Департамент образования города Москвы

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

Институт цифрового образования
Департамент информатики, управления и технологий

ПРАКТИЧЕСКАЯ РАБОТА №2

по дисциплине «Инструменты для хранения и обработки больших данных»
Направление подготовки 38.03.05 – бизнес-информатика
Профиль подготовки «Аналитика данных и эффективное управление»
(очная форма обучения)

Выполнила:
Студентка группы АДЭУ-221
Вознесенская В. Е.

Проверил:
Босенко Т. М., доцент

Москва
2025

КОМПЛЕКСНАЯ АРХИТЕКТУРА ХРАНИЛИЩА БОЛЬШИХ ДАННЫХ ДЛЯ КРУПНОГО ОНЛАЙН-РИТЕЙЛЕРА

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

Вариант 1

Крупный онлайн-ритейлер: анализ поведения пользователей в реальном времени, прогнозирование спроса, персонализация рекомендаций. Источники: кликстрим с сайта/приложения, транзакции, отзывы клиентов, данные из CRM.

1. Описание бизнес-процесса

Бизнес-процесс: «Персональные рекомендации пользователю»

1.1 Цель процесса

Показывать каждому посетителю сайта/мобильного приложения персональные товары и акции, чтобы увеличить клики, конверсии и средний чек. Рекомендации должны работать в реальном времени (или near-real-time) и улучшаться по мере получения новых данных.

1.2 Участники процесса

— пользователь — заходит на сайт, просматривает товары, совершает покупки;
— сайт или приложение — показывает товары и собирает информацию о действиях пользователя;
— команда данных — собирает, хранит и анализирует информацию;
— команда маркетинга — использует результаты для акций и персональных предложений;
— аналитики — следят за показателями и оценивают эффективность рекомендаций

1.3 Источники данных

— кликстрим (clickstream) — все действия пользователя на сайте или в приложении: какие страницы он смотрит, куда нажимает, что добавляет в корзину;

— транзакции — информация о заказах: что куплено, когда и на какую сумму;

— отзывы — тексты и оценки, которые пользователи оставляют о товарах;

— CRM — данные о клиентах: профиль, история покупок, уровень лояльности и т. д.

1.4 Выход

— список персональных рекомендаций для каждого пользователя;

— статистика по тому, как часто пользователи кликают на рекомендации и покупают;

— отчёты и дашборды для аналитиков и менеджеров.

1.5 Основные этапы процесса

На рисунке 1.1 кратко показаны основные этапы данного бизнес-процесса.

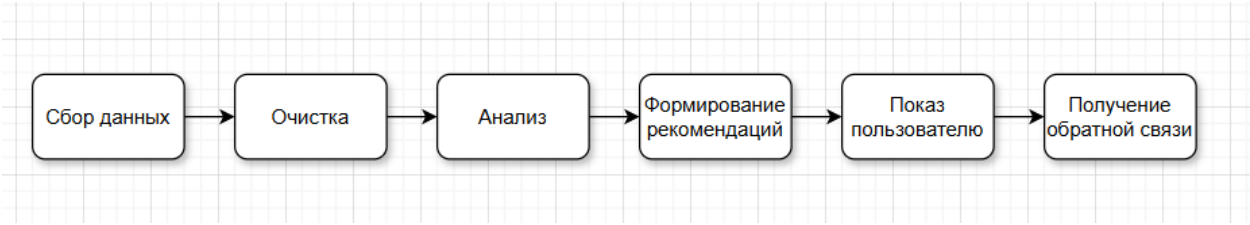

Рисунок 1.1 Основные этапы бизнес-процесса «Персональные рекомендации пользователю»

2. Анализ требований

2.1 Источники данных

Согласно бизнес-процессу у нас есть несколько источников данных, они представлены в таблице 2.1.

Таблица 2.1. Источники данных

Источник данных	Тип данных	Скорость поступления	Объем данных	Примечания/Использование
Кликстрим (clickstream)	Потоковые данные (события пользователя)	Реальное время / near-real-time	Высокий (много действий на сайте)	Используется для формирования персональных рекомендаций в реальном времени
Транзакции (заказы)	Структурированные (таблицы с покупками)	Пакетная загрузка (batch)	Средний	Используется для анализа покупок, формирования моделей рекомендаций
Отзывы	Неструктурированные (текст) + структурированные (оценки)	Периодически / batch	Небольшой — средний	Анализ отзывов для улучшения рекомендаций и оценки товаров
CRM (клиенты)	Структурированные (профиль, история покупок, уровень лояльности)	Периодически / batch	Небольшой	Используется для персонализации предложений и сегментации пользователей

Вывод по источникам: данные разнообразные по типу и скорости поступления. Кликстрим — потоковые и высокоскоростные, остальные — чаще загружаются пакетами (batch).

2.2 Бизнес-цели

— персональные рекомендации (рекомендации должны работать в реальном времени или почти в реальном времени (near-real-time), список товаров или акций, которые интересны каждому пользователю);

— аналитика эффективности(отслеживать клики на рекомендации и покупки, строить отчеты и дашборды для команды маркетинга и аналитиков;

— использование моделей машинного обучения (ML): (ML модели будут анализировать поведение пользователей и прогнозировать, что им может понравиться, модели должны обновляться по мере поступления новых данных).

Вывод: основная цель — увеличить доход компании за счёт персональных рекомендаций, которые могут увеличить средний чек у каждого пользователя.

2.3 Резюме требований

Скорость обработки данных:

— Кликстрим — real-time/near-real-time

— Транзакции, отзывы, CRM — batch (обновление периодически)

Типы данных:

— Структурированные: транзакции, CRM

— Потоковые: кликстрим

— Неструктурированные: тексты отзывов

Цели аналитики:

— Реальные рекомендации пользователям в режиме почти реального времени

— Отчеты и дашборды для аналитиков и маркетинга

3. Выбор компонентов архитектуры

3.1 Слой сбора данных (Ingestion)

Выбор: Apache Kafka

Обоснование:

— позволяет собирать потоковые данные в реальном времени (кликстрим);

— надежная система, поддерживает масштабирование, легко подключается к другим инструментам обработки;

— можно интегрировать с batch-данными через коннекторы (например, для транзакций или CRM).

Альтернатива: Amazon Kinesis — хорош для облака AWS, но Kafka универсальнее и не привязана к конкретному облаку.

3.2 Слой хранения (Storage)

Выбор: MinIO+ Delta Lake

Обоснование:

— MinIO — это аналог S3, который легко разворачивается в России;

— Delta Lake — добавляет управление версиями, транзакции, корректные обновления данных. Поддерживает batch и streaming, что идеально для кликстрима и транзакций;

— подходит для хранения кликстрима, транзакций, отзывов, CRM.

Плюсы: гибко, недорого, не зависит от зарубежных сервисов, совместимо со Spark.

3.3 Слой обработки (Processing)

Выбор: Apache Spark (Databricks)

Обоснование:

— Spark позволяет обрабатывать большие объемы данных быстро;

— поддерживает и batch, и streaming (важно для кликстрима);

— Databricks упрощает работу со Spark: настройка, управление, интеграция с S3/Delta Lake, ML;

Плюсы: мощно, универсально, можно запускать модели машинного обучения.

3.4 Слой аналитики и визуализации (Analytics & Visualization)

Выбор: Metabase

Обоснование:

— простая и понятная платформа для дашбордов и отчетов;

— бесплатная и быстро настраивается;

— подходит для маркетологов и аналитиков, не требует глубоких технических знаний;

Плюсы: экономично, удобно, быстрый старт.

3.5 Слой оркестрации (Orchestration)

Выбор: Apache Airflow

Обоснование:

— управляет потоками данных и расписанием задач (например, обновление моделей, сбор batch-данных);

— популярен, много документации, легко интегрируется с Kafka, Spark и S3.

Плюсы: надежно, гибко, контроль всего процесса.

3.6 Управление данными (Data Governance)

Выбор: Amundsen

Обоснование:

— позволяет следить за качеством и структурой данных, понимать, откуда данные пришли и кто их использует;

— помогает соблюсти стандарты и ускоряет работу команды.

Плюсы: прозрачность, удобство для команды, ускоряет поиск нужных данных.

В таблице 3.1 представлен итоговый стек.

Таблица 3.1. Выбранные компоненты архитектуры

Слой / Назначение	Инструмент	Почему выбран (кратко)
Сбор данных (Ingestion)	Apache Kafka	Реальное время, масштабируемость, коннекторы
Хранилище (Storage)	MinIO + Delta Lake	Локально доступно, версии данных, batch + streaming
Обработка данных	Apache Spark (Databricks)	Большие объёмы, ML, поддержка потоков
Оркестрация	Apache Airflow	Планирование задач, интеграции, автоматизация
Аналитика и визуализация	Metabase	Бесплатно, просто, подходит аналитикам
Управление данными	Amundsen	Каталог данных, поиск, контроль качества

4. Проектирование архитектуры

Распишем поток данных от источников до пользователей.

Источники данных:

— кликстрим (события пользователей на сайте и в приложении);

— транзакции (заказы);

— CRM (профили клиентов, история покупок, уровень лояльности);

— отзывы (текстовые отзывы и оценки товаров).

Сбор данных (Ingestion):

— потоковые данные (кликстрим) поступают в Apache Kafka, который буферизует их и передает дальше;

— пакетные данные (транзакции, CRM, отзывы) загружаются через Apache Airflow по расписанию;

Хранилище данных (Storage):

— все сырые и обработанные данные сохраняются в MinIO;

— над MinIO используется Delta Lake, который управляет версиями данных, объединяет потоковые и пакетные данные, обеспечивает обновление, очистку и хранение в одном формате, готовит данные для аналитики и моделей.

Обработка данных (Processing):

— Apache Spark (Databricks) обрабатывает данные: строит агрегированные таблицы и отчеты, тренирует модели машинного обучения для персональных рекомендаций, обновляет данные в реальном времени и пакетном режиме, обновляет данные в Delta Lake.

Аналитика и визуализация:

— результаты обработки отображаются в Metabase, где маркетологи и аналитики видят дашборды и отчеты;

— маркетинг получает информацию о том, какие рекомендации работают лучше, аналитики контролируют эффективность моделей;

— BI-запросы могут идти к Delta Lake.

Безопасность и управление доступом:

данные шифруются при хранении и передаче:

— TLS для Kafka и MinIO,

— настройки шифрования внутри MinIO;

разграничение доступа проходит через:

— RBAC (роли) в Metabase, Airflow, Spark, MinIO;

— маркетинг видит только дашборды, аналитики — витрины и агрегаты, инженеры — технические слои.

Мониторинг и логирование:

— Prometheus + Grafana следят за состоянием потоков данных, нагрузкой и задержками;

— ELK Stack (Elasticsearch, Logstash, Kibana) собирает и анализирует логи Kafka, Spark и Airflow, MinIO помогает диагностировать ошибки и контролировать процессы.

Стратегия масштабирования и отказоустойчивости:

Масштабирование:

— добавление брокеров Kafka,

— увеличение узлов Spark,

— горизонтальное расширение MinIO.

Отказоустойчивость:

— репликация в Kafka и MinIO,

— версии данных в Delta Lake с возможностью отката,

— повторные перезапуски задач в Airflow и Spark.

5. Создание диаграммы архитектуры

На рисунке 5.1 представлена визуальная схема архитектуры.

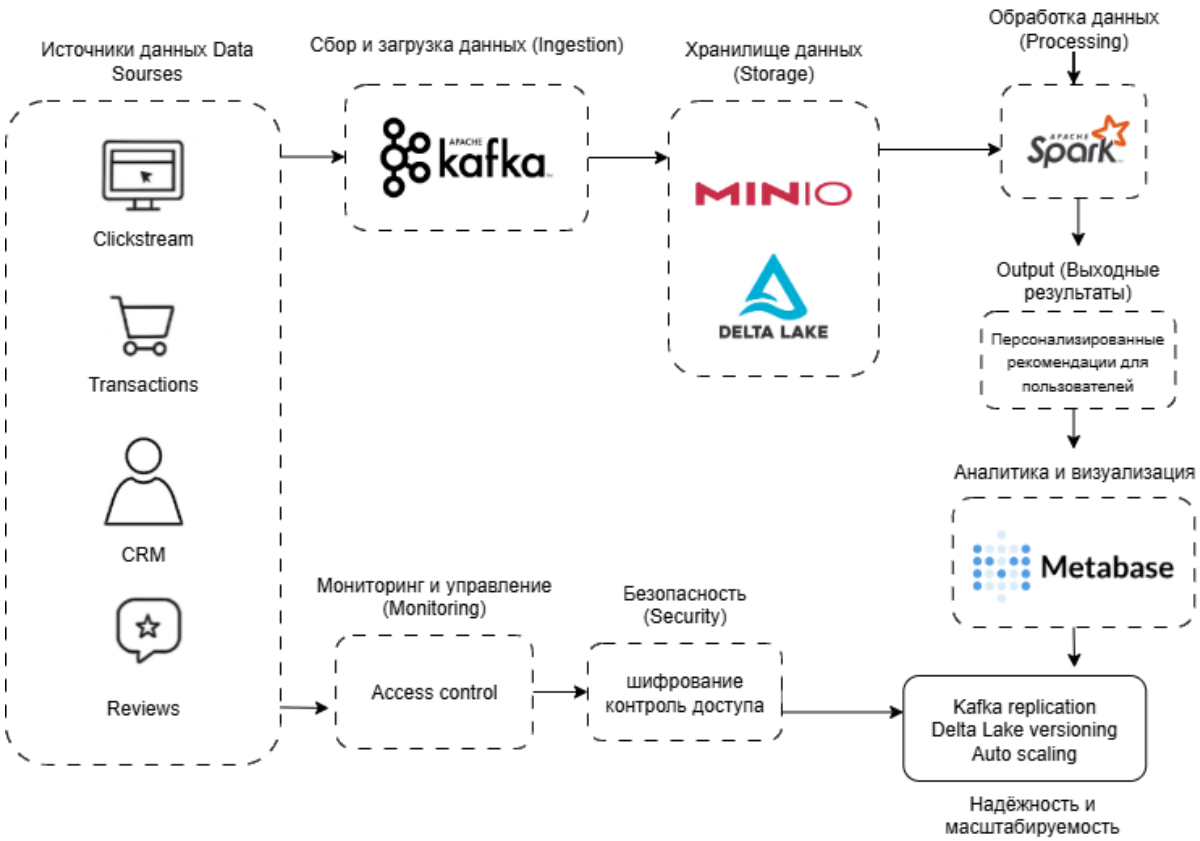

Рисунок 5.1. Диаграмма архитектуры хранилища больших данных онлайн-ритейлера согласно выбранному бизнес-процессу

1) Источники данных (Data Sources)

Система получает данные из четырёх источников:

— Clickstream — действия пользователей на сайте или в приложении;

— Transactions — информация о заказах и покупках;

— CRM — профили клиентов и история взаимодействий;

— Reviews — отзывы и оценки товаров.

2) Сбор данных (Ingestion)

Потоковые данные (clickstream) поступают в Apache Kafka, где временно хранятся и передаются дальше. Это позволяет обрабатывать события пользователей почти в реальном времени.

3) Хранилище данных (Storage)

   Все собранные данные сохраняются в MinIO.

   Поверх него работает Delta Lake, обеспечивая целостность, управление версиями и поддержку как потоковой, так и пакетной обработки.

4) Обработка данных (Processing)

   В Apache Spark (Databricks) данные очищаются, объединяются и анализируются.

   Здесь обучаются и обновляются модели машинного обучения, которые создают персональные рекомендации для пользователей.

5) Output (Выходные результаты)

   Формируются персонализированные рекомендации для пользователей: после обработки данных в Apache Spark (Databricks) и анализа моделей машинного обучения формируется список товаров или акций, подходящих каждому пользователю. Эти рекомендации возвращаются на сайт или в мобильное приложение и отображаются в реальном времени.

6) Аналитика и визуализация (Analytics)

   Обработанные результаты передаются в Metabase, где создаются дашборды и отчёты для аналитиков и менеджеров.

   Этот слой позволяет отслеживать эффективность рекомендаций, конверсии и продажи.

7) Мониторинг и управление (Monitoring)

   Компоненты мониторинга и access control обеспечивают стабильную работу потоков данных и своевременное реагирование на сбои.

8) Безопасность (Security)

   Используется шифрование данных и контроль доступа, чтобы защитить данные и разграничить права пользователей.

9) Масштабируемость и отказоустойчивость

Kafka replication — резервирование потоковых данных;

Delta Lake versioning — возможность отката к предыдущим версиям данных;

Auto scaling — автоматическое масштабирование при росте нагрузки.

6. Описание компонентов и обоснование выбора

Apache Kafka (Ingestion)

Отвечает за сбор потоковых данных (clickstream) в реальном времени.

Выбор: обеспечивает высокую производительность, масштабируемость и надёжность; легко интегрируется с другими инструментами.

MinIO + Delta Lake (Storage)

Хранят все данные — как потоковые, так и пакетные.

Выбор: MinIO — высокопроизводительное и совместимое с S3 объектное хранилище, которое можно развернуть локально или в облаке; Delta Lake добавляет управление версиями, очистку и поддержку batch/stream обработки.

Apache Spark (Databricks) (Processing)

Используется для обработки больших объёмов данных и обучения моделей машинного обучения.

Выбор: мощный и универсальный фреймворк, поддерживает batch и streaming, Databricks упрощает управление и интеграцию.

Metabase (Analytics)

Средство визуализации и построения дашбордов.

Выбор: простое, бесплатное, не требует глубоких технических знаний, подходит аналитикам и маркетологам.

Apache Airflow (Orchestration)

Управляет процессами и расписанием задач.

Выбор: гибкий, легко интегрируется с Kafka, Spark и S3, позволяет контролировать потоки данных.

Amundsen (Data Governance)

Управляет метаданными и качеством данных.

Выбор: повышает прозрачность и ускоряет работу с данными, облегчает поиск и контроль источников.

Security и Monitoring (Encryption, Access Control, Auto Scaling)

Гарантируют безопасность, контроль доступа и устойчивость системы.

Выбор: встроенные механизмы AWS и Databricks обеспечивают защиту данных и масштабируемость без лишней сложности.

7. Анализ потенциальных проблем и их решений

В таблице 7.1 представлены потенциальные проблемы и способы их решения.

Таблица 7.1. Возможные проблемы и их решения

Потенциальная проблема	Описание	Возможное решение
1. Сложность управления потоками данных	При большом объёме событий (clickstream) может быть трудно отслеживать сбои и задержки в Kafka и Spark Streaming.	Использовать Apache Airflow для оркестрации, Prometheus + Grafana для мониторинга, настроить автоматические уведомления о сбоях.
2. Рост объёма хранения данных (MinIO)	При накоплении исторических данных объём MinIO может быстро увеличиваться, что потребует больше ресурсов.	Ввести политику жизненного цикла данных: архивировать старые данные на менее нагруженные ноды или использовать отдельные бакеты для архивов, хранить только актуальные версии в Delta Lake.
3. Обеспечение качества и согласованности данных	Разные источники (CRM, транзакции, отзывы) могут содержать дубликаты или ошибки.	Использовать Delta Lake для версионирования и очистки данных, а также Amundsen для контроля качества и отслеживания происхождения данных.

Вывод: Анализ потенциальных проблем показывает, что основными рисками при работе с потоковыми и пакетными данными являются сложность управления потоками, рост объёма хранения и обеспечение качества данных. Для их решения предлагаются проверенные инструменты и

подходы: оркестрация через Apache Airflow и мониторинг с Prometheus + Grafana позволяют оперативно отслеживать сбои и задержки; управление жизненным циклом данных в MinIO и Delta Lake обеспечивает контроль объёма хранения и версионирование; а интеграция с Amundsen помогает поддерживать высокое качество и согласованность данных. Таким образом, комплексное применение этих решений позволяет повысить надёжность, масштабируемость и управляемость всей аналитической системы.