

Применение облачных сервисов Amazon для разработки распределённых web-приложений

Слисенко К.Ю.; Сиротко С.И.;

Кафедра электронных вычислительных машин
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
e-mail: kslisenko@bsuir.by

Аннотация—В работе рассматриваются облачные сервисы, показываются и обсуждаются их особенности, которые должны учитываться при проектировании приложений для работы в облаке. Ставится задача проектирования распределённой системы, приводится описание её архитектуры. Также приводится описание облачных сервисов Elastic Cloud 2, Simple Storage Service, Simple Queue Service компании Amazon. Данные сервисы были использованы в проектируемой системе.

Ключевые слова: облачные сервисы, облачная инфраструктура, web-приложения, распределённые вычисления, Amazon

I. ВВЕДЕНИЕ

Облачные сервисы дают много привлекательных возможностей, которые недоступны при использовании традиционных вычислительных инфраструктур. Для эффективного использования их преимуществ необходимо учитывать их особенности при проектировании приложений. Целью этой работы является выработка шаблонов, методик и лучших практик проектирования архитектур приложений, работающих в облаке, и использования облаков в целом.

В ходе исследования мы анализируем существующие теоретические материалы и наработки в этой области, проектируем архитектуру распределённого web-приложения и реализуем его.

II. КРАТКАЯ ХАРАКТЕРИСТИКА РАЗРАБАТЫВАЕМОГО ПРИЛОЖЕНИЯ

Разрабатываемая система позволит людям, не имеющим достаточных знаний программирования и навыков android-разработки, создавать android-приложения. Через web-интерфейс можно будет выбрать один из готовых шаблонов, изменить дизайн, заполнить данные. Далее происходит сборка приложения, и оно может быть установлено на мобильное устройство. Требования к системе и описание архитектуры будут перечислены ниже.

III. ОБЩАЯ ИНФОРМАЦИЯ ПРО ОБЛАЧНЫЕ СЕРВИСЫ

На текущий момент самыми распространёнными видами облаков являются: ПО как сервис (software as a service - SAAS), платформа как сервис (platform as a service - PAAS), инфраструктура как сервис (infrastructure as a service - IAAS).

Отличия облачных инфраструктур от традиционных можно условно разбить по двум критериям. Для бизнеса это практически нулевые вложения в инфраструктуру на старте проекта,

возможность в любой момент получить практически неограниченные дополнительные ресурсы или освободить неиспользуемые. С технической стороны появляются возможности горизонтального и вертикального масштабирования. Также стало возможным переносить вычислительные ресурсы ближе к данным и пользователям.

Можно выделить следующие особенности, влияющие на разработку приложений для работы в облачной среде:

1) Особые условия организации защиты и контроля доступа.

2) Большие возможности распараллеливания работы компонентов системы.

3) Множество типов хранилищ данных для различных нужд, их специфика.

4) Свойство “эластичности” инфраструктуры: в любой момент нам доступно практически неограниченное количество ресурсов. При необходимости в любой момент можно использовать дополнительные ресурсы, либо освободить неиспользуемые.

5) Декомпозиция монолитных частей приложения и больших задач, взаимодействие таких компонентов через асинхронные очереди сообщений.

6) Особенности устойчивости приложений к сбоям. Автоматическое восстановление в случае сбоев, либо запуск копии приложения в другом дата-центре.

На сегодняшний день самыми крупными провайдерами являются: Amazon Web Services, Google App Engine, Windows Azure, Force.com, Rackspace, GoGrid. Отдельно можно сказать о сервисе RightScale, который предоставляет “обёртку” над перечисленными провайдерами и позволяет переносить между ними виртуальные машины.

IV. КРАТКАЯ ХАРАКТЕРИСТИКА ПРОВАЙДЕРА ОБЛАЧНЫХ СЕРВИСОВ AMAZON

Для реализации приложения был выбран провайдер облачных сервисов Amazon - крупнейший на сегодняшний день игрок на рынке. Amazon является IAAS-провайдером, что позволяет максимально гибко настроить инфраструктуру.

Следующие сервисы Amazon мы использовали для разработки приложения:

1) Amazon EC2 (elastic cloud) предоставляет в аренду виртуальные машины разной производительности и, соответственно, стоимости. Виртуальные машины можно запускать как с “чистой” операционной системой, так и с предустановленным ПО, причем сервис позволяет

сохранять настроенные конфигурации виртуальных машин в образы, называемые AMI (Amazon Machine Image), из которых их затем можно запустить в любой момент. Также сервис позволяет запускать виртуальные машины в разных дата-центрах, за счёт чего увеличивается надёжность системы в целом - сбои в одном дата-центре не влияют на работу виртуальных машин в другом.

2) Amazon S3 (simple storage service) предоставляет файловое хранилище с неограниченной вместимостью, высокой надёжностью и высокой скоростью чтения. Максимальный размер сохраняемых файлов составляет 5 Тб. В нашей архитектуре в такое хранилище помещаются собранные android-приложения.

3) Amazon SQS (simple queue service) предоставляет асинхронную "очередь сообщений". В нашей системе в очередь помещаются задачи на сборку android-приложений.

Компанией Amazon предусмотрен ряд архитектур, рекомендованных для приложений различного назначения, например: архитектура распределённого web-приложения, архитектура приложения, предназначенного для пакетной обработки данных, архитектура приложения с поддержкой различных способов восстановления в случае сбоев. Первые два примера были взяты нами за основу при проектировании архитектуры нашего приложения. Подробное описание данных и других архитектур можно найти на сайте компании.

V. АРХИТЕКТУРА РАЗРАБАТЫВАЕМОГО WEB-ПРИЛОЖЕНИЯ

Разрабатываемая система должна соответствовать следующим требованиям:

1) Возможность одновременной работы большого числа пользователей, устойчивость к большим нагрузкам.

2) Возможность выполнять большое количество длительных задач по компиляции и сборке android-приложений (не в реальном времени).

3) Возможность масштабирования инфраструктуры относительно нагрузок и оптимальное использование ресурсов: должна быть использована только необходимая в каждый конкретный момент инфраструктура, не должно быть простаивающих ресурсов либо нехватки ресурсов.

Основными компонентами приложения являются:

4) Кластер web-серверов. На данных серверах развёрнут web-интерфейс, с помощью которого пользователи могут создавать android-приложения.

Для распределения нагрузки между серверами кластера используется сервис Amazon Elastic Load Balancing. Сервера объединены в Amazon Auto Scaling Group для автоматического масштабирования относительно нагрузок.

5) Кластер NoSQL Mongo DB серверов БД. Пользовательские данные располагаются на нескольких серверах. Существует центральный сервер, который обладает информацией о том, на каком из остальных серверов хранится запрашиваемая информация.

6) Очередь сообщений, содержащая задания для сборки android-приложений. После выбора пользователем команды "собрать приложение" в очередь помещается задание на сборку.

7) Кластер серверов-сборщиков android-приложений. Каждый сервер берёт первое поступившее задание на сборку из очереди, выполняет её и помещает результат в хранилище Amazon S3.

8) Хранилище файлов Amazon S3 в сочетании с сервисом доставки контента Amazon Cloud Front. Данные сервисы используются для хранения собранных android-приложений.

9) Сервер контроля доступа служит для авторизации и проверки прав доступа пользователей системы.

10) Приложение для администратора предоставляет пользовательский интерфейс для конфигурирования используемых системой ресурсов. Оно позволяет вручную запускать новые сервера, останавливать существующие

- [1] Документация Amazon [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://aws.amazon.com/documentation>
- [2] AWS Architecture center [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://aws.amazon.com/architecture/>
- [3] Matt Tavis, "Web Application Hosting in the AWS Cloud", 2010, с. 1-12. – Электронные данные. – Режим доступа: http://media.amazonwebservices.com/AWS_Web_Hosting_Best_Practices.pdf
- [4] Jinesh Varia, "Cloud Architectures", 2008, с 1-12. – Электронные данные. – Режим доступа: http://media.amazonwebservices.com/AWS_Cloud_Architectures.pdf
- [5] Jinesh Varia, "Architecting for the Cloud: Best practices", 2010, с 1-23
- [6] George Reese, "Cloud Application Architectures. Building Applications and Infrastructure in the Cloud", 2009, O'Reilly Media, с 1-208