

Архитектура программных
систем. Лекция 3
Сервис-ориентированная архитектура

Михаил Пожидаев

12 марта 2024 г.

Сервисы

Сервис-ориентированная архитектура

Давайте разделим!

Сервер разделяется на множество сервисом меньшего размера, каждый из которых отвечает за некоторую подзадачу.

1. Ключевую роль играет качество и фиксированность интерфейсов.
2. Разработка сервисов ведётся различными коллективами, что упрощает взаимодействие.

Технологии RPC

Remote Procedure Call

Центральную роль в сервис-ориентированной архитектуре играет механизм, позволяющий производить удалённый вызов процедур. Примерами подобных технологий могут быть:

- ▶ SOAP (Simple Object Access Protocol);
- ▶ CORBA (Common Object Request Broker Architecture);
- ▶ XMLRPC.

Микросервисы

Сервисы минимального размера

Нельзя!

Запретим иерархические связи между сервисами.
Сервисы должны быть одноранговыми.

Можно!

Допускается любое взаимодействие между сервисами, включая прямое общение по любым протоколам. Хотя приветствуется использование брокера.

Контейнерная виртуализация

Основные черты контейнерной виртуализации

1. Код выполняется нативно.
2. В гостевых системах ядро общее с хост-системой.
3. Каждый контейнер имеет свой IP-адрес.
4. Лимитируется память, процессорное время и пр.

Docker

От компании Google

Docker — система управления контейнерами на базе пространств имён и контрольных групп Linux.

1. Автоматически конфигурирует сеть для гостевых систем и хост-системы.
2. Использует каскадное объединение томов хранения файлов.
3. Поддерживает публичный репозиторий образов контейнеров.
4. Написана на языке Go.

Прямо на ходу

Задача «горячего» обновления

Требования к решению задачи «горячих» обновлений (rolling updates):

1. Контейнеры, не являющиеся контроллером и брокером, периодически получают обновления. Чем чаще, тем лучше.
2. Работу сервиса останавливать нельзя, пользователи должны наблюдать непрерывное функционирование с нормальной обработкой запросов.
3. Близкой является задача динамического изменения набора контейнеров-работников для увеличения или уменьшения задействованных вычислительных мощностей.

Порядок обновления

Действия при «горячем» обновлении

1. Механизм реализуется в непосредственной связи с используемым брокером, выполняющим обслуживание межконтейнерного взаимодействия. Именно он чаще всего хранит информацию о конфигурации системы.
2. Инструмент оркестрации получает команду изменить конфигурацию системы, определяет набор контейнеров, которые должны быть остановлены.
3. Контейнеры получают сигнал завершить свою работу (SIGTERM), после чего уведомляют брокер о своём новом статусе.
4. Контейнеры завершают выполнение тех задач, которые были в работе на момент получения сигнала завершить работу.
5. Контейнеры либо добровольно завершают свою работу, либо их убивают по timeout при помощи SIGKILL.

Оркестрация контейнеров

Когда контейнеров становится много, то управление ими требует применение дополнительных инструментов.

Существует ряд утилит, которые решают задачу так называемой оркестрации контейнеров: запуск необходимого количества контейнеров и поддержания их работоспособности. Среди них наиболее популярны:

- ▶ Kubernetes;
- ▶ Docker Swarm.

Брокер

Его функции и назначение

В микросервисной архитектуре брокер используется для:

1. Гарантированной доставки сообщений между микросервисами.
2. Балансирования нагрузки с толерантностью отказов собственных реплик.
3. Унификации представления структур для работы на разных языках и платформах.

Примеры

Наиболее известные реализации

- ▶ *Apache Kafka*: написанный на Java и Scala наиболее распространённый брокер сообщений проекта Apache.
- ▶ *Rabbit MQ*: написанный на Erlang брокер с расширенными функциями маршрутизации сообщений.
- ▶ *centrifugo*: написанный на Go брокер сообщений с российскими корнями.

Спасибо за внимание!

Всё о курсе: <https://marigostra.ru/materials/Architecture.html>

E-mail: msp@luwrain.org

Канал в Телеграм: @MarigostraRu