

# Содержание

1. Архитектура информационных систем .....	3
1.1. Модуль 1: Архитектурные основы .....	3
1.1.1. Лекция 1: Введение. Принципы построения приложения .....	3
1.1.1.1. Оптимизация системы требует выбора критерия оптимизации .....	3
1.1.1.2. Принципы SOLID .....	3
1.1.1.3. Dependency Injection .....	3
1.1.1.4. Clean Architecture .....	3
1.1.2. Лекция 2: Альтернативные архитектурные паттерны .....	3
1.1.2.1. Архитектура на основе фреймворка (Rails, Django) .....	3
1.1.2.2. Модель акторов (Erlang/Elixir/Akka) .....	3
1.1.2.3. Проектирование, ориентированное на данные (Data-oriented design) .....	3
1.1.3. Лекция 3: Протоколы взаимодействия .....	3
1.1.3.1. REST: стандартный веб-протокол .....	3
1.1.3.2. GraphQL: клиент запрашивает только нужные данные .....	3
1.1.3.3. gRPC: бинарный протокол, высокая производительность .....	3
1.1.3.4. WebSocket: двусторонний канал .....	3
1.2. Модуль 2: Фундаментальные структуры хранилищ данных .....	3
1.2.1. Лекция 4: Базовые механизмы хранения данных .....	3
1.2.1.1. CSV-файлы как минимальная форма персистентности .....	3
1.2.1.2. CSV с добавлением записей и журналированием изменений .....	4
1.2.1.3. Хеш-индексы как первый уровень ускорения выборки .....	4
1.2.2. Лекция 5: Структурированные индексы .....	4
1.2.2.1. Отсортированная таблица строк (Sorted String Table, SSTable) .....	4
1.2.2.2. Дерево слияния структурированного журнала (Log-Structured Merge-Tree, LSM-Tree) ...	4
1.2.2.3. Когда использовать LSM-деревья и B-деревья (LSM хороша для записи, B-Tree для чтения)	
1.2.3. Лекция 6: ACID, уровни изоляции, аналитические и транзакционные хранилища .....	4
1.2.3.1. ACID (Atomicity, Consistency, Isolation, Durability) .....	4
1.2.3.2. Уровни изоляции .....	4
1.2.3.3. Аналитические базы данных (OLAP) и колоночные хранилища .....	4
1.2.3.4. Сжатие колонок .....	4
1.2.3.5. Сравнение подходов .....	4
1.3. Модуль 3: Другие хранилища данных .....	4
1.3.1. Лекция 7: Аналитика в оперативной памяти .....	4
1.3.1.1. Vectorized execution (блочная обработка) .....	4
1.3.1.2. DuckDB .....	4
1.3.2. Лекция 8: Кэш в оперативной памяти .....	4
1.3.2.1. Skip List .....	4
1.3.2.2. Incremental Rehashing .....	4
1.3.2.3. HyperLogLog .....	4
1.3.2.4. Redis .....	4
1.4. Модуль 4: Методы поиска похожих/близких элементов .....	4
1.4.1. Лекция 9: Специализированные индексы для поиска .....	4
1.4.1.1. Inverted Index (перевёрнутые индексы) .....	4
1.4.1.2. Trie (префиксные деревья) .....	4
1.4.1.3. Bitmap indexes .....	4
1.4.2. Лекция 10: Векторные индексы для поиска подобия .....	5
1.4.2.1. Основы векторных представлений .....	5
1.4.2.2. HNSW (Hierarchical Navigable Small World) .....	5
1.4.2.3. LSH (Locality-Sensitive Hashing) .....	5
1.4.2.4. Примеры: Weaviate, Pinecone, Qdrant .....	5

1.4.3. Лекция 11: Геопространственные структуры .....	5
1.4.3.1. R-Tree .....	5
1.4.3.2. QuadTree и KD-Tree .....	5
1.5. Модуль 5: Операционные компоненты .....	5
1.5.1. Лекция 12: Прокси и Rate Limiting .....	5
1.5.1.1. Forward Proxy / Reverse Proxy .....	5
1.5.1.2. Распределение нагрузки .....	5
1.5.1.3. Ограничение частоты запросов .....	5
1.5.2. Лекция 13: Безопасность и управление доступом .....	5
1.5.2.1. Аутентификация и авторизация .....	5
1.5.2.2. RBAC и ABAC .....	5
1.5.2.3. Single Sign-On .....	5
1.5.3. Лекция 14: Логирование, мониторинг и планирование задач .....	5
1.5.3.1. Сбор логов .....	5
1.5.3.2. Сбор метрик (Prometheus) .....	5
1.5.3.3. Мониторинг и диагностика (Grafana) .....	5
1.5.3.4. Планировщики (Celery, Airflow, Dagster) .....	5
1.6. Модуль 6: System Design .....	5
1.6.1. Лекция 15: Разбор дизайна конкретной системы .....	5
1.6.1.1. Применение всех концепций на практике .....	6
2. Распределённые системы обработки данных .....	7
2.1. Модуль 1: Компоненты распределённых систем .....	7
2.1.1. Лекция 1: Основы распределённых систем .....	7
2.1.1.1. CAP-теорема .....	7
2.1.1.2. Механизмы репликации данных .....	7
2.1.1.3. ZooKeeper: распределённое решение проблем согласованности и отказоустойчивости .....	7
2.1.2. Лекция 2: Гарантии консистентности и координация распределённых транзакций .....	7
2.1.2.1. Алгоритмы достижения консенсуса .....	7
2.1.2.2. Модель конечной консистентности (Eventual consistency) и требование идемпотентности операций .....	7
2.1.2.3. Протокол двухфазного коммита (Two-Phase Commit, 2PC) .....	7
2.2. Модуль 2: Распределённые сервисы .....	7
2.2.1. Лекция 3: Асинхронное взаимодействие сервисов .....	7
2.2.1.1. Event Bus и Event-Driven Architecture .....	7
2.2.1.2. Publish-Subscribe vs Message Queue .....	7
2.2.1.3. Saga Pattern .....	7
2.2.2. Лекция 4: Kafka .....	7
2.2.2.1. Внутреннее устройство .....	7
2.2.3. Лекция 5: Kubernetes .....	7
2.2.3.1. Роль и место в современных архитектурах .....	7
2.2.3.2. Области использования .....	7
2.2.3.3. Внутреннее устройство .....	7
2.2.4. Лекция 6: ClickHouse .....	7
2.2.4.1. Архитектура системы хранения и обработки данных .....	7
2.2.4.2. Партиционирование .....	7
2.2.4.3. Таблицы серии MergeTree как основной механизм персистентности .....	8
2.2.5. Лекция 7: Apache Spark .....	8
2.2.5.1. Роль и место в современных архитектурах .....	8
2.2.5.2. Области использования .....	8
2.2.5.3. Внутреннее устройство .....	8
2.2.5.4. API .....	8
2.2.6. Лекция 8: Erlang .....	8
2.2.6.1. Роль и место в современных архитектурах .....	8

2.2.6.2. Erlang VM (BEAM) и процессы .....	8
2.2.6.3. Модель акторов .....	8
2.2.6.4. Fault tolerance и “Let it crash” философия .....	8
2.2.6.5. OTP фреймворк (Supervisor, GenServer) .....	8
2.2.6.6. Синхронизация и распределённые вычисления .....	8
2.2.6.7. Elixir .....	8
2.3. Модуль 3: Распределённые БД .....	8
2.3.1. Лекция 9: Большие данные .....	8
2.3.1.1. Свойства .....	8
2.3.1.2. Архитектуры .....	8
2.3.2. Лекция 10: Full Text Search в распределённых системах .....	8
2.3.2.1. Elasticsearch и Solr .....	8
2.3.2.2. Распределённый поиск .....	8
2.3.2.3. Масштабирование .....	8
2.3.3. Лекция 11: Документные хранилища .....	9
2.3.3.1. MongoDB: sharding strategies, write concerns .....	9
2.3.3.2. Object storage: MinIO .....	9
2.3.4. Лекция 12: Распределённые пространственные и графовые БД .....	9
2.3.4.1. Графовые БД .....	9
2.3.4.2. Пространственные БД .....	9
2.3.4.3. Шардирование графов и пространственных данных .....	9

# 1. Архитектура информационных систем

## 1.1. Модуль 1: Архитектурные основы

### 1.1.1. Лекция 1: Введение. Принципы построения приложения

#### 1.1.1.1. Оптимизация системы требует выбора критерия оптимизации

#### 1.1.1.2. Принципы SOLID

#### 1.1.1.3. Dependency Injection

#### 1.1.1.4. Clean Architecture

### 1.1.2. Лекция 2: Альтернативные архитектурные паттерны

#### 1.1.2.1. Архитектура на основе фреймворка (Rails, Django)

#### 1.1.2.2. Модель акторов (Erlang/Elixir/Akka)

#### 1.1.2.3. Проектирование, ориентированное на данные (Data-oriented design)

### 1.1.3. Лекция 3: Протоколы взаимодействия

#### 1.1.3.1. REST: стандартный веб-протокол

#### 1.1.3.2. GraphQL: клиент запрашивает только нужные данные

#### 1.1.3.3. gRPC: бинарный протокол, высокая производительность

#### 1.1.3.4. WebSocket: двусторонний канал

## 1.2. Модуль 2: Фундаментальные структуры хранилищ данных

### 1.2.1. Лекция 4: Базовые механизмы хранения данных

#### 1.2.1.1. CSV-файлы как минимальная форма персистентности

**1.2.1.2. CSV с добавлением записей и журналированием изменений**

**1.2.1.3. Хеш-индексы как первый уровень ускорения выборки**

**1.2.2. Лекция 5: Структурированные индексы**

**1.2.2.1. Отсортированная таблица строк (Sorted String Table, SSTable)**

**1.2.2.2. Дерево слияния структурированного журнала (Log-Structured Merge-Tree, LSM-Tree)**

**1.2.2.3. Когда использовать LSM-деревья и B-деревья (LSM хороша для записи, B-Tree для чтения)**

**1.2.3. Лекция 6: ACID, уровни изоляции, аналитические и транзакционные хранилища**

**1.2.3.1. ACID (Atomicity, Consistency, Isolation, Durability)**

**1.2.3.2. Уровни изоляции**

- Read Uncommitted
- Read Committed
- Repeatable Read
- Serializable

**1.2.3.3. Аналитические базы данных (OLAP) и колоночные хранилища**

**1.2.3.4. Сжатие колонок**

**1.2.3.5. Сравнение подходов**

**1.3. Модуль 3: Другие хранилища данных**

**1.3.1. Лекция 7: Аналитика в оперативной памяти**

**1.3.1.1. Vectorized execution (блочная обработка)**

**1.3.1.2. DuckDB**

- Чтение из различных источников
- Трансформация данных
- Запись

**1.3.2. Лекция 8: Кэш в оперативной памяти**

**1.3.2.1. Skip List**

**1.3.2.2. Incremental Rehashing**

**1.3.2.3. HyperLogLog**

**1.3.2.4. Redis**

- Архитектура (однопоточность, event-driven)

**1.4. Модуль 4: Методы поиска похожих/близких элементов**

**1.4.1. Лекция 9: Специализированные индексы для поиска**

**1.4.1.1. Inverted Index (перевёрнутые индексы)**

- Постинг-листы (posting lists)
- Применение: Elasticsearch, Lucene

**1.4.1.2. Trie (префиксные деревья)**

- Автодополнение, префиксный поиск

**1.4.1.3. Bitmap indexes**

- Для категориальных данных

## **1.4.2. Лекция 10: Векторные индексы для поиска подобия**

### **1.4.2.1. Основы векторных представлений**

#### **1.4.2.2. HNSW (Hierarchical Navigable Small World)**

- Граф как индекс
- Поиск соседей

#### **1.4.2.3. LSH (Locality-Sensitive Hashing)**

- Вероятностный поиск

#### **1.4.2.4. Примеры: Weaviate, Pinecone, Qdrant**

## **1.4.3. Лекция 11: Геопространственные структуры**

### **1.4.3.1. R-Tree**

- Bounding boxes, MBR
- Применение: PostGIS

### **1.4.3.2. QuadTree и KD-Tree**

- Разбиение пространства
- Когда какое дерево эффективнее

## **1.5. Модуль 5: Операционные компоненты**

### **1.5.1. Лекция 12: Прокси и Rate Limiting**

#### **1.5.1.1. Forward Proxy / Reverse Proxy**

#### **1.5.1.2. Распределение нагрузки**

- Round Robin
- Least Connections
- Weighted Least Connections
- IP Hash
- Consistent Hashing

#### **1.5.1.3. Ограничение частоты запросов**

- Token Bucket
- Sliding Window

### **1.5.2. Лекция 13: Безопасность и управление доступом**

#### **1.5.2.1. Аутентификация и авторизация**

#### **1.5.2.2. RBAC и ABAC**

#### **1.5.2.3. Single Sign-On**

### **1.5.3. Лекция 14: Логирование, мониторинг и планирование задач**

#### **1.5.3.1. Сбор логов**

#### **1.5.3.2. Сбор метрик (Prometheus)**

#### **1.5.3.3. Мониторинг и диагностика (Grafana)**

#### **1.5.3.4. Планировщики (Celery, Airflow, Dagster)**

## **1.6. Модуль 6: System Design**

### **1.6.1. Лекция 15: Разбор дизайна конкретной системы**

#### **1.6.1.1. Применение всех концепций на практике**

## **2. Распределённые системы обработки данных**

### **2.1. Модуль 1: Компоненты распределённых систем**

#### **2.1.1. Лекция 1: Основы распределённых систем**

##### **2.1.1.1. CAP-теорема**

- Следствия компромиссов между согласованностью, доступностью и устойчивостью к разделению сети
- Примеры CA, AP, CP систем

##### **2.1.1.2. Механизмы репликации данных**

- Архитектура “ведущий-подчинённый” (Leader-Follower)
- Репликация с использованием кворумов (quorum-based replication)

##### **2.1.1.3. ZooKeeper: распределённое решение проблем согласованности и отказоустойчивости**

- Механизмы решения классических проблем распределённых систем
- Применение в системах Kafka и ClickHouse

#### **2.1.2. Лекция 2: Гарантии консистентности и координация распределённых транзакций**

##### **2.1.2.1. Алгоритмы достижения консенсуса**

- Raft
- Paxos

##### **2.1.2.2. Модель конечной консистентности (Eventual consistency) и требование идемпотентности операций**

##### **2.1.2.3. Протокол двухфазного коммита (Two-Phase Commit, 2PC)**

- Этапы выполнения
- Гарантии надёжности

### **2.2. Модуль 2: Распределённые сервисы**

#### **2.2.1. Лекция 3: Асинхронное взаимодействие сервисов**

##### **2.2.1.1. Event Bus и Event-Driven Architecture**

##### **2.2.1.2. Publish-Subscribe vs Message Queue**

##### **2.2.1.3. Saga Pattern**

#### **2.2.2. Лекция 4: Kafka**

##### **2.2.2.1. Внутреннее устройство**

#### **2.2.3. Лекция 5: Kubernetes**

##### **2.2.3.1. Роль и место в современных архитектурах**

##### **2.2.3.2. Области использования**

##### **2.2.3.3. Внутреннее устройство**

#### **2.2.4. Лекция 6: ClickHouse**

##### **2.2.4.1. Архитектура системы хранения и обработки данных**

- Принципы организации хранения
- Методы компрессии данных

##### **2.2.4.2. Партиционирование**

- Выбор ключа
- Управление партициями (добавление, удаление)

#### **2.2.4.3. Таблицы серии MergeTree как основной механизм персистентности**

- Базовая таблица MergeTree и её специализированные варианты
- ReplacingMergeTree для управления версионированием
- ReplicatedMergeTree для обеспечения отказоустойчивости

#### **2.2.5. Лекция 7: Apache Spark**

##### **2.2.5.1. Роль и место в современных архитектурах**

##### **2.2.5.2. Области использования**

##### **2.2.5.3. Внутреннее устройство**

##### **2.2.5.4. API**

- Dataframe
- SparkSQL

#### **2.2.6. Лекция 8: Erlang**

##### **2.2.6.1. Роль и место в современных архитектурах**

##### **2.2.6.2. Erlang VM (BEAM) и процессы**

##### **2.2.6.3. Модель акторов**

##### **2.2.6.4. Fault tolerance и “Let it crash” философия**

##### **2.2.6.5. OTP фреймворк (Supervisor, GenServer)**

##### **2.2.6.6. Синхронизация и распределённые вычисления**

##### **2.2.6.7. Elixir**

### **2.3. Модуль 3: Распределенные БД**

#### **2.3.1. Лекция 9: Большие данные**

##### **2.3.1.1. Свойства**

- Объем
- Скорость
- Разнообразие
- Достоверность

##### **2.3.1.2. Архитектуры**

- Modern Data Architecture
- Lambda
- Lakehouse

#### **2.3.2. Лекция 10: Full Text Search в распределённых системах**

##### **2.3.2.1. Elasticsearch и Solr**

- Sharding: hash-based, range-based
- Replication и replica factor
- Eventual consistency при индексировании

##### **2.3.2.2. Распределённый поиск**

- Broadcast query, merge результатов
- Search After вместо offset/limit

##### **2.3.2.3. Масштабирование**

- Hot shards проблема



- Index rollover, segment merging

### **2.3.3. Лекция 11: Документные хранилища**

#### **2.3.3.1. MongoDB: sharding strategies, write concerns**

#### **2.3.3.2. Object storage: MinIO**

### **2.3.4. Лекция 12: Распределенные пространственные и графовые БД**

#### **2.3.4.1. Графовые БД**

#### **2.3.4.2. Пространственные БД**

#### **2.3.4.3. Шардирование графов и пространственных данных**