есть два типа капель:

1. Block blob оптимизирован для потоковой передачи и хранения облачных объектов. Это хорошо для эффективной загрузки больших объемов данных.

2. Page blob - страничный блоб оптимизирован для произвольного доступа. Это коллекция из 512-байтовых страниц. Страницы BLOB-объектов обычно используются для хранения образов жесткого диска.

Хранилище очереди. Хранилище очередей предоставляет решение для обмена сообщениями, обычно используемое для связи между отделенными компонентами приложения. Это дает пользователю надежный интерфейс Get \ Put \ Peek. Сообщение, вставленное в очередь, может содержать до 64 КБ двоичных или текстовых данных. Практически все операции управления Azure можно выполнять через веб-портал управления Microsoft Azure, но для этого требуется, чтобы пользователь просматривал все меню и окна. Microsoft предлагает несколько способов автоматизации управления Azure с помощью нескольких API:

• Собственные SDK для .NET, Java, Python, Ruby, Node.js, PHP и мобильных платформ (iOS, Android, Windows Phone)

• Командлеты PowerShell

• Кроссплатформенный интерфейс командной строки

• REST API

База данных Azure SQL. База данных Azure SQL - это облачная реляционная база данных, разработанная и поддерживаемая Microsoft. Основанный на технологии SQL Server , Azure SQL - это масштабируемая, высокодоступная и безопасная служба базы данных. В отличие от развертывания базы данных локального сервера SQL, для развертывания базы данных SQL Azure требуется, чтобы разработчик управлял логическим сервером. Microsoft управляет и поддерживает физические ресурсы, необходимые для работы базы данных SQL Azure. Кроме того, поскольку разработчики не имеют контроля над физическими ресурсами, они не могут выбрать физический жесткий диск, на котором хранятся файлы базы данных. Однако конечная точка, через которую можно получить доступ, остается прежней [22].

Базы данных Azure SQL выигрывают от плавного автоматического устранения сбоев оборудования и автоматической репликации данных. Кроме того, Azure SQL предназначен для повышения производительности и высокой масштабируемости. Разработчики могут создавать и развертывать базу данных в течение нескольких минут, а размер базы данных автоматически масштабируется при достижении лимита хранилища, если позволяет тарифный план.

Служба Azure SQL находится за сетевым брандмауэром. По умолчанию все подключения к серверу Azure SQL отключены, за одним исключением. IP-адрес, который использовался для создания сервера SQL, имеет доступ, но только через порт 1433. Разработчики должны перейти к меню конфигурации сервера баз данных SQL Azure на портале управления и указать, какие IP-адреса могут обращаться к серверу базы данных. Кроме того, брандмауэр базы данных позволяет разработчикам блокировать доступ приложений MWA к конкретным базам данных.

Еще одна функция безопасности - проверка сертификата. Базы данных SQL Azure принимают сообщения только в зашифрованном виде и отклоняют соединения от приложений с недействительными сертификатами безопасности. Кроме того, установленные соединения с Azure SQL сбрасываются каждый час, поэтому разработчики должны повторно вводить учетные данные администратора базы данных.

Azure AppFabric. Azure AppFabric является важным компонентом архитектуры MWA и предоставляет службы, которые отделяют платформу MWA от других облачных платформ. Он представляет средний уровень платформы MWA, служа мостом связи между приложениями, работающими в разных системах. Одно из преимуществ использования сервисов AppFabric заключается в том, что он упрощает процесс управления приложениями, тем самым повышая производительность разработки. Двумя основными компонентами AppFabric являются служебная шина и контроль доступа. Разработчики могут использовать службы, предоставляемые AppFabric, для безопасного соединения компонентов приложения вместе, то есть различных ролей, и простого управления доступом к ресурсам. Эти две службы необходимы для большинства облачных приложений, и они полагаются и дополняют друг друга для достижения наилучших результатов подключения.

Служебная шина обеспечивает инфраструктуру, необходимую для создания облачных приложений, которые должны взаимодействовать с клиентами или другими приложениями через Интернет. Служба обмена сообщениями, предоставляемая сервисной шиной, упрощает процесс разработки приложений, позволяя разработчикам воспользоваться встроенными функциями, такими как открытые конечные точки. Служебная шина построена вокруг централизованной службы ретрансляции. Она функционирует таким образом, что приложения отправляют сообщения на саму шину, а затем любое приложение с соответствующими правами доступа может подключиться к шине и получить любое сообщение. Используя

общедоступные конечные точки в качестве места назначения для сообщений, приложения не участвуют в прямой связи с клиентом, который делает запрос. Вместо этого служебная шина открывает однонаправленный или двунаправленный канал связи, который обеспечивает средства для преодоления общих проблем безопасности связи, таких как обход сетевых брандмауэров или необходимость включения закрытых портов связи. Возможности служебной шины могут обеспечить преимущества для любого приложения, но могут быть особенно полезны для обмена мгновенными сообщениями, обмена файлами или аналогичных приложений [24].

Служба контроля доступа (ACS) предоставляет разработчикам простой способ аутентификации и авторизации клиентов, которым требуется доступ к облачному приложению. Благодаря использованию ACS разработчикам больше не нужно внедрять код аутентификации в свои приложения. Кроме того, ACS позволяет разработчикам внедрять открытые протоколы аутентификации. Это означает, что приложения могут поддерживать аутентификацию с учетными записями, отличными от тех, которые созданы для самого приложения. Клиенты могут использовать учетные записи, предоставленные несколькими популярными веб-сайтами, такими как Google или Facebook, которые называются поставщиками удостоверений.

Контроль доступа не требует от разработчиков изменения кода приложения для реализации ACS. Вместо этого разработчикам необходимо управлять доверительными отношениями с различными поставщиками удостоверений. Функции, предоставляемые контролем доступа, эффективно интегрированы со служебной шиной, что значительно облегчает сложный процесс разработки собственных механизмов безопасности.

3.3.Безопасность облачной среды Azure.

Безопасность платформы MWA находится в центре внимания этой главы. Сначала представлен обзор служб безопасности, реализованных в MWA, и обсуждаются общие проблемы безопасности. Затем исследуются инструменты мониторинга и диагностики. Наконец, обсуждается процесс разработки безопасности, предложенный Microsoft.

Обзор служб безопасности Microsoft Windows Azure.

MWA предназначен для уменьшения ответственности за управление инфраструктурой, что позволяет разработчикам сосредоточиться на разработке приложений. Microsoft стремится обеспечить высокие стандарты CIA, поэтому в платформе MWA реализовано несколько механизмов безопасности. Во-первых, конфиденциальность данных повышается благодаря механизмам аутентификации, таким как безопасная аутентификация на портале управления и уникальные, случайно сгенерированные ключи учетной записи хранения. Более того, данные физически или логически разделены на серверах Microsoft, что обеспечивает повышенную безопасность в случае нарушения безопасности одного сервера. Разработчики также могут шифровать свои данные с использованием надежных алгоритмов шифрования. Во-вторых, основным механизмом поддержания целостности данных является способ копирования данных на три виртуальных жестких диска. Данные, хранящиеся на двух из трех виртуальных жестких дисков, не могут быть удалены или перезаписаны, в то время как данные на основном виртуальном жестком диске защищены с помощью файла конфигурации, загруженного разработчиком при развертывании приложения. Разработчики могут ограничить доступ к определенным ролям приложения через этот файл конфигурации. Наконец, доступность данных обеспечивается с помощью автоматических механизмов копирования данных [19].

На уровне платформы и инфраструктуры платформа MWA автоматически устраняет несколько угроз безопасности, не требуя никаких

действий со стороны разработчика. Первой мерой безопасности, применяемой на этих уровнях, является брандмауэр, который блокирует все не специально открытые коммуникационные порты, включается на каждой виртуальной машине, выполняющей роль приложения. Другая серьезная угроза безопасности, атаки типа «отказ в обслуживании» (DoS), особенно хорошо обрабатываются MWA. Благодаря автоматическим службам балансировки нагрузки и высокой масштабируемости за счет увеличения количества образцов ролей MWA может легко обрабатывать избыточный сетевой трафик.

Сервисный уровень подвержен распространенным интернет-атакам. Поскольку приложения доступны клиентам через интернет-протоколы, Microsoft может обеспечить только безопасность облачной среды. Поэтому разработчики обязаны применять лучшие практики при написании кода для приложений. На уровне сервисного обслуживания наиболее распространенными мерами безопасности, которые могут использовать разработчики, являются защита данных приложения, аудит и регистрация событий, проверка входных данных, использование действительных сертификатов SSL и шифрование хранимых данных.

Во-первых, данные приложения могут быть защищены путем применения списков ACL, которые ограничивают доступ к самому приложению. Только разработчикам, которым требуется прямой доступ к коду и данным приложения, должны быть предоставлены права доступа. Кроме того, разработчикам рекомендуется разрешать приложению взаимодействовать с пользователями только через определенные порты и только по протоколам HTTPS, когда это возможно.

Во-вторых, аудит и регистрация(logging) - это меры безопасности, которые разработчики должны включить на всех уровнях кода приложения, где работают механизмы аутентификации и авторизации. Это позволяет разработчикам проверять правильность работы приложения и диагностировать проблемы, такие как проблемы с низкой

производительностью или внешние атаки через Интернет. Кроме того, разработчикам рекомендуется хранить журналы событий, используя службу хранения MWA. В частности, хранилище таблиц обеспечивает лучшее решение для хранения большого количества журналов событий. Хранение журналов событий на виртуальной машине, в которой выполняется роль приложения, нецелесообразно. Поскольку физическое местоположение виртуальной машины может измениться при перезапуске виртуальной машины, разработчик теряет доступ к файлам журнала событий.

В-третьих, разработчикам рекомендуется рассматривать все входные данные, полученные от клиентов, как недействительные или вредоносные. Все данные должны быть проверены до того, как приложение их обработает. Обычные методы проверки входных данных включают проверку длины, типа и формата ввода данных и ограничение диапазона символов, которые можно использовать при отправке данных от клиента.

Наконец, разработчикам рекомендуется избегать хранения конфиденциальных данных, таких как пароли и PAK хранилища. Кроме того, разработчики должны зашифровать все сохраненные данные и все данные, передаваемые через Интернет, используя действующие сертификаты SSL или зашифровывая данные перед передачей. Для повышения безопасности данных, PAK-хранилища, пароли и другие подобные конфиденциальные данные должны периодически обновляться.

На уровне платформы и инфраструктуры MWA внедряет автоматизированные меры по предотвращению и снижению безопасности. Платформа MWA защищена от распространенных интернет-угроз, таких как спуфинг, DoS и прослушивание. Первая мера безопасности, реализованная MWA, - это ограничение связи через все порты, которые явно не включены. Разработчикам рекомендуется включать только те порты, которые необходимы для работы приложения. Эта мера безопасности минимизирует

методы, с помощью которых потенциальные злоумышленники могут получить доступ к данным приложения [21].

Кроме того, спуфинг представляет собой атаки, при которых злоумышленник пытается получить доступ к приложению, предоставляя ложную идентификацию, которая может быть получена путем перехвата сеансов клиента и кражи действительных учетных данных клиента. Платформа MWA имеет несколько брандмауэров, которые фильтруют связь на разных уровнях архитектуры. Помимо защиты от спуфинговых атак, в платформе MWA реализовано несколько механизмов защиты от DoS-атак. Служба автоматической балансировки нагрузки, выполняемая AppFabric, частично смягчает атаки DoS, распределяя входящие клиентские запросы по всем запущенным экземплярам роли приложения. Чтобы дополнить меры безопасности MWA против DoS-атак, разработчикам рекомендуется запускать более одного экземпляра любой роли приложения. Кроме того, автоматизированные службы безопасности MWA блокируют любые сообщения от клиентов, которые определяются как вредоносные.

SDL(Secure Development Lifecycle) - это модель разработки программного обеспечения, разработанная Microsoft. Он направлен на уменьшение количества уязвимостей в программных приложениях, следуя рекомендациям по безопасности и конфиденциальности на протяжении всего процесса разработки. Принятый самими Microsoft, SDL представляет собой семиэтапный процесс, который может быть применен к облачным приложениям, среди других типов приложений. Чтобы обеспечить эффективные результаты, действия SDL должны выполняться вместе, как набор. Кроме того, разработчики должны сосредоточиться на достижении полных и качественных результатов вместо ускорения процесса SDL [20].

Хотя процесс SDL оказывается полезным для некоторых приложений, он может не подходить для любого приложения. Microsoft рекомендует разработчикам использовать SDL в качестве руководства и не гарантирует

надежных результатов. Кроме того, действия SDL предназначены для реализации на любой ОС или платформе разработки. С помощью SDL Microsoft удалось создать методологию разработки с широким применением, которая снижает затраты на разработку, а также повышает безопасность приложений.

ЗАКЛЮЧЕНИЕ

Рост облачных вычислений знаменует собой начало новой эры информационных технологий, которая вступает в модель агрегированных вычислений. Между тем, это приводит к трансформации приложений из локальной в облачную среду, что значительно улучшает нашу жизнь и работу. Более того, облачные вычисления популяризировали таинственные и неуловимые профессиональные концепции, такие как распределенные вычисления, параллельные вычисления, виртуализация и так далее. Тем не менее, по сравнению с другими терминами, название «Облако» кажется абстрактным и иллюзорным, и трудно представить связь между «Облаком» и ИТ. С другой стороны, для публики будет более приемлемым и понятным

описание технологии визуально. Кроме того, эта энергично развивающаяся отрасль будет влиять на другие области в будущем. Например, облачные вычисления будут сильно стимулировать развитие интернет-индустрии, поскольку они являются основой для реализации облачных вычислений, в то время как их интенсивное влияние на аппаратную индустрию можно предсказать, поскольку облачные вычисления обработали сложные вычисления и доставили результаты клиентам.

В связи с постоянным расширением облачные технологии будут использоваться все больше и больше. Между тем, проблема, связанная с проблемами безопасности, является беспрецедентной, что требует постоянного изучения исследователей в области ИТ и информационной безопасности. Следовательно, такие известные организации, как CSA, поддерживающие безопасность облачных вычислений, были созданы для того, чтобы способствовать наилучшему использованию облачных вычислений и обеспечить надежную защиту для них.

В целом, облачные вычисления - это новая модель, основанная на расширении, использовании и взаимодействии Интернета. В диссертации подробно рассмотрены концепция, структура, развертывание, типичное использование и основные проблемы безопасности облачных вычислений. Целью этой диссертации было изучение и исследование принципа облачных вычислений и текущих основных проблем безопасности, а также их изложение простыми словами вместо сложных технических терминов. Будущее облачных вычислений безгранично, и развитие электронной коммерции и услуг невообразимо, а образ жизни и базовые знания облачных вычислений проложат путь к вершине облачных технологий.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Муса Ф. «Типы облачных вычислений», 2014.
2. Клементьев И.П., Устинов В.А. «Введение в облачные вычисления», 2009.
3. Грегг Б. «Производительность систем: Enterprise и Cloud» , 2014.
4. Мурзин Ф.А. «облачные технологии: основные модели, приложения, концепции и тенденции развития» , 2016.
5. Меднов С. «Облачные вычисления» http://www.4cio.ru/pages/index/129.
6. Широкова Е. А. «Облачные технологии» , 2011.
7. Черняк Л. Интеграция - «основа облака. //Открытые системы», 2011.
8. Фингар, Питер Dot.Cloud: «облачные вычисления - бизнес-платформа XXI века» 2013.

9. Миронович В. «Обзор: Облачные вычисления», 2017.
10. РизДжордж. «Облачные вычисления», 2011.
11. Фокин Н.Б., статья «Обоснование эффективности использования Облачных технологий», http://journal.itmane.ru/node/649
12. Беленький А. «Облачные» технологии начинают и выигрывают», 2011.
13. Википедия, статья «Облачные вычисления» http://ru.wikipedia.org/wiki/Облачные_вычисления
14. Монахов Д.Н., Монахов Н.В., Прончев Г.Б., Кузьменков Д.А. «Облачные Технологии. Теория и практика», 2017.
15. Николас Карр « «Великий переход»- что готовит революция облачных технологий», 2014.
16. http://it.sander.su -- статья «Облачные технологии и распределенные вычисления»
17. http://www.bureausolomatina.ru статья «Будущее облачных технологий: европейский взгляд»

18. Кис Блокленд, Йерун Менгеринк, Мартин Пол, «Тестирование облачных сервисов: как тестировать SaaS, PaaS и IaaS», 2017.
19. Сангита Н. «Облачные вычисления и технологии виртуализации в библиотеках», 2017.
20. Томас Эрл, Рикардо Путтини, Зайгам Махмуд, «Облачные вычисления: концепции, технологии и архитектура», 2018.
21. Билл Уильямс, «Экономика облачных вычислений: обзор для лиц, принимающих решения, 2017
22. Техасви Редкар, Тони Гуидичи, «Windows Azure Platform, Apress», 2011
23. Роджер Дженнингс, «Облачные вычисления на платформе Windows Azure», 2016
24. Дэнни Гарбер, Джамал Малик, Адам Фацио, «Windows Azure Hybrid Cloud», 2013,

XÜLASƏ

Magistr dissertasiyası giriş hissə, üç fəsil, nəticələr və təkliflər və istifadə olunmuş ədəbiyyatın siyahısından ibarətdir.

Birinci fəsildə cloud hesablamasının əsas cəhətləri, onun Greed-hesablanması və müəssisələrdə bulud sistemlərinin tətbiqi nəticələri izah edilmişdir.

İkinci fəsildə məlumatların işlənməsi sistemlərinin əsas cloud xidməti modeli müzakirə olunur. Burada üç əsas cloud hesablama modeli ətraflı təhlil edilir.

Üçüncü fəsil əsas cloud hesablayıcı təminatçılarının icmalına həsr edilmişdir. Ən böyük cloud xidməti təmin edənlərindən biri olan Microsoft Azure-yə xüsusi diqqət yetirilir.

SUMMARY

The master's thesis consists of an introductory part, three chapters, conclusions and suggestions, and a list of references.

The first chapter described the main aspects of cloud computing, its comparison with Greed-computing and the results of the introduction of cloud systems in enterprises.

The second chapter discusses the basic cloud service model of data processing systems. Here, three main cloud computing models are analyzed in detail.

The third chapter is devoted to an overview of the main cloud computing providers. Particular attention is paid to Microsoft Azure, one of the largest cloud service providers.