Департамент образования города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики, управления и технологий

Тяпкина Полина Андреевна

Лабораторная работа по «Инструменты для хранения и обработки больших данных»

Выполнил:
Тяпкина Полина Андреевна
Проверил:
Босенко Тимур Муртазович

Курс обучения: 4
Форма обучения: очная

Москва
2025

Лабораторная работа 3-1. Проектирование архитектуры хранилища больших данных

Цель работы: разработать комплексную архитектуру хранилища больших данных для предложенного бизнес-сценария, обосновать выбор технологического стека и визуализировать потоки данных.

Оборудование и программное обеспечение

- Инструменты для создания диаграмм: draw.io (Diagrams.net), Miro, Lucidchart или аналоги.
- Доступ к сети Интернет для исследования технической документации по современным платформам и инструментам для работы с большими данными.

Краткая теоретическая справка

Архитектура больших данных — это каркас, который описывает сбор, хранение, обработку, анализ и визуализацию больших и сложных наборов данных. Правильно спроектированная архитектура является основой для извлечения ценных инсайтов и принятия решений на основе данных.

- Data Lake (Озеро данных): централизованное хранилище, которое позволяет хранить огромные объемы структурированных, полуструктурированных и неструктурированных данных в их исходном, необработанном формате. Основная идея — собрать все данные в одном месте для последующего анализа.
- Data Warehouse (Хранилище данных, DWH): система, предназначенная для хранения и анализа структурированных данных из различных источников. Данные предварительно очищаются, трансформируются (ETL/ELT) и моделируются для оптимизации аналитических запросов и отчетности.
- Data Lakehouse: современная гибридная архитектура, которая сочетает в себе гибкость и масштабируемость Data Lake с производительностью и возможностями управления данными Data Warehouse. Она позволяет выполнять BI-запросы и задачи

машинного обучения непосредственно на данных в озере, используя такие форматы, как Apache Iceberg, Delta Lake или Apache Hudi.

Компоненты архитектуры:

- Слой сбора данных (Ingestion): Apache Kafka, Confluent, Amazon Kinesis, Vector, Airbyte, Fivetran.
- Слой хранения (Storage): облачные хранилища (Amazon S3, Google Cloud Storage), форматы таблиц для Data Lakehouse (Delta Lake, Apache Iceberg, Hudi).
- Слой обработки (Processing): Apache Spark, Apache Flink, облачные платформы (Databricks, Snowflake, Google BigQuery), аналитические СУБД (ClickHouse, Greenplum).
- Слой аналитики и визуализации (Analytics & Visualization): Tableau, Power BI, Metabase, Looker, Apache Superset, Grafana.
- Слой оркестрации (Orchestration): Apache Airflow, Dagster, Prefect.
- Управление данными (Data Governance): Apache Atlas, Amundsen, OpenMetadata.

Задание для самостоятельной работы

Задача: Платформа "Умного города": управление городской инфраструктурой (светофоры, освещение), анализ транспортных потоков, мониторинг экологической обстановки. Источники: данные с дорожных камер, датчиков качества воздуха, общественного транспорта.

Цель работы: Разработать комплексную архитектуру хранилища больших данных для платформы "Умного города", обеспечивающую эффективное управление городской инфраструктурой, анализ транспортных потоков и мониторинг экологической обстановки. Архитектура должна обеспечивать надежное хранение, обработку и анализ больших объемов разнородных данных, а также поддерживать современные требования к безопасности, масштабируемости и отказоустойчивости.

1. Анализ требований

Источники данных:

- Дорожные камеры (потоковое видео/изображения, структурированные данные с распознавания)
- Датчики качества воздуха (потоковые и периодические значения, полуструктурированные данные)
- Общественный транспорт (GPS-трекинг, расписание, структурированные данные)

Типы данных:

- Структурированные: телеметрия транспорта, светофоров, освещения
- Полуструктурированные: данные с датчиков, JSON из систем мониторинга
- Потоковые данные: видео и сенсорные данные в реальном времени

Объемы и скорость:

- Высокая скорость поступления данных с камер и датчиков (потоковые данные)
- Ежесуточное накопление десятков-тепрограмм терабайт данных с видео и телеметрии

Бизнес-цели:

- Аналитика в реальном времени: управление светофорами для оптимизации движения
- Пакетная аналитика: анализ загруженности дорог, прогнозы трафика
- Мониторинг экологии и оперативные оповещения
- Визуализация и дашборды для операторов и управленцев
- Возможность обучения моделей машинного обучения (например, предсказание транспортных заторов)

2. Выбор компонентов архитектуры и технологического стека

Слой	Инструменты и технологии	Обоснование выбора
Сбор данных (Ingestion)	Apache Kafka — потоковая передача данных	Высокая пропускная способность, надежность
	Vector — для сбора данных с различных источников	Универсальность, легкая интеграция
	Airbyte — интеграция внешних систем	Быстрая коннективность
Хранение (Storage)	Облачное хранилище Amazon S3	Масштабируемость, надежность

	Delta Lake (на базе Apache Spark)	Data Lakehouse — ACID-транзакции, надежное хранение
Обработка (Processing)	Apache Spark — пакетная работа, анализ больших данных	Широкая поддержка, гибкость
	Apache Flink — реальное время	Минимальная задержка обработки
Аналитика и машинное обучение	Grafana (для дашбордов мониторинга)	Хорошая интеграция с Prometheus и потоковыми данными
	Power BI / Apache Superset	BI-аналитика, удобство создания отчетов
Оркестрация и мониторинг	Apache Airflow	Автоматизация ETL-процессов
	Prometheus + Grafana	Слежение за системой и своевременное оповещение
Управление данными	Apache Atlas	Каталогизация и управление метаданными

3. Проектирование архитектуры и потоков данных

- Данные с дорожных камер, датчиков воздуха и транспорта поступают в систему через слой сбора — Kafka и Vector.
- Потоковые данные сразу передаются в слой обработки на Flink для анализа в реальном времени (например, изменение сигналов светофоров).

- Данные также сохраняются в Delta Lake на S3 для дальнейшей пакетной обработки Apache Spark (например, анализ трендов и построение отчетов).
- Результаты аналитики и агрегированные данные сохраняются в DWH-слое (оптимизированный для BI).
- Аналитики и операторы получают доступ к дашбордам через Grafana и Power BI.
- Оркестрация ETL-процессов обеспечивается Airflow.
- Безопасность реализована через шифрование данных, Kerberos-аутентификацию и контроль доступа Apache Ranger.
- Мониторинг системы ведется с помощью Prometheus и Grafana.

4. Масштабирование и отказоустойчивость

- Горизонтальное масштабирование Kafka, Spark и Flink кластерами.
- Репликация данных Delta Lake с multi-AZ (Availability Zone).
- Автоматическое масштабирование ETL и аналитических процессов с Kubernetes (если применимо).
- Резервное копирование данных и аварийное восстановление.

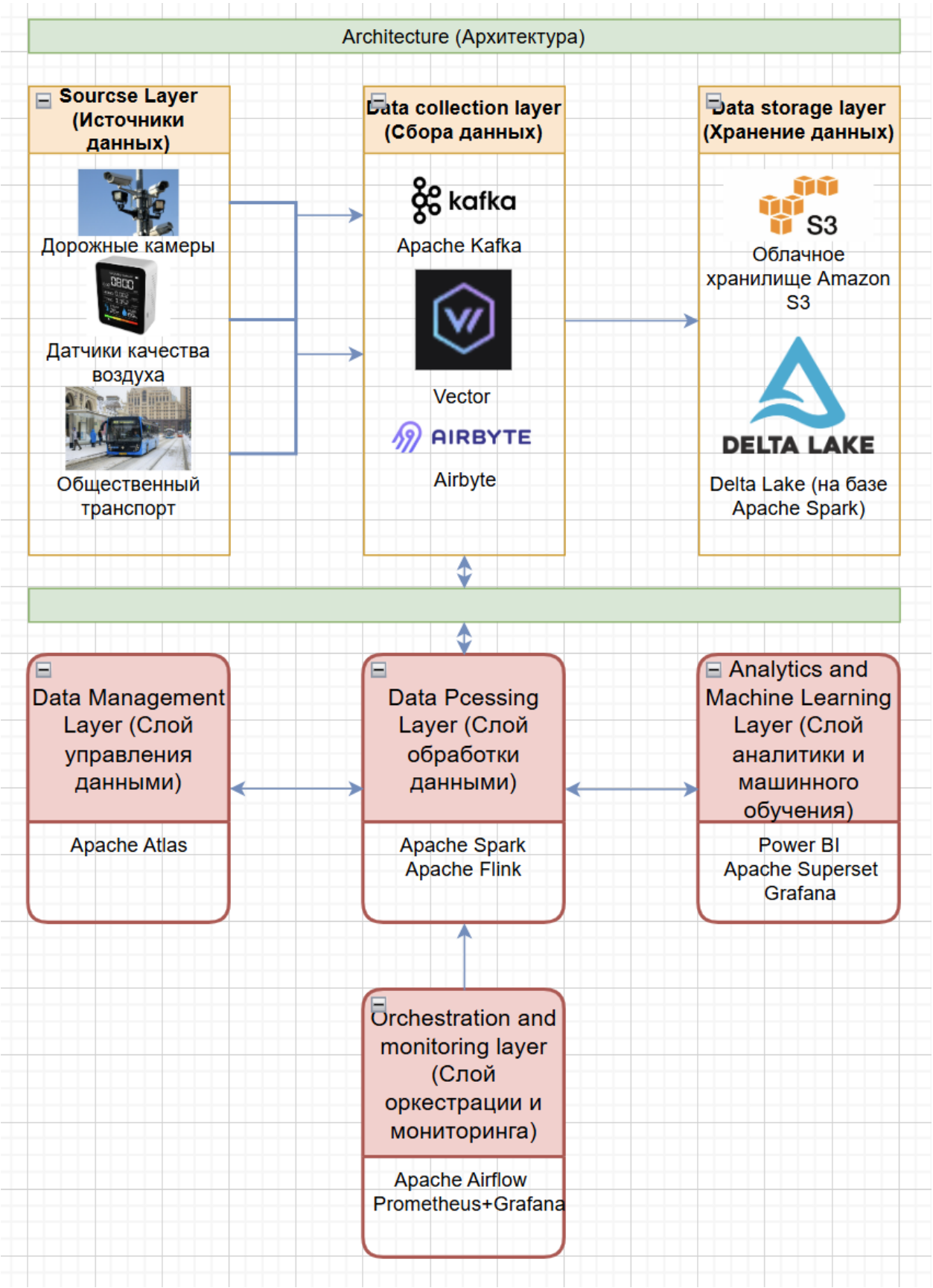
5. Потенциальные проблемы и решения

- Узкое место 1: Высокая нагрузка на потоковую обработку данных из камер.
    - Решение: Использование Apache Flink с выделенными ресурсами и оптимизацией топологии потоков.
- Узкое место 2: Рост стоимости хранения видео данных и долговременного хранения.
    - Решение: Архивное хранение с tiered storage в S3 Glacier и периодический чистка данных по срокам.

- Узкое место 3: Обеспечение безопасности и конфиденциальности данных граждан.
    - Решение: Полное шифрование, аудит доступа, многофакторная аутентификация и соответствие нормативам защиты данных.

6. Краткое описание выбранных ключевых компонентов

- Apache Kafka: Обеспечивает надежную транспортировку потоковых данных между системами.
- Delta Lake: Позволяет надежно хранить данные в формате Data Lakehouse, обеспечивая согласованность.
- Apache Flink: Позволяет проводить низколатентную обработку потоковых данных для оперативных решений.
- Apache Spark: Используется для пакетной аналитики и подготовки данных для BI-отчетов.
- Grafana и Power BI: Инструменты визуализации для мониторинга и управленческой отчетности.
- Apache Airflow: Управляет мониторингом и автоматизацией рабочих процессов.
- Apache Ranger и Atlas: Обеспечивают безопасность доступа и управление метаданными.


Architecture (Архитектура)
Sourcse Layer (Источники данных)
Дорожные камеры
Датчики качества воздуха
Общественный транспорт
Data collection layer (Сбора данных)
kafka
Apache Kafka
Vector
AIRBYTE
Airbyte
Data storage layer (Хранение данных)
S3
Облачное хранилище Amazon S3
DELTA LAKE
Delta Lake (на базе Apache Spark)
Data Management Layer (Слой управления данными)
Apache Atlas
Data Pcessing Layer (Слой обработки данными)
Apache Spark
Apache Flink
Analytics and Machine Learning Layer (Слой аналитики и машинного обучения)
Power BI
Apache Superset
Grafana
Orchestration and monitoring layer (Слой оркестрации и мониторинга)
Apache Airflow
Prometheus+Grafana

Итоги и выводы

В ходе лабораторной работы была разработана и проанализирована комплексная архитектура хранилища больших данных, адаптированная под задачи управления городской инфраструктурой, анализа транспортных потоков и мониторинга экологической обстановки.

Ключевые итоги:

- Проведен детальный анализ источников данных, типов, объемов и требований к скорости поступления и обработке, что позволило четко определить архитектурные потребности.
- Выбран современный и сбалансированный стек технологий с использованием Apache Kafka и Vector для надежного сбора потоковых данных, Delta Lake для хранения в формате Data Lakehouse, а также Apache Flink и Apache Spark для обработки в реальном времени и пакетной аналитики.
- Спроектирована архитектура, обеспечивающая целостность данных, масштабируемость и отказоустойчивость за счет горизонтального масштабирования, репликации и облачных решений.
- Организованы средства визуализации и оркестрации (Grafana, Power BI, Apache Airflow), что обеспечивает удобство мониторинга и управления рабочими процессами.
- Внедрены меры по обеспечению безопасности, включая контроль доступа, шифрование данных и аудит, что критично при работе с чувствительной городской информацией.
- Выявлены и проработаны потенциальные узкие места системы с предложениями по их устранению — например, оптимизация потоковой обработки, организация архивного хранения и усиление безопасности.
- Архитектура позволяет добиться бизнес-целей проекта: оперативное управление дорожным движением, мониторинг и реагирование на изменения в экологической обстановке, а также поддержку принятия решений на основе данных и внедрение машинного обучения.

Вывод: Разработанная архитектура хранилища больших данных является надежной, масштабируемой и функционально полной платформой для реализации «Умного города». Она обеспечивает эффективное управление различными источниками данных, высокую

производительность аналитических процессов и безопасность, что соответствует современным требованиям к городским IT-системам.