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(а): Морозова Валерия АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2024

Вариант 10.

Средняя энергетическая компания

1. Определение требований

1.1 Объем данных: 120 ТБ в год, рост 25% ежегодно.

1.2 Скорость получения: до 1500 событий в секунду.

1.3 Типы данных: 70% структурированные, 25% полуструктурированные, 5% неструктурированные.

1.4 Требования к обработке: мониторинг энергопотребления в реальном времени, прогнозирование нагрузки на сеть.

1.5 Доступность: 99.99%, время отклика < 5 секунд

1.6 Безопасность данных

- Двухфакторная аутентификация для доступа к данным.
- Соответствие требованиям 152-ФЗ "О персональных данных".

2. Выбор модели хранилища данных

Для структурированных данных я бы выбрала объектно-реляционную базу данных (ORDBMS) - Microsoft SQL Server.

Для полуструктурированных выбрала бы графовую базу данных - Neo4j, так как она эффективна для анализа отношений данных, исходя из чего можно получить ценную информацию.

Для неструктурированных выбрала бы хранилище данных в виде ключ-значение - Redis

Или же в качестве альтернативы, вместо трех инструментов взять один OrientDB.

OrientDB — это система управления базами данных NoSQL с открытым исходным кодом, написанная на Java**.**

Это многофункциональная база данных, которая поддерживает:

- графовую модель;
- документы;

- ключ-значение;
- объектно-ориентированные данные.

OrientDB использует гибридный подход, объединяя графовую модель с документами. Она может создавать вершины и рёбра, а также хранить данные в формате документов.

OrientDB обеспечивает:

+ гибкость моделирования;
+ скорость запросов;
+ масштабируемость (горизонтальное масштабирование и репликацию)

Выбор схемы «Звезда», так как в среднем объем данных не сильно большой и его прирост составляет 25%, а скорость до 1500 событий в секунду, модель «Звезда» работает быстрее нежели «Снежинка». Кроме того схема "Снежинка" сложнее ввиду транзитивности, и несмотря на то, что она надежнее, требует больше времени и ресурсов для разработки и поддержки.

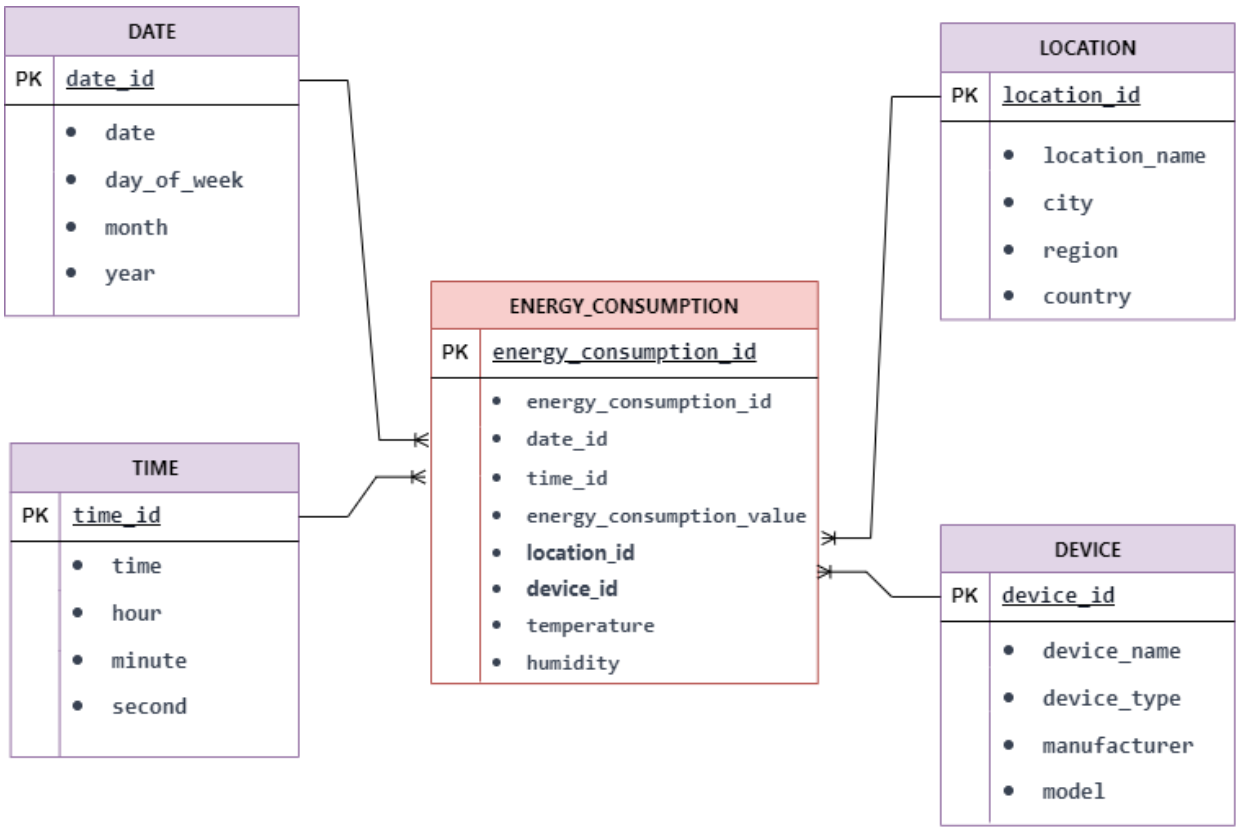

Рисунок 1. Схема «Звезда»

3. Архитектура хранилища больших данных

Слой источника данных

- Сенсоры и измерительные устройства, собирающие данные о энергопотреблении и нагрузке на сеть.
- SCADA-системы, передающие данные в центр управления.
- Логические контроллеры, собирающие данные о состоянии оборудования и энергопотреблении.
- Другие источники данных, такие как погодные станции, системы мониторинга энергопотребления и т.п.

Слой сбора данных

- Logstash для преобразования данных в желаемый формат (горизонтально масштабируемый конвейер обработки данных с поддержкой Elasticsearch и Kibana).
- Apache Kafka система обработки потоков данных для обработки высокоскоростных потоков данных.

Слой хранения данных

- NoSQL-база данных MongoDB, для хранения полуструктурированных данных, таких как данные о состоянии оборудования и энергопотреблении.
- Реляционная база данных PostgreSQL, для хранения структурированных данных, таких как данные о клиентах и счетах.
- Объектное хранилище HDFS (Hadoop Distributed File System) — это распределённая файловая система, предназначенная для хранения больших массивов данных в рамках кластера из нескольких узлов.

Слой обработки данных

- Apache Spark для обработки и анализа данных в реальном времени.
- Apache Flink для обработки высокоскоростных потоков данных.

Слой аналитики и машинного обучения

- Python для анализа данных и построения моделей в Jupyter Notebook.

- Apache Mahout для построения моделей прогнозирования нагрузки на сеть.
- Power BI для представления результатов анализа.

Слой управления данными

- Apache Atlas для управления метаданными и обеспечения их качества.
- Apache Falcon для обеспечения управления данными и их обработки.

Слой оркестрации и мониторинга

- Prometheus для обеспечения мониторинга и оповещения о проблемах.
- Grafana для представления результатов мониторинга.

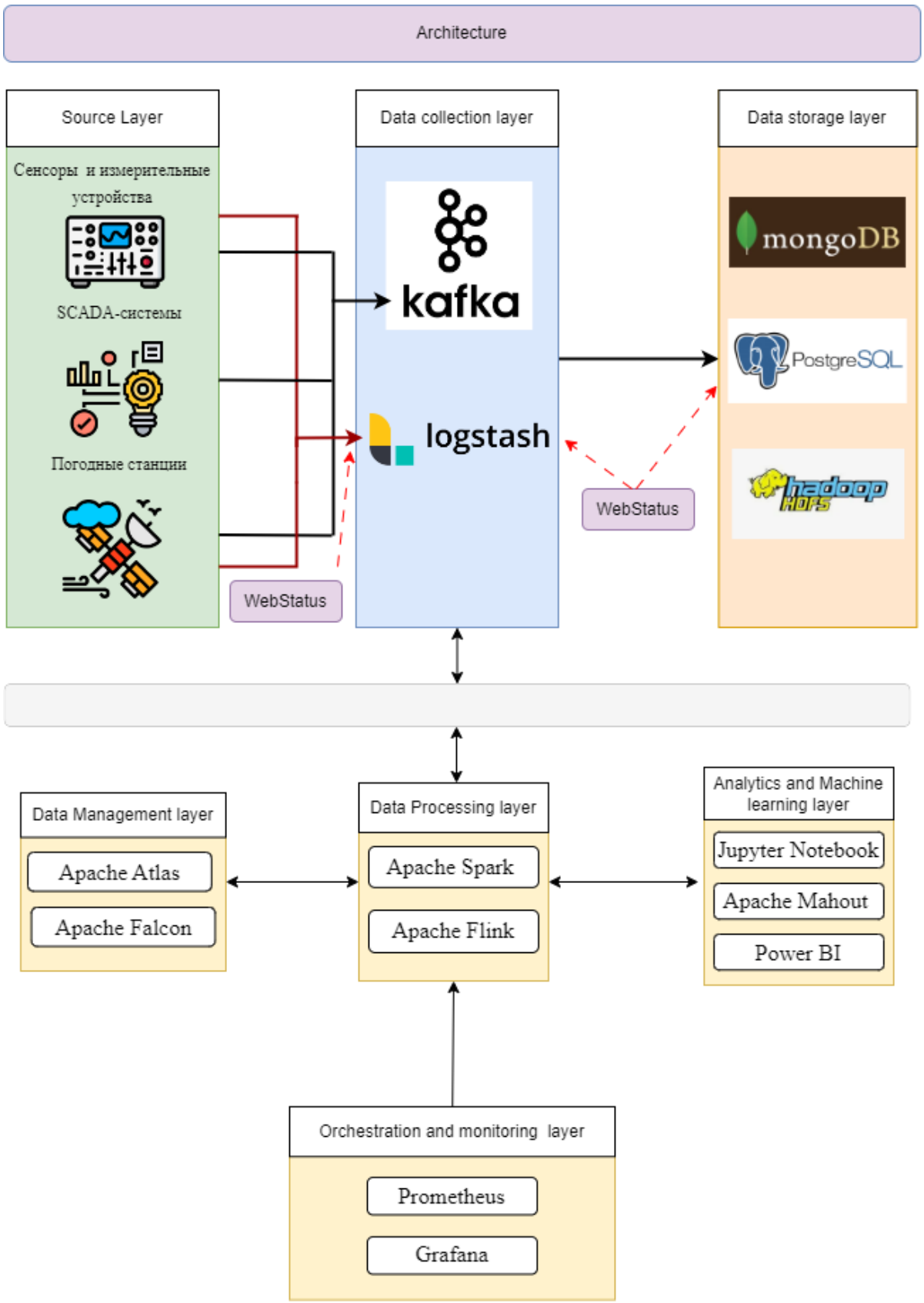

Рисунок 2. Архитектура хранилища больших данных

4. Процесс обработки данных

– Данные собираются из различных источников через слой сбора данных.

– Сырые данные сохраняются в HDFS для долгосрочного хранения.

– Потоковые данные обрабатываются в реальном времени с помощью Apache Spark для быстрой аналитики.

– Аналитики используют Jupyter Notebooks и Power BI для исследования данных и создания отчетов.

– Модели машинного обучения обучаются на исторических данных и развертываются для прогнозирования и рекомендаций.

– Для построения моделей прогнозирования нагрузки на сеть используют Apache Mahout.

5. Масштабирование и отказоустойчивость

- Использование HDFS для легкого масштабирования по горизонтали. Если объём данных или нагрузка увеличиваются, то можно просто добавить больше серверов в вычислительный кластер.
- Репликация данных в HDFS и PostgreSQL для обеспечения отказоустойчивости. Каждый блок данных в HDFS дублируется на несколько узлов. Если один узел выходит из строя, информация может быть восстановлена из других.
- Применение Prometheus для обеспечения мониторинга и оповещения о проблемах и Grafana для представления результатов мониторинга.

6. Безопасность

- ✓ Двухфакторная аутентификация для доступа к данным.
- ✓ Регулярное резервное копирование и план аварийного восстановления.

Выводы

Согласно изначальным требованиям средней энергетической компании, были выполнены все поставленные задачи, и в качестве результата получена масштабируемая и отказоустойчивая архитектура хранения данных, которая позволяет обработать большие объемы данных в реальном времени, что необходимо для мониторинга энергопотребления и прогнозирования нагрузки на сеть.

Не смотря на гарантированную надежность благодаря транзитивности схемы «Снежинка», выбор был сделан в сторону модели «Звезда», так как она больше подходит для быстрого выполнения запросов.

Как итог личного опыта, мною были изучены новые инструменты для сбора, хранения и управления данными.