

Архитектор программного обеспечения

Best Practice по разработке архитектуры программного обеспечения

Длительность курса: 108 академических часов

1 Centralized Systems. Single node patterns

1 Когда нужен монолит, а когда микросервисы и компоненты

Домашние задания

1 Базовый проект

сделать простой проект на выбор:

- 1) чат платформу
- 2) интернет магазин
- 3) что-то свое по согласованию с преподавателем

Для базовой реализации достаточно регистрации, авторизации, базовых действий через CRUD

2 **Sidecar,
Ambassador,
Adapter
паттерны.
Принципы
работы.**

Домашние задания

1 Уведомления

Необходимо создать 2 сервиса для рассылки уведомлений (email и push, например). Сделать над ними универсальный интерфейс

2 Centralized Systems. Multi node patterns (microservices)

1 Паттерны и антипаттерны декомпозиции микросервисов.

Домашние задания

1 Микросервисы

Необходимо поделить ваше монолитное приложение на группу микросервисов

2 Sharding. Принцип работы. Реализация паттерна на примере распределенного поиска

Домашние задания

1 Умный поиск

Необходимо придумать систему умного (семантического) поиска (например поиск товаров в интернет магазине). Систему поиска сделать распределенной и данные побить по некоторому критерию

3 Принципы организации сервисов. Типы межсервисного взаимодействия.

4 введение в docker. Обзор docker compose

5 Масштабирование stateless и stateful сервисов. Паттерны кэширования в микросервисной архитектуре. Идемпотентность и коммутативность.

Домашние задания

1 Кэширование

Нужно определить, какие данные, в вашей системе являются перманентными / или редко меняются, и которые часто запрашиваются. Необходимо для этих данных использовать один из паттернов кэширования: хранение кэша в выделенном сервисе (например, redis), кэширование внутри приложения, распределенный кэш и т.д.
Для того, чтобы дз было принято должен быть использован хотя бы один из паттернов в вашем проекте.

6 Вопросы отказоустойчивости информационной системы. Отложенные сообщения (очереди). Реализация системы уведомлений пользователей, состоящей из нескольких сервисов

Домашние задания

1 Асинхронное взаимодействие

Нужно определить, какие сервисы в вашем приложении, должны работать асинхронно (например, отправка уведомлений). Далее необходимо выбрать очередь сообщений и перевести данные сервисы на асинхронное взаимодействие

7 **подходы к организации инфраструктуры: мониторинг, профилирование, взаимодействие между сервисами**

Домашние задания

1 Мониторинг

Необходимо добавить систему мониторинга для вашего приложения на основе стека grafana + prometheus

8 **Паттерны декомпозиции данных в микросервисах**

топики:

- 1) Паттерны синхронизации данных между сервисами
 - 2) CQRS.
 - 3) Event Sourcing
-

9 **serverless**

- 1) что такое serverless вычисления
- 2) типы serverless сервисов
- 3) обзор lambda функций на примере AWS lambda functions и GCP cloud functions
- 4) паттерны проектирования на lambda функциях

3 Decentralized Systems

1 **Общий концепт.
Решаемые
задачи.
Примеры**

2 **CAP теорема.
Общие
принципы и
описание. BASE
и ACID**

3 **CP системы.
RAFT, PAXOS,
ZAB**

Домашние задания

- 1 Используя любой CP алгоритм, необходимо реализовать свою базу данных (например K-V).
-

4 **PA системы.
GOSSIP,
CasPAXOS.
CRDT структуры**

Домашние задания

- 1 Используя любой PA алгоритм, необходимо реализовать механизм обмена сообщениями (на подобии чата).
-

5 **Практический семинар: реализация своей распределенной sql / nosql базы данных. Реализация своего оркестратора для управления задачами**

6 **Основы блокчейн: bitcoin**

7 **Основы блокчейн: ethereum**

8 **Практический семинар: создание своего приватного блокчейн на основе EVM**

Домашние задания

1 Кripto-платежи

Нужно создать сервис для возможности покупки товара / подписки через криптовалюту

9 **основы работы DAG**

- 1 **Оценка стоимости спроектированной системы.
Определение рисков на разных стадиях жизненного цикла:
проектирование,
разработка,
внедрение,
поддержка и эксплуатация,
утилизация**

1 Вводный урок по итоговому проекту

Домашние задания

1 Итоговый проект

Цель: В завершении курса студентам будет предложено выполнить проектную работу. Задание включает в себя разработку распределенной / децентрализованной, отказоустойчивой системы.

1) Консультация по проекту. Студентам будет предложено одно из заданий на выбор (либо свое):

1. распределенная база данных.
2. распределенный data-lake.
3. реализация приватного блокчейн
4. распределенная система семантического поиска

Необходимо выбрать и обсудить спецификации проекта.

2 консультация по проекту

консультация по проекту. Промежуточный скрининг проектов.

Домашние задания

- 1 Необходимо предоставить первый прототип системы (MVP)
-

3 **Итоговый урок**

защита проекта. Защита архитектуры и реализации проекта