

АРХИТЕКТУРА РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ОБРАБОТКИ ВИДЕО

Пац Е.А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Насуро Е.В. – к.т.н., доцент каф. ЭВМ

Архитектура распределенной системы обработки видео на основе микросервисов и облачных технологий. Основными преимуществами системы являются: высокая доступность, автоматическая масштабируемость, отказоустойчивость.

В современном мире большую популярность получают облачные технологии. Применение облачных технологий значительно снижают затраты на инфраструктуру, уменьшают время доставки кода и увеличивают доступность сервисов для конечных пользователей. Существуют различные виды предоставления облачных услуг. Каждый уровень имеет свою зону ответственности и модель обслуживания как для провайдера, так и для конечного пользователя. Ключевыми критериями являются время доступности и SLA.

К основным видам предоставления услуг относятся: программное обеспечение как сервис (Software as a service, SaaS), платформа как сервис (Platform as a service, PaaS), инфраструктура как сервис (Infrastructure as a service, IaaS). Все данные уровни разграничивают ответственность между провайдером облачных услуг и заказчиком. Разграничение ответственности подробно изображено на рисунке 1.



Рисунок 1 – Принцип разграничения ответственности облачных услуг

Для построения распределенной системы достаточно использовать модель предоставления IaaS, так как на данном уровне облачные провайдеры берут на себя ответственность за виртуализацию. Это означает, что в случае отказа физического сервера виртуальные машины запущенные на нем будут мигрированы на исправные сервера с минимальным временем недоступности. Таким образом система не зависит от аппаратного обеспечения и все инциденты решаются на стороне провайдера. Так же, при таком подходе, отсутствует зависимость от определенного облачного провайдера, таких как Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, Alibaba Cloud. Существует вариант использования приватного облака на основе OpenStack. Появляется возможность изменить облачного провайдера без изменения программного кода и архитектуры платформы.

Микросервисная архитектура подразумевает собой развертывание приложений в контейнерах. Существует множество платформ, осуществляющих управление и оркестрацию. К таким платформам относятся OpenStack (RedHat), Kubernetes (Google), Mesos (Apache), Docker Swarm. Для решения поставленной задачи идеально подходит платформа Kubernetes. Данная платформа разработана на языке программирования, что позволяет его модифицировать. Так же огромное количество крупных проектов использует данную платформу. Kubernetes позволяет с легкостью масштабировать кластер используя API облачного провайдера, что позволяет экономить на ресурсах. Особую роль в платформе играет механизм репликации, это повышает доступность конечных сервисов и позволяет распределять нагрузку. К преимуществам Kubernetes можно отнести то, что данная платформа не диктует выбор определенного типа приложений или фреймворка, кроме этого, для разработки приложений можно использовать любой язык программирования. При необходимости развертывания баз данных, возможно использование как SQL (PostgreSQL, MySQL, Oracle), так и NoSQL (MongoDB, Cassandra). Kubernetes предоставляет абстракции для кластеризации и репликации баз данных, возможность и методы бекапирования. Использование данной платформы позволяет проводить обновления сервисов и приложений без времени недоступности.

Основная функция Kubernetes заключается в оркестрации контейнеров что понимается, как планирование работы контейнеров разной степени загруженности на физических и виртуальных устройствах. Так же платформа контролирует ограничения, налагаемые средой развертывания и конфигурацией кластера. Kubernetes имеет механизмы слежения за всеми запущенными контейнерами, по средствам которых проводится анализ состояния приложений.

Использование контейнеров произвело настоящий прорыв в парадигме разработки и выполнения крупных, сложных программных систем. По сравнению с традиционными моделями они имеют такие преимущества. Во-первых, использование микросервисной архитектуры уменьшает время разработки приложений, так же уменьшается время их развертывания, что позволяет прибегнуть к

методологиям непрерывной разработки, интеграции и развертывания. Разграничивается ответственность между разработчиком и администратором, так как поставка производится в виде образов контейнеров. Образ представляет собой шаблон из набора слоев, используемый для запуска контейнера.

Запуск контейнеров происходит в определенной среде выполнения контейнеров, что позволяет с легкостью воссоздавать окружения для разработки, тестирования и промышленного использования. Так же, отсутствует привязка к аппаратному обеспечению и операционной системе вычислительной машины, на которой будет запущен контейнер. Принцип работы среды запуска контейнеров подробно изображен на рисунке 2.



Рисунок 2 – Принцип работы среды запуска контейнеров

В качестве среды выполнения контейнеров был выбран Docker. На текущий момент данное решение является самым востребованным. Docker имеет множество функций, которые отсутствуют у конкурентов. Так же имеется площадка DockerHub на которой собрано большое количество образов позволяющих решать различные задачи. Для построения системы приложение необходимо упаковать в контейнер, предварительно создав образ. Для разработки образа необходимо использовать Dockerfile.

Таким образом, разработанная архитектура соответствует основным параметрам, каждый компонент системы отказоустойчив и имеет возможность горизонтального масштабирования. В разработанной архитектуре применены самые актуальные облачные решения.

Список использованных источников:

1. Sean P. Kane, Karl M. Docker: Up & Running: Shipping Reliable Containers in Production – O'Reilly Media, 2015 – 201 с.
2. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга – Санкт-Петербург : Издательский дом «Питер», 2020. – 544 с.
3. Сайфан Джиджи Осваиваем Kubernetes. Оркестрация контейнерных архитектур. – СПб.: Питер, 2019. – 400 с.
4. Ньюмен С. Создание микросервисов – Санкт-Петербург : Издательский дом «Питер», 2016. – 304 с.