Департамент образования города Москвы

Государственное автономное образовательное учреждение высшего образования города Москва «Московский Городской Педагогический Университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ЛАБОРАТОРНАЯ РАБОТА №3.1 (Вар.9)

по дисциплине «Инструменты для хранения и обработки больших данных»

Тема: «Проектирование архитектуры хранилища больших данных»

Направление подготовки 38.03.05 – «бизнес-информатика»

Выполнила:

Студентка группы АДЭУ-221

Муханова А. И.

Преподаватель:

Босенко Т. М.

к.т.н., доц. департамента

информатики, управления и

технологий

Москва

2025

1. Цель работы

Сценарий: образовательная онлайн-платформа, предназначенная для персонализации траекторий обучения, анализа успеваемости студентов, рекомендации курсов и проведения A/B-тестирования новых функций.

Источники данных: логи взаимодействия с платформой (clickstream), результаты тестов, видеолекции, метаданные курсов и профили пользователей.

Типы данных:

Структурированные — профили пользователей, результаты тестов, оценки, расписания.

Полуструктурированные — логи взаимодействия (JSON), события кликов, события воспроизведения видео.

Неструктурированные — видеолекции, текстовые отзывы, загруженные файлы.

Объёмы и скорость:

- до 1–3 ТБ данных в год,

- потоковая генерация событий: 1 000–5 000 сообщений/сек,

- пакетные загрузки: результаты тестов, выгрузки LMS.

Скорость поступления: смешанный режим — потоковые события (до нескольких тысяч событий в секунду) и пакетные загрузки (результаты тестов, дампы).

Бизнес-цели: персонализация обучения в near-real-time, мониторинг успеваемости, A/B-тестирование интерфейсов и рекомендаций, отчётность для преподавателей и администраторов.

Требования к задержке/доступности:

Персональные рекомендации: задержка < 2 секунд для интерактивных сценариев.

Отчёты и модельное обучение: пакетная обработка с периодичностью от часов до дней.

Требования к безопасности: VK Cloud IAM: разграничение ролей, шифрование данных в S3-хранилище.

2. Выбор компонентов архитектуры

Модель: Data Lakehouse — объединение гибкости Data Lake и управляемости DW.

Компоненты архитектуры:

Слой	**Реализация**
Источники	Веб и мобильное приложение, LMS, CDN, БД
Ingestion	Apache Kafka, Apache NiFi или Airbyte, Debezium (CDC)
Data Lake	VK Cloud Object Storage (S3), форматы Parquet/Delta Lake
DWH/Serving	ClickHouse (аналитика), PostgreSQL (OLTP)
Stream Processing	Apache Flink
Batch Processing	Apache Spark (на Kubernetes / VK Cloud)
Оркестрация	Apache Airflow
Метаданные & безопасность	OpenMetadata + Apache Ranger / VK Cloud IAM
ML	MLflow, JupyterHub, библиотека Feast

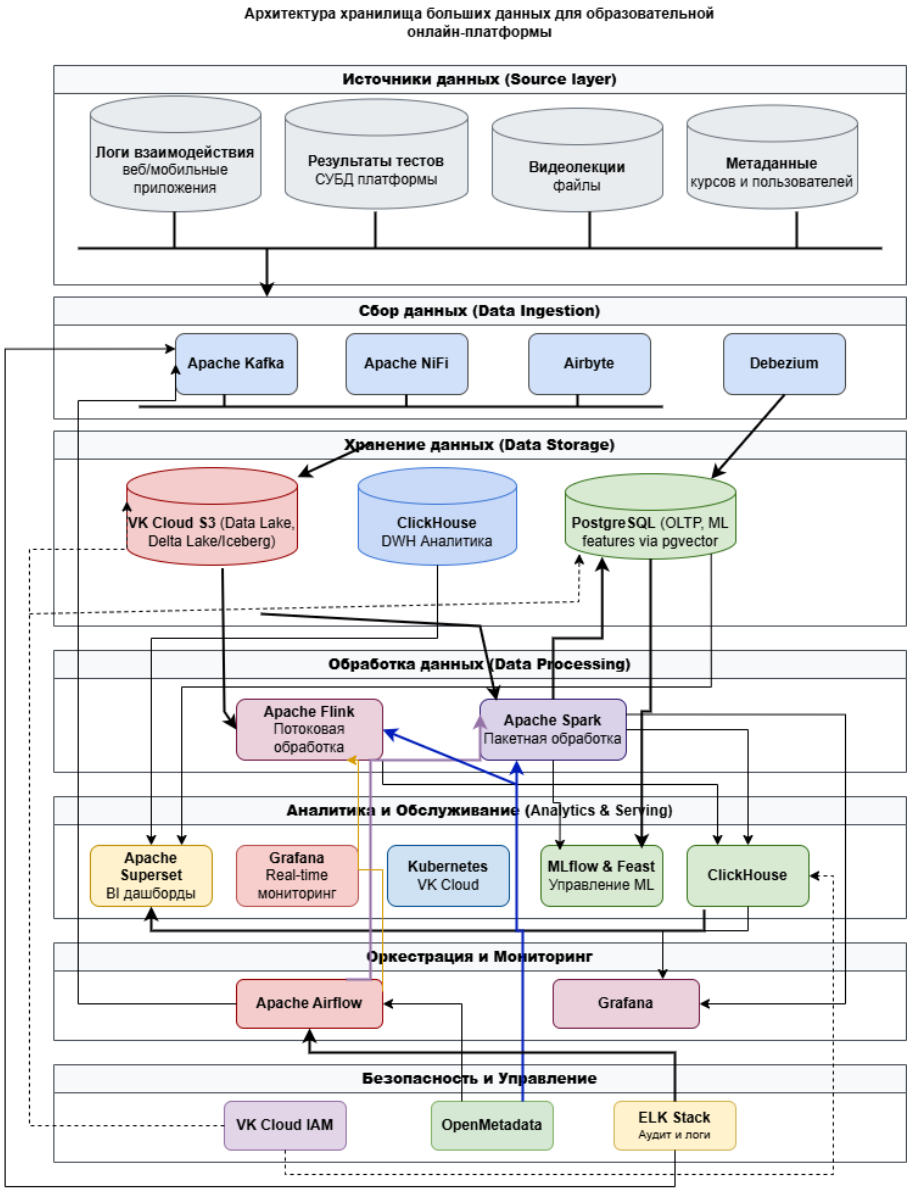
Визуализация	Apache Superset, Grafana
Мониторинг	Prometheus, Grafana, ELK Stack

3. Обоснование выбора

Компонент	**Почему выбран**
Kafka	Высокая скорость, гарантии доставки, буферизация событий платформы
VK Cloud S3 + Delta Lake	Хранение любых типов данных, поддержка версий и ACID-транзакций
ClickHouse	Быстрые аналитические запросы для администраторов и преподавателей
PostgreSQL	Надёжная реляционная БД для транзакционных данных (оценки, пользователи)
Flink	Потоковые рекомендации, обработка событий в реальном времени
Spark	Пакетные расчёты: успеваемость, отчёты, ML-обработка

Superset	Бесплатная альтернатива Tableau/Power BI, работает с ClickHouse
Airflow	Планирование ETL-процессов (ежедневные отчёты, обновление витрин)
MLflow + JupyterHub	Управление экспериментами, моделями и обучением рекомендаций
Prometheus + Grafana	Мониторинг сервисов, задержек, нагрузки
OpenMetadata / Ranger	Каталог данных, lineage, права доступа

4. Диаграмма архитектуры



5. Описание потоков данных

Пример 1 — Clickstream → Рекомендации:

1. Клиент (браузер) отправляет событие → Kafka.

2. Flink читает события → вычисляет активность, предпочтения, результаты публикует в ClickHouse и Kafka (для рекомендательного сервиса).
3. Spark периодически выгружает данные из S3 → строит витрины → обновляет ClickHouse.

Пример 2 — Результаты тестов:

1. LMS отправляет таблицу тестов → Airbyte/ NiFi → S3 (Raw Zone).
2. Spark обрабатывает → считает средний балл по курсам и студентам → пишет в ClickHouse и PostgreSQL.

Пример 3 — Видео и просмотры:

1. Видео хранится в VK Cloud Storage / CDN.
2. Метрики просмотра → Kafka → Flink → ClickHouse.

6. Отказоустойчивость и масштабируемость

Kafka — несколько брокеров и партиции тем.

Spark и Flink — запускаются в Kubernetes, масштабируются автоматически.

S3 — теоретически неограниченное хранилище.

checkpointing (Flink), snapshot (Spark), репликация ClickHouse.

Резервное копирование Airflow, OpenMetadata.

7. Потенциальные проблемы и решения

1) Рост стоимости хранения: Архивация: S3 Standard → Infrequent → Cold Storage

2) Сложность интеграции batch и stream: использовать Delta Lake как единый формат таблиц.

3) Конфиденциальность и соответствие: шифрование, IAM, аудит через Ranger, минимизация хранения PII и применение псевдонимизации.

8. Выводы

Предложенная архитектура обеспечивает персонализацию обучения, формирование рекомендаций в режиме реального времени и создание гибкой аналитической платформы для преподавателей и администраторов. Использование связки Apache Kafka + Apache Flink позволяет обрабатывать потоковые события с минимальной задержкой. Data Lake на базе VK Cloud S3 с поддержкой Delta Lake обеспечивает надёжное хранение как сырых, так и обработанных данных. Apache Spark реализует пакетную аналитику и построение витрин данных. Внедрение инструментов Data Governance, мониторинга и контроля доступа (OpenMetadata, Apache Ranger, Grafana, ELK) гарантирует качество данных и безопасность обработки персональной информации.