

## **СЕКЦИЯ 3: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ**

### **НОВЫЕ МОДЕЛИ ПРИЛОЖЕНИЙ И ОБЛАЧНЫЕ СЛУЖБЫ ДЛЯ РЕАЛИЗАЦИИ ПРЕИМУЩЕСТВ В БИЗНЕСЕ**

*С.В. Разумников, к.т.н., доцент*

*Юргинский технологический институт (филиал) Национального исследовательского  
Томского политехнического университета  
652055, г. Юрга, Ленинградская 26, тел. 8(38451)77764  
E-mail: demolove7@inbox.ru*

**Аннотация:** Облачные вычисления кардинально изменили способ разработки, тестирования и развертывания приложений. Легкодоступные облачные ресурсы помогают быстро предоставлять новые приложения и обновлять их. Поэтому возникли новые модели поддержки быстрой разработки и развертывания приложений. В статье рассматриваются новые модели приложений и облачные службы для реализации преимуществ в бизнесе.

**Abstract:** Cloud computing has revolutionized the way you develop, test, and deploy applications. Easily accessible cloud resources help you quickly deliver and update new applications. Therefore, new models have emerged to support rapid application development and deployment. This article explores new application models and cloud services to realize business benefits.

**Ключевые слова:** облачные технологии, облачные службы, приложения, преимущества.

**Keywords:** cloud technologies, cloud services, applications, benefits.

Введение

Концептуально перенос приложений в облачную модель «инфраструктура как услуга» (IaaS) прост и обладает рядом преимуществ. Однако можно предоставить принципиально новые преимущества для ИТ и бизнеса, используя возможности, присущие облаку: новые модели приложений и облачные службы. Рассмотрим, как предприятия могут воспользоваться уникальными возможностями облака, чтобы реализовать их преимущества в бизнесе. Начнем с рассмотрения новых архитектур приложений: их сути и преимуществ.

Что означает трансформация

В стартапе, который создает приложение для Интернета вещей (IoT), собирает выходные данные бытового медицинского оборудования через Интернет. На рисунке 1 показана довольно простая архитектура этого приложения.



*Рис. 1. Простая архитектура IoT*

Мы видим, что устройства передают данные по протоколу REST непосредственно на веб-сервер или через маршрутизатор на сервер. В свою очередь сервер обрабатывает их и помещает в базу данных. Затем эти данные отображаются в приложении собственной разработки для создания отчетов.

Зададим несколько простых вопросов:

- Что произойдет, когда компания станет невероятно успешной и ей придется непрерывно поддерживать по Интернету десятки тысяч устройств?
- Как компания обеспечит устойчивость?
- Как она будет выполнять профилактическое или прогнозное обслуживание?
- Как обнаруживать ненадежных поставщиков для своих устройств?
- Как компания сможет быстро добавлять новые отчеты?

Такой компании предложили использовать возможности облака. В предлагаемой архитектуре (рис. 2) стартапу необходимо всего лишь подключить свои устройства к различным службам.

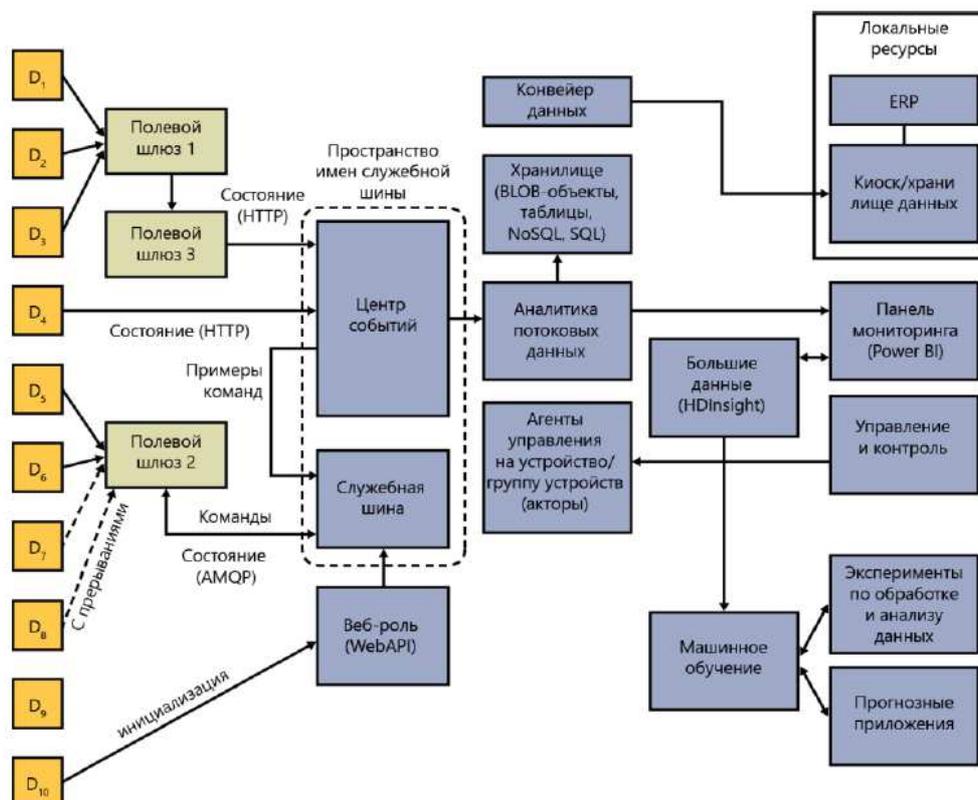


Рис. 2. Многофункциональное комплексное приложение IoT

Используя концентраторы событий Microsoft Azure с высокой пропускной способностью при приеме событий, можно удобно масштабировать приложение стартапа в соответствии с числом устройств. Полученные данные хранятся в таблицах Azure, при этом автоматически создаются две реплики. Это гарантирует защиту от потери данных. Данные можно анализировать в Azure HDInsight с помощью программ, обрабатывающих большие объемы информации в соответствии с моделью MapReduce. Они предоставляют возможность использовать в реальном времени программируемые команды и элементы управления. Можно создавать приложения машинного обучения для прогнозирования скачков продаж или сбоев оборудования, при этом все данные можно просматривать на интуитивно понятной, привлекательной панели мониторинга, и все это – с минимальным кодированием.

Иными словами, довольно ограниченное некогда приложение очень быстро стало инновационным, аналитическим и трансформационным.

Это связано с тем, что благодаря преимуществам облака приложения получают в свое распоряжение множество возможностей для расширения.

Платформа как услуга

Перенос приложений в облачную модель IaaS – это один из подходов и наиболее упрощенный. Конечно, IaaS обладает рядом преимуществ, таких как передача ответственности за центр обработки данных поставщику облачных услуг. Чтобы в действительности выполнить трансформацию в облачную модель, необходимо сделать следующий шаг – разработать приложения специально для облака.

IaaS накладывает определенные ограничения: вы по-прежнему отвечаете за поддержку системного программного обеспечения, операционной системы и базы данных для приложения, а также за периодические исправления и обновления программного обеспечения. Фактически, можно сказать, что IaaS – это только первый шаг к использованию всех возможностей облака.

На рисунке 3 показано, что в моделях «платформа как услуга» (PaaS) вы поддерживаете только свое приложение (синие поля на рисунке), а системное программное обеспечение предоставляет и поддерживает поставщик облачных услуг. Кроме того, решения PaaS обычно обеспечивают удобную масштабируемость и устойчивость с помощью горизонтального масштабирования и репликации данных. Чтобы надежно управлять удостоверениями, PaaS может взаимодействовать с облачными службами, например с Microsoft Azure Active Directory (Azure AD).

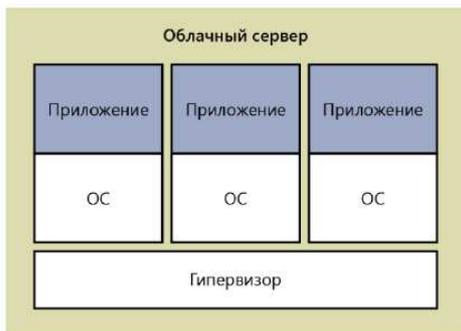


Рис. 3. Модель «платформа как услуга»

Служба веб-приложений Azure App Service, например, дает возможность быстро создать в облаке масштабируемый веб-сайт, приложив минимальные усилия. Microsoft предоставляет базовую веб-инфраструктуру (операционную систему, сетевой стек, хранилище, поддержку языков и функции масштабирования), которая устраняет большую часть временных затрат на управление крупномасштабным веб-приложением. Настройка инструментов масштабирования, резервного копирования и мониторинга для веб-приложений не вызывает затруднений. Она также подключается ко всем остальным службам, которые предоставляет облако для многофункциональных приложений.

Облачные службы Azure – это облачный аналог трехуровневых бизнес-приложений (LoB), использовавшихся десять лет назад. Приложение, относящееся к облачным службам, состоит из трех компонентов: веб-роли (фактически, веб-интерфейс, который масштабируется независимо от остальных частей приложения), рабочей роли (обеспечивает фоновые вычисления и обработку аналогично уровню бизнес-логики в трехуровневой модели) и хранилища (используется входящая в состав Azure база данных SQL Azure, представляющая собой версию SQL Server). Чтобы существующее приложение можно было превратить в облачную службу, необходимо частично его реконструировать: сделать это будет несложно, поскольку намеренно используется модель, аналогичная трехуровневой.

#### Контейнеры и оркестрация

Приложения PaaS предусматривают компромисс: хотя поставщик облачных услуг предоставляет и поддерживает среду операционной системы, во время выполнения ОС и приложение по-прежнему действуют как виртуальная машина, то есть время запуска у приложения такое же, как у всей системы.

Одной из важнейших новых тенденций в архитектуре приложений стала так называемая контейнерная модель, берущая свое название от вездесущих морских контейнеров, которые мы ежедневно видим на судах и полуприцепах. Программные контейнеры похожи на физические – это стандартизированные пакеты программного обеспечения, которые можно легко переносить и быстро развертывать.

Чтобы понять контейнерную модель, необходимо сравнить ее с моделями IaaS и PaaS. В этих моделях гипервизор управляет несколькими экземплярами операционной системы на сервере (в IaaS операционную систему предоставляете вы, в PaaS – поставщик облачных услуг).

На любом сервере найдется несколько очень крупных операционных систем, работающих одновременно в параллельном режиме, в которых полностью дублируется функциональность (каждая операционная система имеет файловый менеджер, подсистему сети и т. д.).

В модели контейнеров (рис. 4) приложения совместно используют один экземпляр операционной системы. В Microsoft Windows и различных дистрибутивах Linux реализована поддержка изоляции, которая гарантирует, что каждое приложение «владеет» операционной системой. Приложения «упакованы» для развертывания в системах, совместимых с контейнерами.

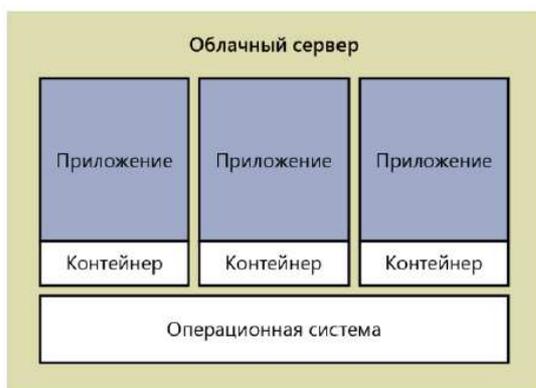


Рис. 4. Архитектура контейнеров

верная логика, поиск, аналитика в реальном времени и т. д. Программное обеспечение для развертывания нужного числа экземпляров каждого контейнера, для управляемого обновления, для обработки ошибок и управления масштабированием называется оркестрацией (рис. 5). Большинство служб оркестрации предоставляет инструменты, позволяющие администраторам создавать правила, например можно запретить размещение на одном сервере контейнеров разных типов или выполнять обработку отказа и восстановления в управляемом последовательном режиме. К хорошо известным продуктам оркестрации относятся Kubernetes, Mesosphere, Docker Swarm и Deis, а также Microsoft Service Fabric.

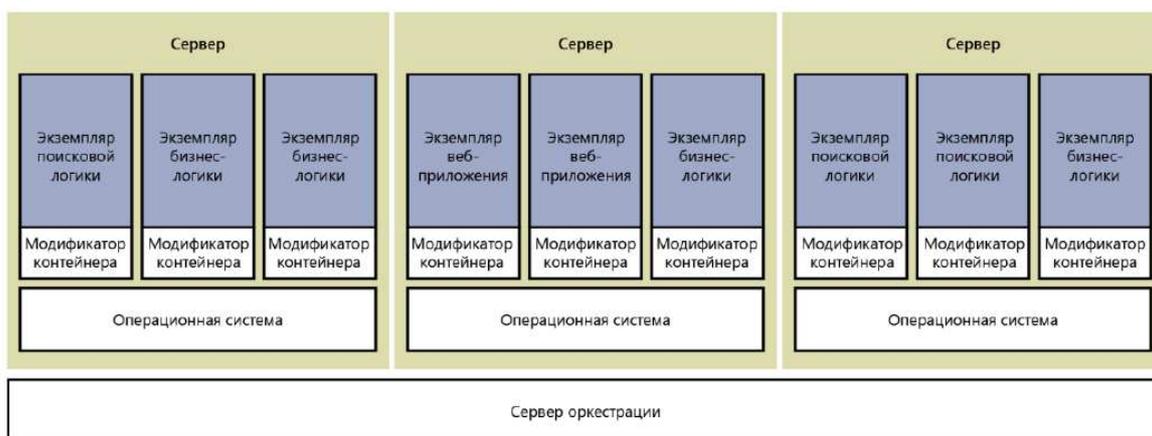


Рис. 5. Оркестрация

Контейнеры – это хороший способ перенести устаревшие приложения в облако и повысить эффективность. Специалисты ИТ-отдела Microsoft перенесли ряд приложений в контейнеры Docker на платформе Azure и получили четыре основных преимущества:

- Более эффективное использование инфраструктуры.
- Стандартизированная конфигурация инфраструктуры.
- Изолированные среды приложений.
- Улучшенные возможности переноса и повторного использования приложений.

По завершении пробного переноса 10 приложений инфраструктура ИТ-отдела Microsoft стала на 400 % компактнее, чем без контейнеров, а общая инфраструктура уменьшилась на 300 %.

#### Микрослужбы

Термин «микрослужбы» появился несколько лет назад. Он охватывает подходы, при которых крупные целостные приложения разбиваются на небольшие компонентные службы. Например, на рисунке 5 мы видим всего три крупномодульных компонента приложения: веб-сервер, бизнес-логика и база данных.

Однако если мы создаем приложение для электронной коммерции, возникнет ряд вопросов.

Возможно, для более эффективного использования ресурсов лучше создать группу для разработки компонента «Каталог», другую – для разработки компонента «Заказ», и еще одну – для внедрения функции поиска (платной или с открытым исходным кодом) стороннего производителя. Затем эти компоненты можно разрабатывать и обновлять независимо друг от друга.

Микрослужбы – это больше архитектурный принцип (или шаблон проектирования), чем реальная технология. Вы можете создавать микрослужбы в IaaS, PaaS или с помощью контейнеров.

#### Модель акторов

В инструментарии разработчика облачных решений есть еще один инструмент под названием модель акторов (рис. 6). Актор – это простой, обычно небольшой объект в облаке, который обладает уникальной идентичностью, может взаимодействовать с другими акторами и поддерживает состояние. Как правило, акторы представляют физические объекты, например людей или устройства. Для такого рода технологий недавно появился новый термин – цифровой двойник: объект актора, по существу, отражает в цифровом виде то, что происходит в реальном мире. Платформа акторов абстрагирует инфраструктурные концепции, такие как серверы. Это означает, что акторы могут взаимодействовать друг с другом, не зная, на каких физических серверах они находятся.

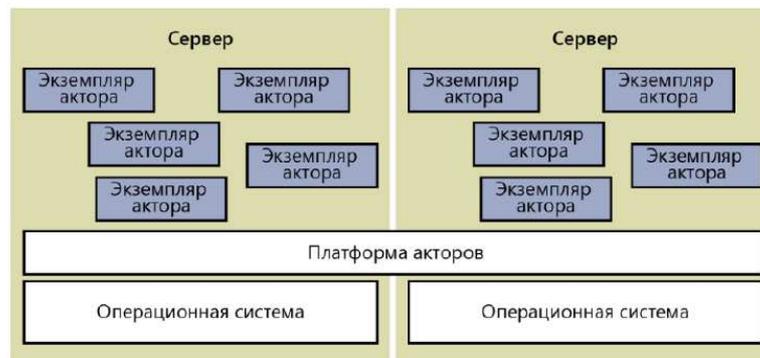


Рис. 6. Модель акторов

Пример использования – сетевые игры, в которых каждый экземпляр актора представляет игрока и содержит состояние, например, текущий счет игрока, расположение в игре и список других участвующих игроков.

Акторов также можно использовать для управления большим числом устройств IoT. Они особенно эффективны, когда есть много похожих устройств, например датчиков контроля. Каждый экземпляр актора может получать обновления состояния от конкретного датчика, например датчика давления, и автоматически уведомлять актора, контролирующего клапан или другое устройство, что необходимо принять соответствующие меры.

#### Устойчивость в облаке

Важно отметить, что в облаке цены остаются низкими из-за использования недорогого стандартного оборудования. В прошлом предприятия приобретали высокопроизводительные компьютеры для вертикального масштабирования, однако в облаке производительность достигается за счет горизонтального масштабирования. Поэтому вместо покупки мощного суперкомпьютера вы выполняете масштабирование с помощью множества компьютеров в распределенной системе, а это означает, что необходимо подготовиться к случайному сбою.

Рассмотрим требуемое соглашение об уровне обслуживания (SLA). В таблице 1 показан потенциальный совокупный простой для различных уровней SLA.

Таблица 1

Соглашения об уровне обслуживания			
SLA	Время простоя в неделю	Время простоя в месяц	Время простоя в год
99%	1,68 ч.	7,2 ч.	3,65 дня
99,9%	10,1 мин.	43,2 мин.	8,76 ч.
99,95%	5 мин.	21,6 мин.	4,38 ч.
99,99%	1,01 мин.	4,32 мин.	52,56 мин.
99,999%	6 с.	25,9 с.	5,26 мин.

Необходимость в мероприятиях по увеличению числа «девяток» после запятой зависит от бизнес-сценария.

Заключение. Перенос приложений в облако – важная и серьезная задача, требующая изменения способа работы предприятия и ИТ-инфраструктуры. В данной статье были рассмотрены новые архитектуры приложений: их суть и преимущества. Показано, как предприятия могут воспользоваться уникальными возможностями облака, чтобы реализовать их преимущества в бизнесе.

*Работа выполнена при финансовой поддержке стипендии Президента РФ.*

Список используемых источников:

1. Гребнев Е. Облачные сервисы. Взгляд из России – М.: CNews, 2011. – 282.
2. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM // Доклады Томского государственного университета систем управления и радиоэлектроники. 2020. Т. 23. № 1. С. 53-61.
3. Razumnikov S.V., Kurmanbay A.K. Models of evaluating efficiency and risks on integration of cloud-base IT-services of the machine-building enterprise: a system approach // IOP Conference Series: Materials Science and Engineering. - 2016 - Vol. 124 - №. 1, Article number 012089. - p. 1-5.
4. Разумников С.В. Интегральная модель оценки результативности внедрения облачных ИТ-сервисов // Научные труды Вольного экономического общества России. - 2016 - Т. 201. № 4. - С. 492-504.
5. Razumnikov S., Prankevich D. Integrated model to assess cloud deployment effectiveness when developing an it-strategy // IOP Conference Series: Materials Science and Engineering. – 2016. – Vol. 127: Urgent Problems of Modern Mechanical Engineering.
6. Разумников С.В. Некомпенсаторное агрегирование и рейтингование провайдеров облачных услуг // Доклады Томского государственного университета систем управления и радиоэлектроники. 2018. Т. 21. № 4. С. 63-69.

#### **ПОСТАНОВКА ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЙ НА ЭТАПЕ СТРАТЕГИЧЕСКОГО КОНТРОЛЯ ГИБРИДНОЙ МОДЕЛИ СТРАТЕГИЧЕСКОГО УПРАВЛЕНИЯ ПАССАЖИРСКИМ АВТОТРАНСПОРТНЫМ ПРЕДПРИЯТИЕМ**

*О.А. Колегова, специалист по УМР каф. ИС*

*«Юргинский технологический институт (филиал) Национального исследовательского Томского политехнического университета»,*

*652050, г. Юрга, ул. Ленинградская, 26, тел. (38451) 7-77-64*

*E-mail: Olga030188@mail.ru*

**Аннотация:** В статье представлена математическая модель задачи принятия решений для этапа стратегического контроля гибридной модели стратегического управления пассажирским автотранспортным предприятием. Отличительной особенностью гибридной модели является возможность формализации и обработки экспертных оценок на всех этапах стратегического управления: разработки стратегических целей, отбора проектов развития и контроля выполнения стратегии.

**Abstract:** The article presents a mathematical model of the decision-making problem for the stage of strategic control of the hybrid model of strategic management of a passenger transport enterprise. A distinctive feature of the hybrid model is the ability to formalize and process expert assessments at all stages of strategic management: developing strategic goals, selecting development projects, and monitoring the implementation of the strategy.

**Ключевые слова:** модель задачи принятия решений, стратегия, нечеткие множества, контроль реализации стратегии, пассажирское автотранспортное предприятие.

**Keywords:** model of the decision-making problem, strategy, fuzzy sets, strategy implementation control, passenger transport company.

Введение

В работе [1] описан интегрированный подход к разработке гибридной модели стратегического управления пассажирским автотранспортным предприятием. Особенность данного подхода заключается в том, что его использование в гибридной модели позволяет реализовать три базовых этапа стратегического управления (целеполагания, стратегического выбора, контроля выполнения стратегии) одновременно, путем комбинирования нескольких методов на разных этапах и установления