Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Инструменты для хранения и обработки больших данных

**Лабораторная работа №3.1**

**Тема:**

«Проектирование архитектуры хранилища больших

данных»

Выполнил: Егоров А. С., группа: АДЭУ-211

Преподаватель: Босенко Т. М.

Москва

2024

СОДЕРЖАНИЕ

[ХОД РАБОТЫ 3](#_Toc183122373)

[Шаг 1. Определение требований 3](#_Toc183122374)

[Шаг 2. Выбор модели хранилища данных 3](#_Toc183122375)

[Шаг 3. Схема архитектура хранилища больших данных 6](#_Toc183122376)

[Шаг 4. Процесс обработки данных 7](#_Toc183122377)

[Шаг 5. Масштабирование и отказоустойчивость 8](#_Toc183122378)

[Шаг 6. Безопасность 8](#_Toc183122379)

[ЗАКЛЮЧЕНИЕ 9](#_Toc183122380)

# ХОД РАБОТЫ

Цель работы: разработать архитектуру хранилища больших данных для заданного сценария использования.

**Работа выполнена по варианту 4, соответствующего порядковому номеру в списке группы.**

## Шаг 1. Определение требований

**Средняя логистическая компания.**

1.1. Объем данных:

* Ожидаемый объем: 50 ТБ в год;
* Рост: 25% ежегодно.
  1. . Скорость получения данных до 500 событий в секунду.
  2. . Типы данных:
* Структурированные: 70%;
* Полуструктурированные: 20%;
* Неструктурированные: 10%.
  1. . Требования к обработке:
* Оптимизация маршрутов в реальном времени;
* Прогнозирование спроса на услуги.

1.5. Доступность данных:

* Время отклика: <30 секунд;
* Доступность системы: 99.9%.

1.6 Безопасность данных

* Базовое шифрование;
* Соответствие требованиям 152-ФЗ «О персональных данных».

## Шаг 2. Выбор модели хранилища данных

Для средней логистической компании с нашим типом данных подойдет модель Hybrid Data Storage, которая объединяет преимущества Data Lake и Data Warehouse, предоставляя возможность хранить разнообразные типы данных (структурированные, полуструктурированные, неструктурированные) и обеспечивая требуемый уровень доступности, безопасности и масштабируемости. Lakehouse поддерживает как аналитику в режиме реального времени, так и сложные аналитические запросы.

Была построена схема «Звезда». Ежегоднвй прирост данных 25% и до 500 событий в секунду позволяют ей быстрее работать, чем схеме «Снежинка».

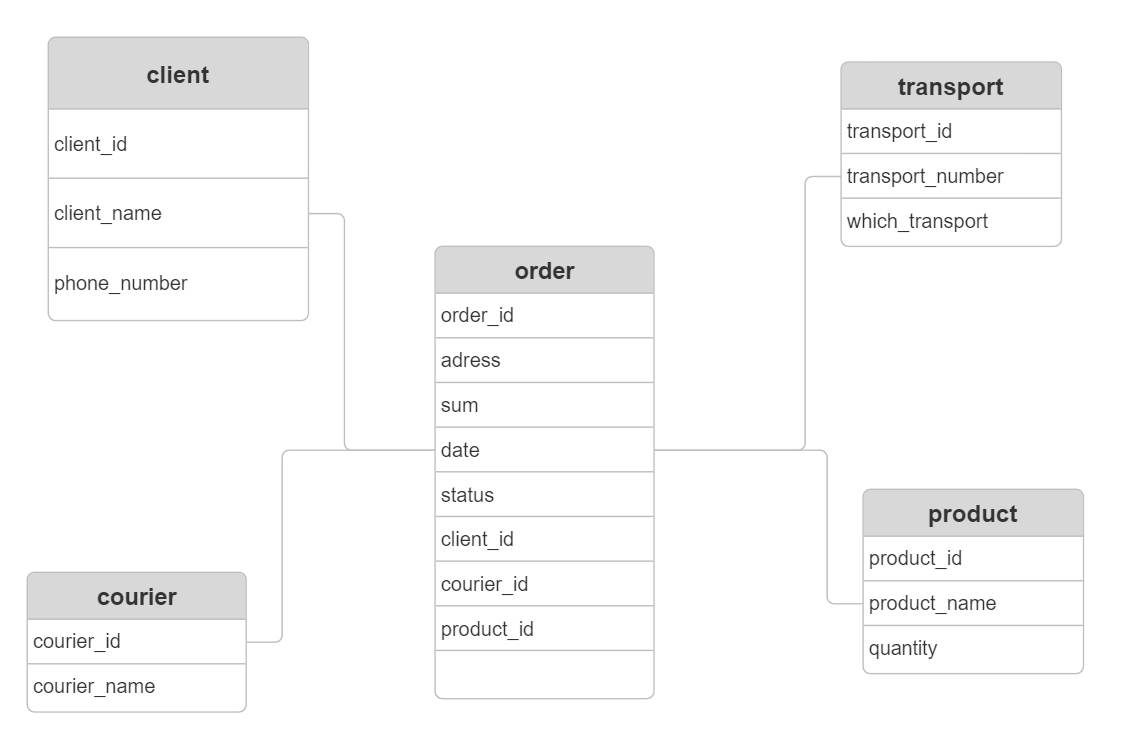


Рисунок 2 – Схема «Звезда»

Источники данных:

* Социальные сети
* Веб-сайт
* Данные из систем учета и CRM

Слой сбора данных

* Logstash - программное обеспечение для сбора, хранения и обработки данных из журналов в разных системах. Он является частью экосистемы Elastic Stack (ELK), который включает в себя Elasticsearch, Kibana, Beats.
* Apache Kafka – система централизованного сбора, обработки, безопасного хранения и передачи большого количества данных.

Слой хранения данных.

* MongoDB - документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных.
* PostgreSQL — это объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом. Она помогает создавать и хранить базы данных, а также работать с ними на языке SQL.
* HDFS (Hadoop Distributed File System) - распределённая файловая система Hadoop для хранения файлов больших размеров с возможностью потокового доступа к информации, поблочно распределённой по узлам вычислительного кластера.

Слой обработки данных

* Apache Spark - фреймворк с открытым исходным кодом для реализации распределённой обработки данных в режиме реального времени.
* Apache Flink - фреймворк с открытым исходным кодом для реализации обработки потоков.

Слой аналитики и машинного обучения:

* Python в Jupyter Notebook.
* Apache Mahout - предоставляет бесплатные реализации распределённых алгоритмов машинного обучения для построения моделей прогнозирования.
* Power BI.

Слой управления данными

* Apache Atlas - сервисов для управления метаданными и большими данными.
* Apache Falcon - фреймворк по обработке и управлению данными в Hadoop, в том числе для координации данных и управления их жизненным циклом.

Слой оркестрации и мониторинга

* Prometheus – система мониторинга и оповещения.
* Grafana – программная система визуализации данных на основе мониторинга.

## Шаг 3. Схема архитектура хранилища больших данных

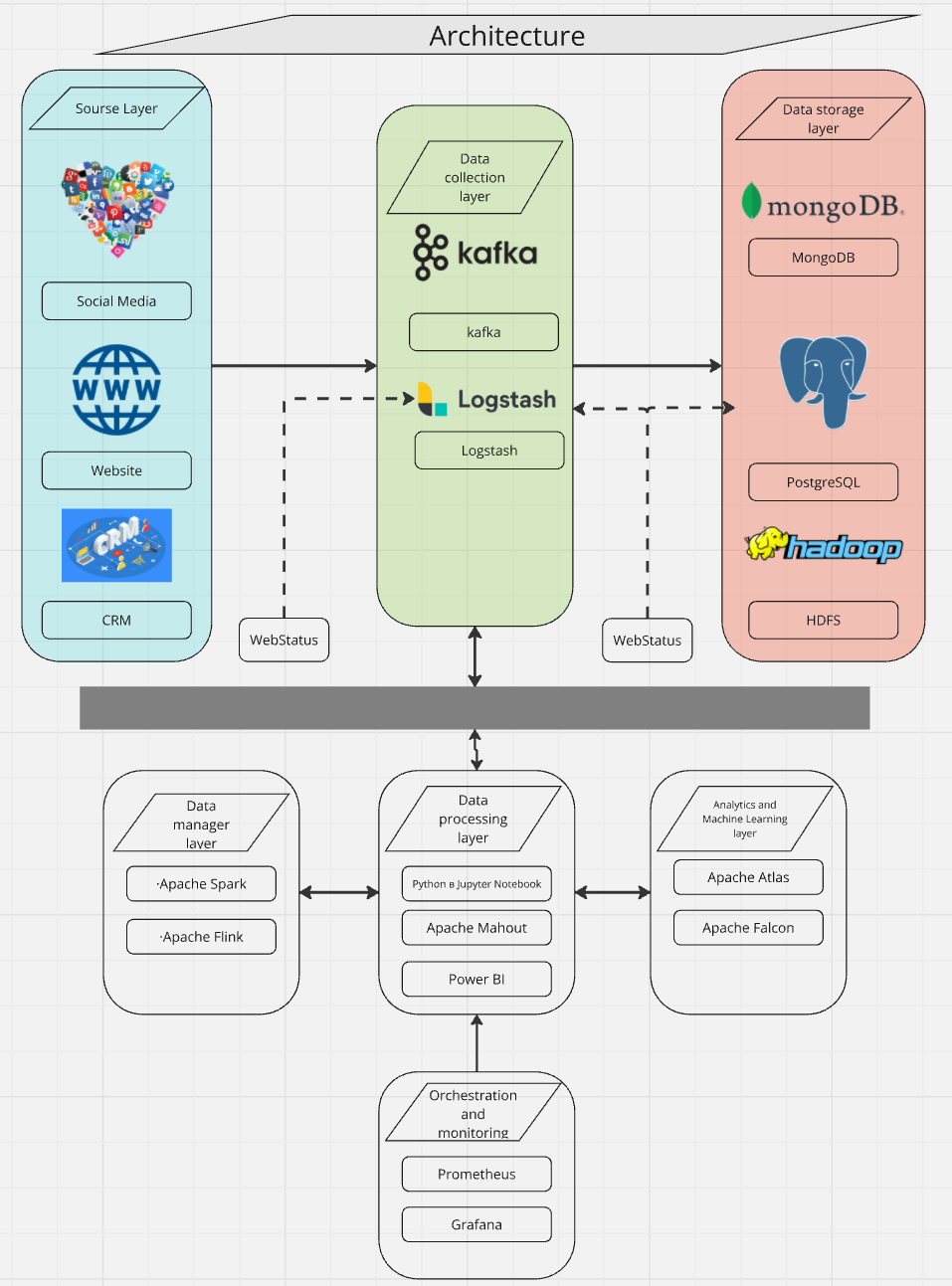


Рисунок 2 – Схема архитектуры

## Шаг 4. Процесс обработки данных

* Данные поступают из множества источников через слой сбора информации. Социальные сети для неструктурированных данных, веб-сайты и мобильные приложения для полуструктурированных данных, и CRM системы для структурированных данных.
* Сырые данные хранятся в HDFS для долговременного хранения.
* Потоковые данные обрабатываются в реальном времени при помощи Apache Spark для оперативной аналитики.
* Для анализа и формирования отчетов применяют Jupyter Notebooks и Power BI.
* Модели машинного обучения обучаются на исторических данных и используются для предсказаний и рекомендаций.

## Шаг 5. Масштабирование и отказоустойчивость

* Использование HDFS для горизонтального масштабирования. При увеличении объема данных или нагрузки просто добавляются дополнительные серверы к кластеру.
* Дублирование данных в HDFS и PostgreSQL для обеспечения отказоустойчивости. Каждый блок данных в HDFS имеет несколько копий на различных узлах. В случае выхода из строя одного узла информацию можно восстановить из других.
* Использование Prometheus для наблюдения и мониторинга, а также Grafana для визуализации результатов мониторинга.

## Шаг 6. Безопасность

* Делать постоянно резервные копии, чтобы можно было быстро восстановить данные, откатившись назад.
* Использовать двухфакторную аутентификацию.
* Шифровать данные.

# ЗАКЛЮЧЕНИЕ

Таким образом, архитектура хранилища данных работает со структурированными, полуструктурированными и неструктурированными данными и приспособлена для растущего объема данных.

Получена легко масштабируемая и отказоустойчивая архитектура, обрабатывающая большие объемы данных в режиме реального времени.

В нашем случае для хранения данных подходит «Звезда», так как небольшой прирост данных и не так много событий поступает в секунду. «Звезда» будет работать быстрее «Снежинки», на создание которой требуется больше времени и ресурсов.