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

Вариант - 3

Лабораторная работа 3.1

Тема: «Проектирование архитектуры хранилища больших данных»

Дисциплина «Инструменты для хранения и обработки больших данных»

Выполнила:
Арлинская Александра Викторовна
Проверил:
Босенко Тимур Муртазович

Курс обучения: 4

Форма обучения: очная

Москва

2025

1. Определение требований

1.1 Объем данных:

- Ожидаемый объем: 100–200 ТБ годовых, с ростом ~40% в год (учитывая детализацию CDR/IPDR и логи с базовых станций).

1.2 Скорость поступления данных:

- Базовые станции: пайплайн в реальном времени, до 10 000 сообщений/секунда.
- Биллинг: обновления каждые несколько минут/часов, порциями.
- Техподдержка: обновления событий, тикеты, дневные выгрузки.

1.3 Типы данных:

- Структурированные: CDR (Call Detail Records), IPDR, биллинг, логи обращений.
- Полуструктурированные: json/xml отчеты по оборудованию, статистика нагрузки.
- Неструктурированные: текстовые обращения в поддержку, отчеты об инцидентах.

1.4 Требования к обработке:

- Прогнозирование нагрузки на сеть: почасово/помесячно.
- Анализ качества связи: ежедневные отчеты, оперативные панели.
- Отток клиентов (churn prediction): ежемесячные и ежеквартальные прогнозы.
- Интерактивная аналитика и визуализация для бизнес-аналитиков.

1.5 Доступность:

- Реальное время для мониторинга и прогнозирования.
- SLA — 99.9%, отклик <30 секунд для критических запросов.

1.6 Безопасность:

- Шифрование данных при хранении и передаче.
- Многофакторная аутентификация, сегментация доступов.
- Аудит действий, соблюдение 152-ФЗ, GDPR.

2. Выбор модели хранилища

- Data Lake — для хранения сырых CDR/IPDR, биллинга, логов.
- Data Warehouse — для агрегированных и аналитических данных по клиентам и нагрузке.
- Hybrid Storage — сочетание lake (быстрая потоковая запись) и warehouse (аналитика, отчеты).

3. Проектирование архитектуры

Источники:

Базовые станции (системы OSS/BSS, SNMP брокеры),

Биллинг-CRM,

Техподдержка (тикеты, чаты, обращения).

Слой сбора данных:

Kafka — потоковая доставка;

Logstash/Flink — обработка, маршрутизация.

Слой хранения:

HDFS — долгосрочное сырое хранение;

HBase/PostgreSQL — быстрый доступ к агрегированным, структурированным записям.

Слой обработки:

Apache Spark — пакетная/стрим-аналитика, ML обработка;

Apache Flink — обработка событий в реальном времени;

Apache Hive — SQL запросы к большому объему данных.

Слой аналитики и ML:

Jupyter Notebook, TensorFlow, Apache Superset — аналитика, построение ML, визуализация результатов и динамики по регионам/кластерам.

Слой управления и безопасности:

Apache Atlas — управление метаданными;

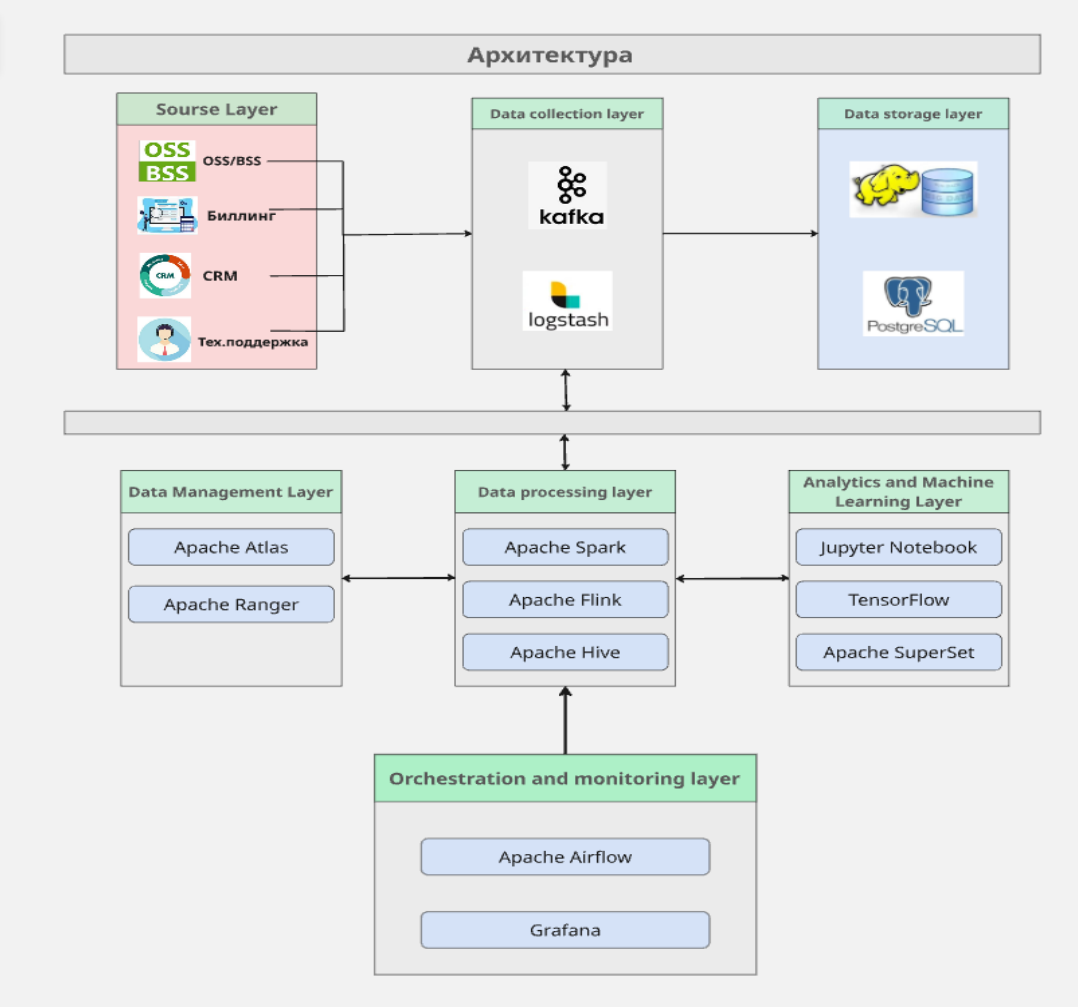
Apache Ranger — контроль доступа.

Оркестрация и мониторинг:

Apache Airflow — построение ETL пайплайнов, расписание задач;

Grafana — мониторинг процессов, визуализация.

4. Диаграмма



5. Описание компонентов выбранных инструментов и обоснование выбора

Apache Kafka

Роль: Потоковый брокер для сбора и доставки данных от источников (CDR/IPDR, биллинг, техподдержка) в систему.

Обоснование: Высокая производительность, масштабируемость и надежность; поддержка real-time данных с гарантией доставки. Аналоги (RabbitMQ, Pulsar) менее масштабируемы при таком объеме.

Logstash / Apache Flink

Роль: Logstash служит для агрегации и нормализации данных из разных источников, Flink — для обработки потоков в реальном времени с низкой задержкой.

Обоснование: Flink выбран за нативную поддержку потоковой обработки с минимальной задержкой (миллисекунды), что критично для мониторинга качества связи. Spark Streaming менее подходящ для ultra-low latency, так как использует micro-batching.

HDFS и Apache HBase

Роль: HDFS — долговременное хранение сырой информации, HBase — быстрая аналитика и модели доступа к структурированным данным.

Обоснование: Hadoop-экосистема проверена временем, хорошо интегрируется с другими компонентами; HBase обеспечивает быстрый доступ, в отличие от классических реляционных СУБД.

PostgreSQL

Роль: Хранение операционных метаданных, конфигураций, результатов агрегирования и отчетов.

Обоснование: Надежность и богатый функционал реляционной СУБД, широкая поддержка.

Apache Spark

Роль: Пакетная обработка больших объемов и построение ML моделей.

Обоснование: Обширная экосистема, поддержка SQL, ML libraries; подходит для задач batch-аналитики и создания витрин.

Jupyter Notebooks, TensorFlow, Apache Superset

Роль: Интерфейс исследователей данных и аналитиков, построение ML, визуализация данных.

Обоснование: Открытые и известные инструменты, обеспечивающие гибкость и масштабируемость аналитики.

Apache Atlas и Apache Ranger

Роль: Управление метаданными и контроль безопасности доступа.

Обоснование: Предоставляют прозрачность управления данными и соответствие требованиям безопасности.

Apache Airflow и Grafana

Роль: Оркестрация и автоматизация ETL и ML процессов, мониторинг системы и оповещения.

Обоснование: Стандартизированные решения с открытым исходным кодом для пайплайнов и мониторинга.

6. Анализ потенциальных проблем и их решений

Потоковая обработка с ultra-low latency (узкое место в Flink/Logstash):

Решение: масштабирование кластеров Flink, настройка параметров памяти и backpressure; мониторинг нагрузки для предотвращения узких мест.

Рост расходов на хранение больших объемов данных:

Решение: Использование Data Lifecycle Management, архивирование старых данных и tiered storage; внедрение сжатия и дельта-обновлений.

Сложность управления большим количеством потоков и коннекторов:

Решение: централизованный мониторинг через Grafana и Apache Atlas; автоматизация оркестрации через Airflow с alerting.