1. Структура курса: Архитектура информационных систем (17 лекций)

1.1. Модуль 1: Архитектурные основы (Лекции 1-3)

1.1.1. Лекция 1: Введение. Принципы построения приложения

- Оптимизация системы требует выбора критерия оптимизации. (человеческие ресурсы, масштабируемость, надежность, стоимость)
- Каждая проблема в отдельности решаема, но комбинация простых решений приводит к экспоненциальному росту сложности системы.
- SOLID принципы и Dependency Injection на практике
- Clean Architecture

1.1.2. Лекция 2: Альтернативные архитектурные паттерны

Не всегда SOLID и Clean Architecture — оптимальный выбор. Разные подходы лучше работают в разных контекстах.

- Framework-driven (Rails, Django): быстрый старт за счёт конвенций, но vendor lock-in
- Actor Model (Erlang/Elixir/Akka): изоляция состояния, встроенная отказоустойчивость и параллелизм
- Data-oriented design: максимум производительности, ориентирован на конкретное железо
- Functional Core, Imperative Shell: чистые функции в логике, побочные эффекты на границах

1.1.3. Лекция 3: API дизайн для распределённых систем

Выбор протокола взаимодействия влияет на производительность, масштабируемость и удобство разработки.

- REST: стандартный веб-протокол, простая отладка, но избыточный трафик с комплексными данными
- GraphQL: клиент запрашивает только нужные данные, снижает трафик, сложнее кэшировать
- gRPC: бинарный протокол, высокая производительность, использование в микросервисах
- WebSocket: двусторонний канал, необходим для реал-тайм, требует управления состоянием

1.2. Модуль 2: Фундаментальные структуры современных хранилищ данных (Лекции 4-6)

Связь с Модулем 1: Архитектурные решения требуют компромиссов между критериями оптимизации. Модуль 2 демонстрирует это на примере хранилищ данных, показывая, как разные структуры данных решают одну задачу под разные задачи.

1.2.1. Лекция 4: Базовые механизмы хранения данных

Рассматривается путь от простых файлов к системам с устойчивостью к сбоям.

- CSV-файлы как минимальная форма персистентности
- Append-only CSV: идея Write-Ahead Log и журналирования изменений
- Хеш-индексы как первый уровень ускорения выборки
- Ограничения и проблемы таких подходов (фрагментация, поиск, консистентность)

1.2.2. Лекция 5: Структурированные индексы

Продолжение темы индексации и оптимизации доступа. Фокус на идее упорядоченности и организации данных для масштабных систем.

- Sorted String Table (SSTable): концепция сортированных сегментов
- Log-Structured Merge-Tree (LSM-Tree): лог-ориентированная запись и фоновая компакция
- B-деревья и их модификации
- Сравнение LSM- и B-деревьев: производительность, стоимость обновлений, области применения
- Вывод: компромиссы между скоростью записи и чтения

1.2.3. Лекция 6: ACID, уровни изоляции, отличие аналитических и транзакционных хранилищ

Завершающая лекция объединяет принципы консистентности, параллелизма и аналитических нагрузок.

- ACID и уровни изоляции: практическое влияние на производительность
- OLAP и колоночные базы данных (пример — ClickHouse): принципы сжатия и эффективных агрегаций
- Сравнение подходов: CSV / Append-only / Hash Index / SSTable / LSM / B-Tree
- Итоговая таблица CRUD-сложности и применимости

1.3. Модуль 3: Нереляционные и специализированные БД (Лекции 7-10)

Модуль 3 показывает, как разные типы нереляционных и специализированных БД решают задачи, где традиционные структуры становятся ограничением — по масштабируемости, латентности или типу данных.

1.3.1. Лекция 7: Документные хранилища

- MongoDB: sharding strategies, write concerns

1.3.2. Лекция 8:

- Inverted indexes: Lucene (Elasticsearch, Solr)
- Time-series: Prometheus architecture, downsampling
- Object storage: MinIO

1.3.3. Лекция 9:

- Geospatial: Minimum Bounding Rectangles, R-Tree, QuadTree
- Графовые БД: Список смежности, Матрица смежности, B-Tree для индексации

1.3.4. Лекция 10: Векторные БД

- Вектор как единица данных: эмбеддинги текста, изображений, аудио
- Семантический поиск документов / Поиск похожих изображений
- Inverted File Indexing
- Hierarchical navigable small world

1.3.5. Лекция 11: In-memory БД

- DuckDB: встроенная аналитика и векторизованное выполнение запросов
- Redis: структуры данных и модели персистентности

1.4. Модуль 4: Компоненты распределённых систем (Лекции 12-15)

Модуль 4 показывает как компоненты распределённых систем решают фундаментальные проблемы консистентности, надёжности и производительности в масштабе.

1.4.1. Лекция 12: Основы распределенных систем

Ключевой вызов распределённых систем заключается в необходимости согласования состояния при условиях неопределённости сетевых задержек, отказов узлов и асинхронности коммуникации.

- Теорема CAP и практические примеры компромиссов
- Репликация
  - Лидер-последователь (Leader-Follower)
  - На основе кворума (quorum-based)
- Консенсусные алгоритмы
  - Raft
  - Paxos
- Окончательная консистентность (Eventual consistency) и идемпотентность операций
- Двухфазный коммит (2PC, Two-Phase Commit)

1.4.2. Лекция 13:

- Forward Proxy
- Reverse Proxy
  - Load Balancing
    - Round-Robin
    - Least Connections

    - IP Hash
- Rate Limiting
  - Fixed Window
  - Token Bucket
  - Leaky Bucket
  - Sliding Window Log / Counter

1.4.3. Лекция 14:

- Authentication / Authorization
- Управление доступом
  - На основе ролей (Role-Based Access Control, RBAC)
  - На основе аттрибутов (Attribute-Based Access Control, ABAC)
- Единый вход (Single Sign-On)

1.4.4. Лекция 15: Асинхронная коммуникация

Асинхронная коммуникация через очереди сообщений и события.

- Event Bus и архитектура, ориентированная на события (Event Driven Architecture)
- Publish-Subscribe vs Message Queue
- Saga Pattern: распределённые транзакции без двухфазного коммита, компенсирующие действия при ошибках
- Гарантии доставки сообщений: at-most-once (может потеряться), at-least-once (могут быть дубли), exactly-once (идеал, но дорого)

1.4.5. Лекция 16:

- Логгирование
  - structured logging
  - log aggregation
  - ELK stack
- Observatility & Monitoring
  - Prometheus
  - Grafana
  - alerting
- Планировщики задач
  - Примитивы
    - Direct Acyclic Graphs
    - Queue / Worker Pool
  - Конкретные примеры
    - Celery
    - Airflow
    - Dagster

(остановился тут, дальше сгенерено)

1.5. Модуль 5: System Design (Лекция 17)

1.5.1. Лекции 17: разбор дизайна конкретной системы