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

Проектирование архитектуры хранилища данных

Выполнил(а): Ванярина Ю.А., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2024

Цель работы:

Обеспечить надежное хранение, эффективную обработку и анализ больших объемов данных, получаемых из различных источников, таких как датчики дорожного движения, температур, дыма и др.

Задача:

Создать архитектуру хранилища больших данных для компании в сфере «умного города»

Вариант 16

Задача: создать архитектуру хранилища больших данных для Средняя компания в сфере "умного города"

- Объем данных: 150 ТБ в год, рост 55% ежегодно
- Скорость получения: до 3000 событий в секунду
- Типы данных: 45% структурированные, 45% полуструктурированные, 10% неструктурированные
- Требования к обработке: управление городской инфраструктурой в реальном времени, анализ транспортных потоков
- Доступность: 99.99%, время отклика <5 секунд

Безопасность: продвинутое шифрование, строгое соответствие 152-ФЗ и требованиям к критической инфраструктуре компании, занимающейся сферой “умного города”.

От умного города получают разные типы данных:

- **Традиционные данные наблюдения**, измеряемые статически в городских пространствах. Например, придорожные датчики с индуктивными петлями, определяющие интенсивность движения.
- **Динамические городские данные**, собираемые в реальном времени мобильными агентами, перемещающимися по городу.

Также к данным умного города относятся **сведения, собранные от граждан, устройств, зданий и активов**. Они обрабатываются и анализируются для мониторинга и управления дорожным движением и транспортными системами, электростанциями, коммунальными услугами,

сетями водоснабжения, отходами, раскрытием преступлений, информационными системами, школами, библиотеками, больницами и другими общественными службами.

Шаг 1.

1.1 Объём данных

- 150 ТБ в год, рост 55% ежегодно

1.2 Скорость получения данных:

- до 3000 событий в секунду

1.3 Типы данных

- 45% структурированные, 45% полуструктурированные, 10% неструктурированные

Требования к обработке:

- управление городской инфраструктурой в реальном времени, анализ транспортных потоков

1.5 Доступность данных:

- 99.99%, время отклика <5 секунд

1.6 Безопасность данных:

- продвинутое шифрование, строгое соответствие 152-ФЗи требованиям к критической инфраструктуре

Шаг 2. Выбор модели хранилища данных

2. Архитектура хранилища больших данных

2.1 Компоненты архитектуры:

Источники данных для умного города:

- Структурированные данные:

1. Данные транспортных систем: расписания общественного транспорта, данные по пробкам, статистика использования городских парковок.

2. Здоровье и социальные услуги: реестр пациентов, статистика обращений за медицинской помощью, данные поликлиник.

3. Финансовые и экономические данные: бюджет города, расходы на различные программы, налоговые сборы.

4. Экологические показатели: уровень выбросов, данные с метеорологических станций (температура, влажность), уровень загрязнений.

5. Данные о ЖКХ: потребление воды, газа, электричества, счетчики в домах.

- Полуструктурированные данные

1. Данные с сенсоров и IoT-устройств: информация от камер наблюдения, датчиков освещения, контроля температуры и влажности.

2. Логи событий и активности: например, логи транспорта, события на светофорах, данные об инцидентах на дороге.

3. Файлы конфигурации и настройки: JSON или XML-файлы с настройками различных городских систем (например, уличного освещения).

4. Данные с соцсетей и городских приложений: сообщения от жителей города с тегами и метаинформацией (время, место).

- Неструктурированные данные

1. Фотографии и видео: данные с камер наблюдения, уличных камер, видеорегистраторов.

2. Тексты и документы: отчеты, электронные письма, письма жителей с жалобами или предложениями.

3. Записи разговоров: например, обращения жителей в службу 112 или другие горячие линии.

4. Данные соцсетей и новостных ресурсов: посты, комментарии, видеообращения, текстовые сообщения, публикации с упоминанием города или инцидентов.

Слой сбора данных:

1. Logstash для сбора логов.

2. Cisco Kinetic обеспечивает сбор данных, отправляя их в облако для дальнейшей обработки

3. Пользовательские коннекторы для CRM.

Слой хранения данных

1. Hadoop HDFS подходит для хранения больших объемов данных в исходном формате.

2. Apache Cassandra обеспечивают высокую производительность и масштабируемость для хранения полуструктурированных данных (данные IoT в JSON-формате).

3. PostgreSQL подходит для хранения структурированных данных, таких как метаданные и учетные записи

Слой управления данными

Apache Ranger для контроля доступа и аудита.

Слой обработки данных

1. Apache Kafka — решение для потоковой обработки данных, которое поддерживает высокую пропускную способность и надежность.

2. Apache Flink способен обрабатывать большие объемы событий с большой скоростью.

3. Google Cloud Dataflow - полностью управляемый сервис потоковой обработки, который упрощает разработку и развертывание конвейеров обработки данных для анализа и управления данными умного дома.

Слой аналитики и машинного обучения

1. Google BigQuery обеспечивает быструю обработку больших объемов данных, поддерживая сложные SQL-запросы и аналитику.

2. TensorFlow для работы с большими массивами данных.

3. Tableau помогает создавать интерактивные дашборды

Слой оркестрации и мониторинга

1. Kubernetes управляет контейнеризированными приложениями и их масштабированием

2. Prometheus широко используется для мониторинга метрик в реальном времени и визуализации данных.

Для хранения данных подойдет больше всего «Звезда», она изображена на рисунке 1.

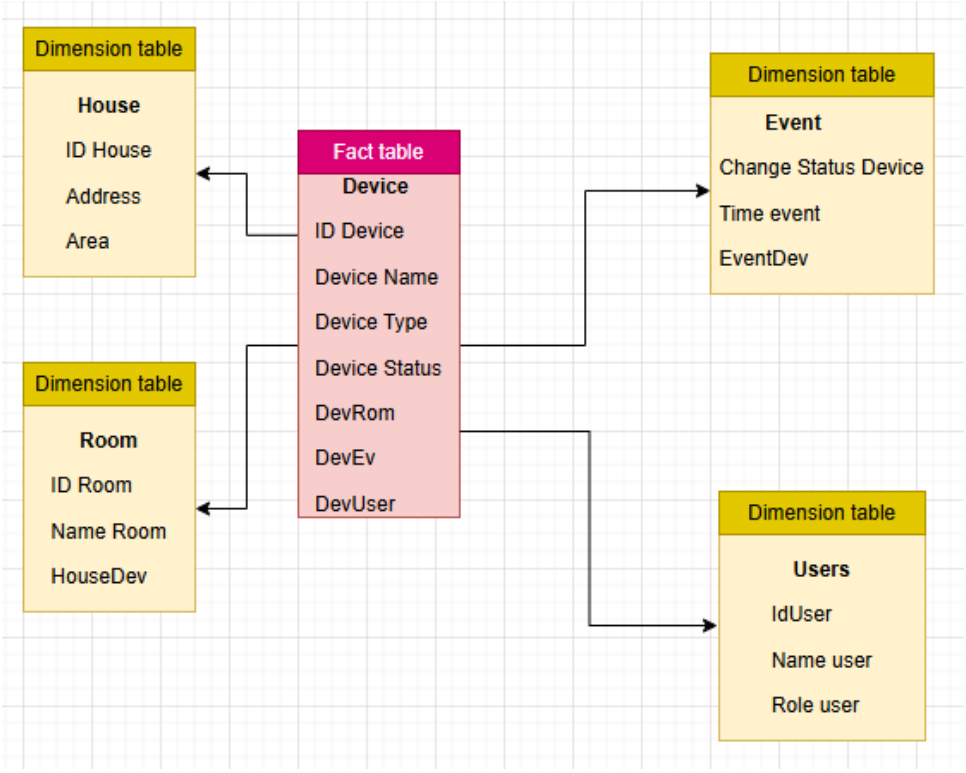

Рисунок 1. Схема хранения данных «Звезда»

OLAP и OLTP

- OLAP (многомерная аналитическая обработка)

Сильные стороны:

1. Подходит для анализа больших объемов структурированных данных.

2. Обеспечивает быструю агрегацию и выявление закономерностей.

Слабые стороны:

1. Менее эффективен для обработки неструктурированных и полуструктурированных данных.

2. Может требовать предварительной агрегации данных.

- OLTP (обработка транзакций в режиме реального времени)

Сильные стороны:

1. Оптимизирован для обработки большого количества событий в режиме реального времени.

2. Поддерживает как структурированные, так и неструктурированные данные.

Слабые стороны:

1. Менее подходит для анализа больших объемов данных.

2. Может не обеспечить такой же уровень агрегации, как OLAP.

Вывод: OLTP является более подходящим вариантом, так как она больше подходит для большого объема данных, высокой скорость получения, наличия различных типов данных

Общая схема архитектуры изображена на рисунке 2.

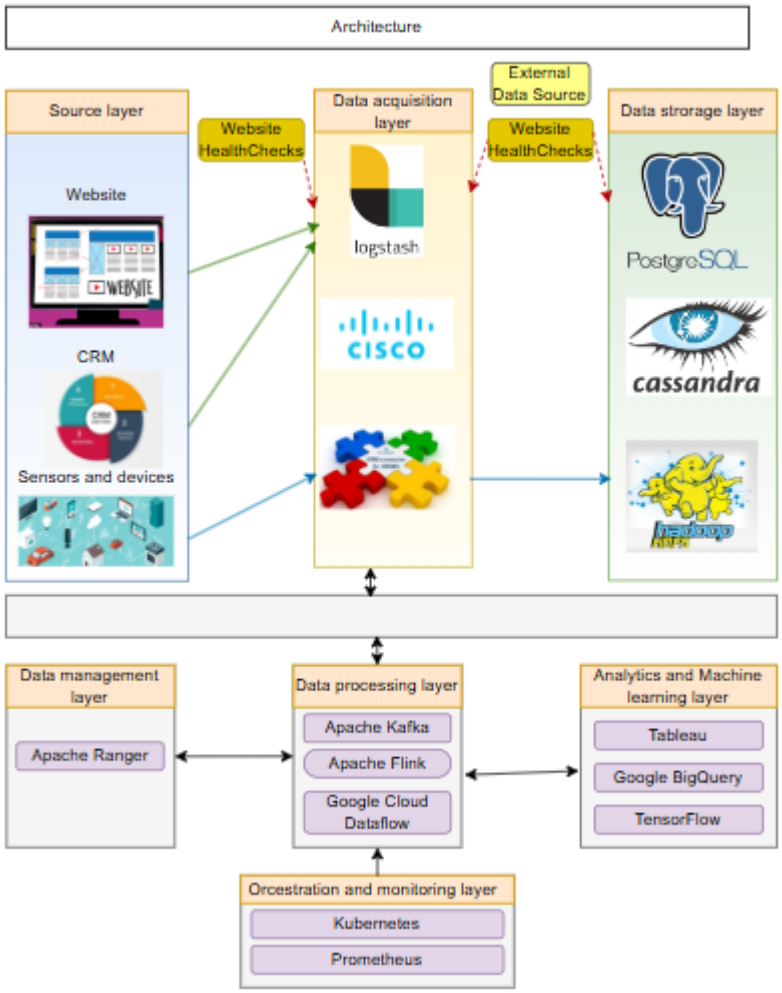

Рисунок 2. Общая схема архитектуры

Процесс обработки данных

1. Структурированные данные преобразуются, агрегируются и нормализуются, полуструктурированные данные, такие как XML и JSON, преобразуются в структурированный формат, неструктурированные данные, такие как текст и изображения, извлекаются и обрабатываются.

2. Пакетные задачи выполняются с помощью Spark

3. Для визуализации используется Tableau

Масштабируемость и отказоустойчивость

1. Apache Flink способен обрабатывать большие объемы событий с большой скоростью

2. Apache Spark – это быстрый кластерный движок для обработки данных в памяти, предлагающий отказоустойчивость и высокую производительность.

3. Apache Kafka – это потоковый движок сообщений для обработки больших объемов данных в режиме реального времени с высокой пропускной способностью и низкой задержкой.

Безопасность

1. Cloudera Manager (CM): Платформа управления, обеспечивающая безопасность, аудит и контроль доступа для всех компонентов кластера Hadoop

2. Kubernetes – безопасность путем реализации механизмов аутентификации, авторизации

Выводы:

1. Для схемы хранения данных была выбрана «Звезда»;

2. OLTP для обработки большого количества данных;

3. Архитектура, разработанная ранее, позволит работать с неструктурированными, структурированными, полуструктурированными данными, а также обеспечивает безопасность, хранение и обработку (то есть управление городской инфраструктурой в реальном времени, анализ транспортных потоков).