Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»

Институт цифрового образования
Департамент информатики, управления и технологий

Вариант 13

Лабораторная работа 3.1
Тема: «Проектирование архитектуры хранилища больших данных»
Дисциплина «Инструменты для хранения и обработки больших данных»

Выполнила:
Студентка группы АДЭУ-221
Селиверстова Светлана Николаевна

Преподаватель:
Тимур Муртазович Босенко
доцент, к.т.н.

Москва
2025

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

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Анализ требований:

Источники:

– спутниковые снимки,

– данные с датчиков в почве,

– показания с метеостанций.

Спутниковые снимки:

Тип данных: Неструктурированные данные (изображения), метаданные (дата и время, координаты локации и тд) — структурированные.

Объем: Большой (десятки-сотни ГБ в день на крупный холдинг).

Скорость: Пакетная загрузка 1-2 раза в день

Данные с датчиков в почве (влажность, температура, pH(кислотность), NPK-уровни (наличие питательных веществ в почве):

Тип данных: Структурированные данные (временные ряды).

Объем: Средний (миллионы показаний в день).

Скорость: Потоковая, постоянная отправка с интервалом.

Данные с метеостанций (температура, влажность, давление, осадки, скорость ветра, UV-излучение):

Тип данных: Структурированные данные (временные ряды).

Объем: Средний.

Скорость: Потоковая, постоянная отправка с интервалом.

Основные бизнес-цели и аналитика:

– Аналитика в реальном времени для мониторинга текущего состояния полей;

– Пакетная аналитика и ML-модели, использующие исторические данные (снимки, погода, урожайность прошлых лет) для прогноза урожайности;

– ML-модели для построения карт задач для распределения удобрений и полива;

– Отчеты для контроля эффективности использования ресурсов (воды, удобрений, топлива и тд).

2. Выбор компонентов архитектуры

Слой сбора данных:

– Apache Kafka (надежный, отказоустойчивый, для потоковых данных) и Airbyte (для пакетной загрузки данных из внешних API спутниковых провайдеров)

Слой хранения:

– MinIO (надежный, быстрый, стабильный доступ к данным, для неструктурированных данных) и PostgreSQL (хранение структурированных метаданных и результатов анализа)

Слой обработки:

– Apache Flink (обработка потоковых данных с датчиков в реальном времени),

– Spark (пакетная обработка снимков и сложные ML-вычисления)

Слой аналитики и машинного обучения:

– Apache Superset (визуализация, интеграция с spark),

– Jupyter Notebook (исследовательский анализ и ml - модели)

Слой оркестрации и мониторинга:

– Apache Airflow (для мониторинга обработки спутниковых снимков),

– Prometheus и Grafana (сбор метрик с Kafka, мониторинг производительности MinIO и PostgreSQL, отслеживание ресурсов Spark/Flink кластеров)

Управление данными:

– OpenMetadata (единый каталог данных)

– Apache Ranger (управление доступом к чувствительным данным)

3. Схема архитектуры

На рисунке 1 представлена схема архитектуры хранения больших данных для агрохолдинга.

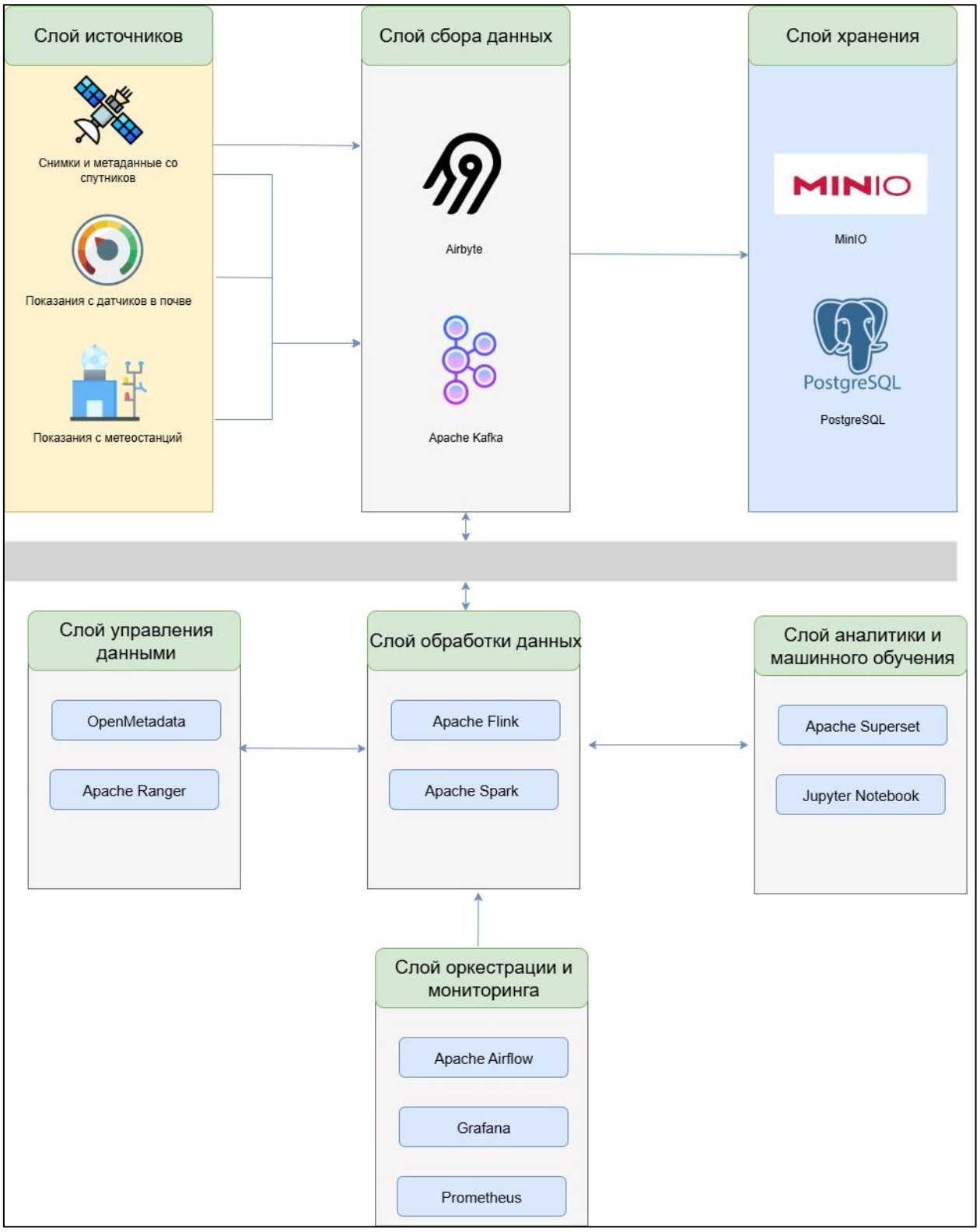

Рисунок 1 – Схема архитектуры хранилища больших данных

4. Процесс обработки данных

Логическая схема потоков данных:

Данные с почвенных датчиков (влажность, температура, NPK) и метеостанций поступают в Apache Kafka в реальном времени.

Apache Flink берет данные из Kafka, добавляет данные геопозиции и метаданные полей, агрегирует показатели по зонам полей, вычисляет производные метрики (тенденции изменений, отклонения от нормы), генерирует оповещения при критических значениях (засуха, переувлажнение), записывает очищенные данные в HDFS в формате Parquet, сохраняет агрегированные показатели в PostgreSQL для быстрого доступа.

Спутниковые снимки загружаются в MinIO через Airbyte по расписанию. Apache Airflow оркестрирует пайплайны обработки, запускает Spark Jobs для обработки снимков:

– Вычисление вегетационных индексов (NDVI, NDRE, MSAVI),

– Сегментация полей на зоны по состоянию растительности,

– Сравнение с историческими показателями,

– Сохранение результатов в PostgreSQL с геопривязкой.

Машинное обучение и аналитика:

Jupyter Notebook используется для:

– Исследовательского анализа данных,

– Разработки и тестирования ML-моделей прогнозирования урожайности,

– Визуализации важности признаков и диагностики моделей.

Apache Spark ML выполняет подготовку данных для обучения ИИ на исторических данных, обучение моделей прогнозирования урожайности (XGBoost, Random Forest), пакетное применение уже обученной модели ИИ для всех полей.

Apache Superset подключен к PostgreSQL и Spark SQL для:

– Построения бизнес-отчетов по эффективности хозяйства и применения ресурсов,

– Визуализации карт урожайности и вегетационных индексов,

– Анализа динамики изменения показателей за периоды.

Управление данными и мониторинг:

OpenMetadata интегрирован со всеми компонентами, он автоматически собирает метаданные из MinIO, PostgreSQL, Kafka, строит систему отслеживания происхождения данных от сырых до бизнес-отчетов, предоставляет единый каталог данных для аналитиков.

Apache Ranger управляет доступом:

– RBAC для чувствительных данных (урожайность, финансовые показатели),

– Политика доступа к данным в MinIO и PostgreSQL.

Prometheus и Grafana осуществляют мониторинг метрики производительности всех компонентов, бизнес-показателей в реальном времени, системных оповещений и метрик качества данных.

5. Масштабирование и отказоустойчивость

Горизонтальное масштабирование:

– Kafka: когда данных становится слишком много, добавляются в кластер новые серверы-брокеры. Нагрузка автоматически распределится между ними;

– MinIO: при нехватке места подключаются новые серверы с дисками. MinIO автоматически распределяет данные между узлами, поддерживая балансировку нагрузки;

– Spark/Flink кластеры: когда задачи по обработке снимков или потоковых данных начинают выполняться слишком медленно, добавляются новые вычислительные узлы. Диспетчер автоматически начнет отправлять задачи на новые узлы, ускоряя обработку.

Отказоустойчивость архитектуры обеспечивается на нескольких уровнях через репликацию данных, резервирование компонентов и механизмы самовосстановления.

6. Безопасность

– Шифрование данных: MinIO поддерживает шифрование на стороне сервера и на стороне клиента,

– Контроль доступа: Apache Ranger для управления политиками доступа к данным в MinIO и PostgreSQL через RBAC,

– Резервное копирование: Регулярные снимки состояния системы MinIO и PostgreSQL с возможностью восстановления в другой дата-центр.

7. Анализ потенциальных проблем и их решений

Проблема 1 - Рост объема спутниковых снимков

Риск: Переполнение хранилища, рост затрат

Решение:

– Настройка жизненного цикла данных в MinIO (автоматический перенос старых снимков в хранилище),

– Использование стирающего кода для эффективного использования дискового пространства,

– Политики автоматического удаления устаревших сырых данных после обработки.

Проблема 2 - Качество данных с датчиков

Риск: Некачественные данные, поломка датчиков

Решение:

– Валидация в Flink (проверка диапазонов значений),

– Машинное обучение для поиска аномалий,

– Автоматические уведомления о сбоях датчиков.