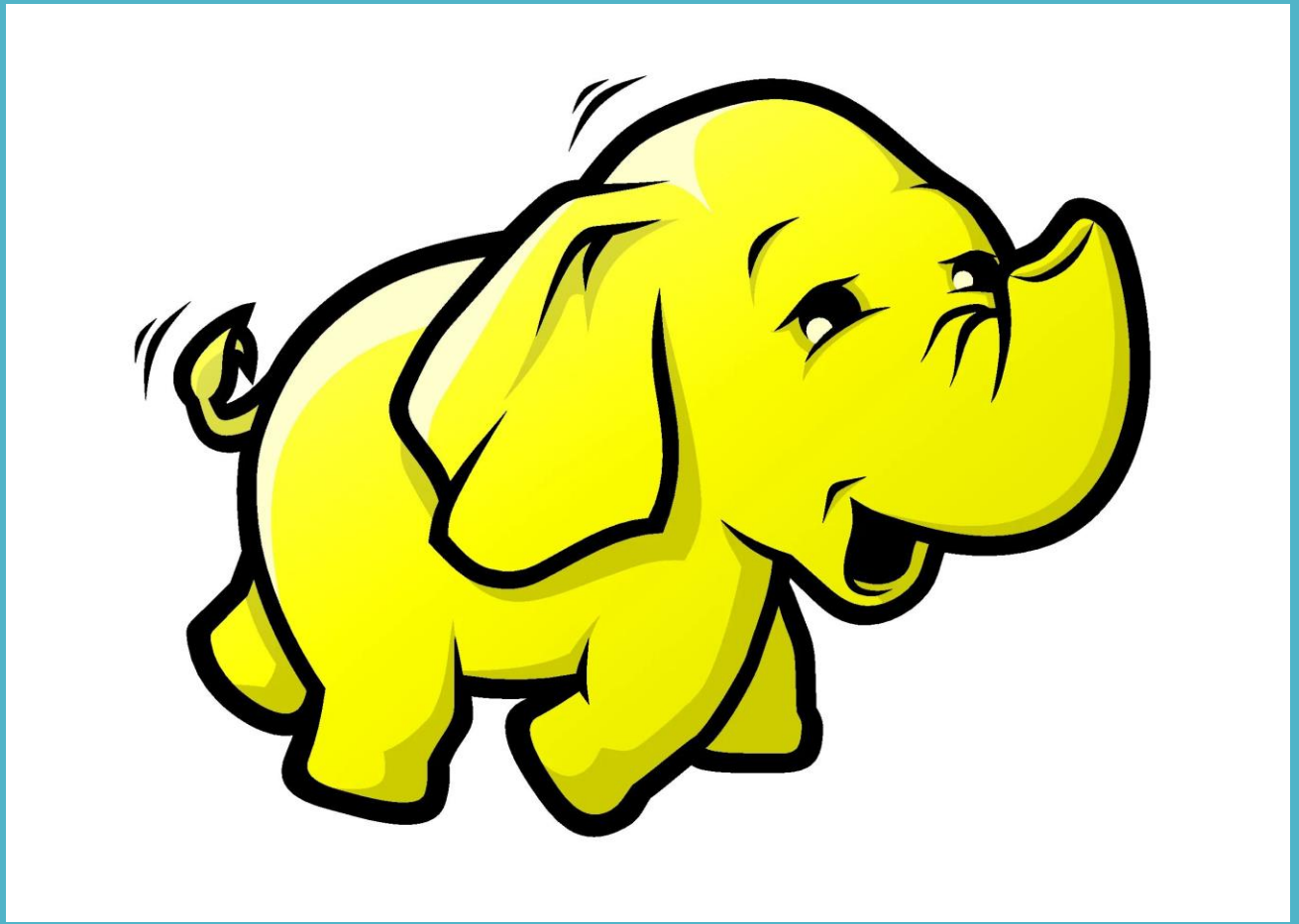
Тема 18. Введение в Big Data и Hadoop.



Цель занятия:

Изучить различные подходы к интеграции MongoDB с приложениями.

Учебные вопросы:

1. Введение в большие данные (Big Data)
2. Основные концепции обработки больших данных
3. Введение в Hadoop
4. Компоненты экосистемы Hadoop
5. Применение Hadoop в реальном мире

1. Введение в большие данные (Big Data)

Большие данные (Big Data) — это термин, который описывает огромные объемы данных, которые слишком велики или сложны для обработки традиционными методами и инструментами.

Эти данные могут поступать из различных источников и имеют разные форматы, включая структурированные, полуструктурированные и неструктурированные данные.

Основные характеристики больших данных:

- Объем (Volume). Большие объемы данных, которые могут достигать терабайтов и петабайтов. Эти данные могут поступать от различных источников, таких как сенсоры, устройства IoT, социальные сети, транзакционные системы и многое другое.
- Скорость (Velocity). Быстрота, с которой данные создаются и обрабатываются. В современном мире данные поступают в реальном времени или почти в реальном времени, что требует мгновенной обработки и анализа.
- Разнообразие (Variety). Разнообразные форматы данных, включая текст, изображения, видео, аудио, а также структурированные (например, базы данных) и неструктурированные (например, текстовые документы) данные.

Применение больших данных:

- Аналитика в бизнесе: Использование данных для оптимизации процессов, улучшения клиентского обслуживания и предсказания поведения потребителей.
- Медицинские исследования: Анализ больших объемов данных для выявления паттернов заболеваний и улучшения лечения.
- Финансовый сектор: Выявление мошенничества и оценка кредитных рисков.
- Управление ресурсами: Оптимизация логистики и управления цепочками поставок.

Big Data: области применения

Классификация данных

Данные можно классифицировать на три основных типа:

- структурированные
- полуструктурированные
- неструктурированные

Эта классификация основана на том, как данные организованы и хранятся, а также на уровне их обработки.

1. Структурированные данные

Структурированные данные имеют четко определенную структуру, что позволяет легко их организовать, хранить и обрабатывать с использованием реляционных баз данных. Они обычно представлены в виде таблиц с фиксированными полями.

Примеры:

Реляционные базы данных: данные в таблицах, например, базы данных клиентов с такими полями, как имя, адрес, номер телефона и т.д.

CSV и Excel файлы: данные, организованные в строки и столбцы.

Форматы JSON и XML с фиксированной схемой.

2. Полуструктурированные данные

Полуструктурированные данные не имеют строгой структуры, но содержат теги или другие метаданные, которые упрощают их организацию. Они могут быть менее организованными, чем структурированные данные, но всё равно позволяют некоторую степень анализа.

Примеры:

JSON и XML: документы, содержащие данные с метаданными, которые описывают структуру.

Логи и события: записи с переменными полями, которые могут изменяться от одной записи к другой.

Email: сообщения, которые могут содержать текст, изображения и вложения, но имеют определённые метаданные (например, заголовки).

3. Неструктурированные данные

Неструктурированные данные не имеют фиксированной структуры и сложно поддаются количественному анализу. Они могут занимать значительные объемы и требуют сложных методов обработки для извлечения полезной информации.

Примеры:

Текстовые документы: статьи, блоги, отчеты, которые содержат много информации, но не имеют четкой структуры.

Мультимедиа: изображения, видео и аудио файлы, которые не могут быть организованы в таблицы.

Социальные сети: посты, комментарии, фотографии и видео, которые могут содержать разнообразный контент.

Примеры источников данных:

Социальные сети:

- Данные пользователей: посты, комментарии, лайки и взаимодействия.
- Аналитика: данные о взаимодействии с контентом и пользователями.

IoT (Интернет вещей):

- Сенсоры и устройства: данные, собранные с различных устройств, таких как умные термометры, камеры, автомобили и т.д.
- Потоковые данные: постоянный поток информации о состоянии и активности устройств.

Транзакционные системы:

- Финансовые транзакции: данные о покупках, продажах и других финансовых операциях.
- ERP и CRM системы: данные о клиентах, продажах, запасах и других бизнес-процессах.

Проблемы работы с большими данными

Хранение:

- Огромные объемы данных требуют значительных ресурсов.
- Необходимость распределённых систем хранения.
- Обеспечение надёжности и доступности данных.

Обработка:

- Недостаточная скорость традиционных методов обработки.
- Сложности с параллельной обработкой.
- Поддержка разнообразных форматов данных.

Анализ:

- Сложность извлечения полезной информации из больших объемов данных.
- Выбор подходящих инструментов и технологий.
- Необходимость специальных навыков в анализе и машинном обучении.

Визуализация:

- Проблемы с отображением больших объемов информации.
- Выбор эффективных методов визуализации.
- Создание интерактивных визуализаций требует дополнительных усилий.

Трудности с реляционными базами данных:

- Ограниченная масштабируемость и фиксированная схема.
- Низкая производительность при сложных запросах.
- Проблемы с транзакционной целостностью.

2. Основные концепции обработки больших данных

- Распределённые системы хранения данных: Использование распределённых файловых систем, таких как **HDFS** (Hadoop Distributed File System), для хранения больших объемов данных на множестве узлов. Это позволяет обеспечить масштабируемость и отказоустойчивость.
- Параллельная обработка: Обработка данных в параллельном режиме на нескольких узлах для повышения скорости и эффективности. Это может быть реализовано с помощью технологий, таких как MapReduce и Apache Spark.

- Потоковая обработка: Обработка данных в режиме реального времени по мере их поступления. Используются фреймворки, такие как Apache Kafka и Apache Flink, которые позволяют работать с потоками данных и реагировать на события мгновенно.
- Хранилища данных и базы данных NoSQL: Использование NoSQL баз данных (например, MongoDB, Cassandra) для работы с полуструктурированными и неструктурированными данными. Эти базы данных предлагают большую гибкость и масштабируемость по сравнению с традиционными реляционными системами.
- Аналитика и машинное обучение: Применение алгоритмов машинного обучения и статистического анализа для извлечения инсайтов из больших данных. Инструменты, такие как Apache Spark MLlib, позволяют реализовывать аналитические задачи на больших объемах данных.
- Визуализация данных: Применение инструментов для визуализации больших данных (например, Tableau, Power BI) для представления результатов анализа в понятном и доступном формате. Это помогает пользователям быстрее принимать решения на основе данных.

- Метаданные и управление данными: Использование метаданных для описания структуры и контекста данных, что облегчает их поиск и управление. Эффективное управление данными включает в себя задачи по очистке, интеграции и трансформации данных.
- Обеспечение безопасности и конфиденциальности: Реализация мер по защите данных и соблюдению нормативных требований, таких как GDPR. Это включает в себя шифрование, управление доступом и аудит.
- Облачные технологии: Использование облачных платформ (например, Amazon Web Services, Google Cloud Platform) для хранения и обработки больших данных, что позволяет обеспечить масштабируемость и гибкость без необходимости управлять физической инфраструктурой.

3. Введение в Hadoop

Hadoop — это мощный фреймворк для обработки и хранения больших данных. Он позволяет распределять хранение и обработку данных на кластере из обычных серверов, что обеспечивает масштабируемость и надежность.

Hadoop был разработан для работы с большими объемами разнообразных данных, что делает его одним из основных инструментов в области больших данных.

C:\Что_сегодня_обсудим.txt

Большие данные нужны крупным компаниям, чтобы развивать бизнес-процессы, обгонять конкурентов и улучшать клиентский сервис.

Все эти данные (информация о клиентах, сотрудниках, финансовых показателях, транзакциях, операционной деятельности) нужно где-то хранить, как-то обрабатывать и потом анализировать.

Hadoop - одно из решений для хранения и анализа больших данных.

Сегодня разберём, что такое Hadoop, и какие его функции полезны для бизнеса.

C:\Что_такое_Hadoop.txt

Hadoop помогает хранить и обрабатывать массивы информации, готовить её для выгрузки в другие сервисы, собирать статистику.

Это такой конструктор, на основе которого строят хранилища данных под потребности бизнеса.

Лучше всего Hadoop подходит для работы с неструктурированными данными – неупорядоченной информацией без определённой структуры, которую сложно классифицировать и разбить на группы (файлы документов, сообщения, аудио- и видеозаписи, изображения).

Система может искать нужные сведения в огромном архиве и получать из массива избыточной информации небольшое количество значимой информации для компании.

Так крупная сеть супермаркетов может собирать и обрабатывать информацию о поведении и предпочтениях клиентов из Интернета, обрабатывать её и помещать в хранилище. Там эти данные объединяют с информацией о продажах, анализируют, и становится ясно, какие действия на сайте магазина приводят к покупкам.

ПРЕИМУЩЕСТВА HADOOP

Хранение и быстрая обработка любых данных:

Hadoop можно настроить так, чтобы он обрабатывал информацию со всех Интернет-ресурсов и соцсетей компании, финансовых отчётов и других источников.

Устройчивость к отказам:

В случае аппаратного сбоя, например, если узел вышел из строя, данные пойдут на другой узел, что исключает ошибки. Копии данных сохраняются в системе автоматически.

Высокая мощность вычислений:

Hadoop быстро обрабатывает данные, и мощность зависит от числа вычислительных узлов.

Масштабируемость:

Можно добавлять дополнительные узлы, если объём данных увеличится.

Не нужно обрабатывать данные перед сохранением:

Hadoop обрабатывает и неструктурированные данные (тексты, изображения, видео, т. п.).

Основные компоненты Hadoop:

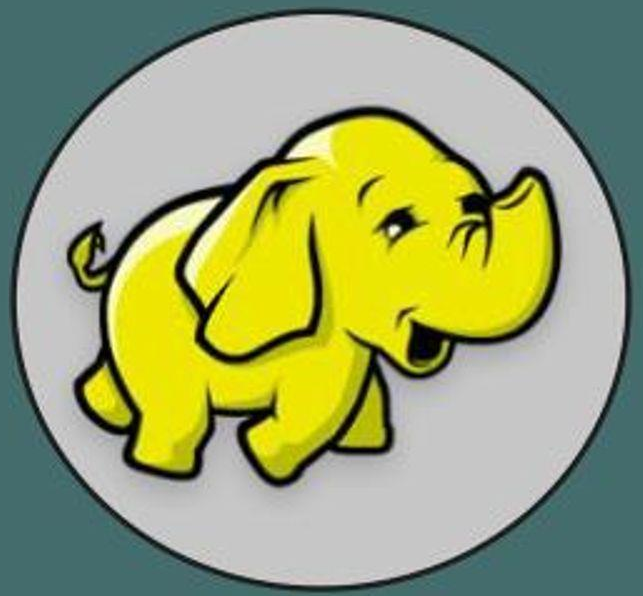
- Hadoop Distributed File System (HDFS):Это распределенная файловая система, предназначенная для хранения больших объемов данных. HDFS разбивает файлы на блоки и распределяет их по узлам кластера, что обеспечивает высокую доступность и отказоустойчивость.
- MapReduce: Это программная модель для обработки больших объемов данных. MapReduce позволяет разбивать задачи на небольшие подзадачи, которые могут выполняться параллельно на разных узлах кластера. Процесс состоит из двух этапов: "Map" (отображение), который обрабатывает входные данные, и "Reduce" (уменьшение), который объединяет результаты.
- Apache Spark: фреймворк для обработки больших данных, предоставляет более быстрые и гибкие возможности обработки данных по сравнению с традиционным MapReduce.

- Apache Hive: предоставляет удобный интерфейс для работы с большими данными, позволяя пользователям выполнять SQL-подобные запросы к данным, хранящимся в HDFS.
- YARN (Yet Another Resource Negotiator): Это компонент управления ресурсами, который позволяет управлять вычислительными ресурсами в кластере. YARN обеспечивает распределение ресурсов между различными приложениями, работающими в Hadoop.
- Hadoop Common: Это набор общих библиотек и утилит, необходимых для работы других компонентов Hadoop. Он включает в себя необходимые файлы и библиотеки для выполнения приложений.

Hadoop Ecosystem Components

JanBask TRAINING
Learn and Grow

- HDFS
- MapReduce
- Hive
- Pig
- HBase
- Hue



- Sqoop
- Flume
- Impala
- Cloudera Search
- spark
- Oozie

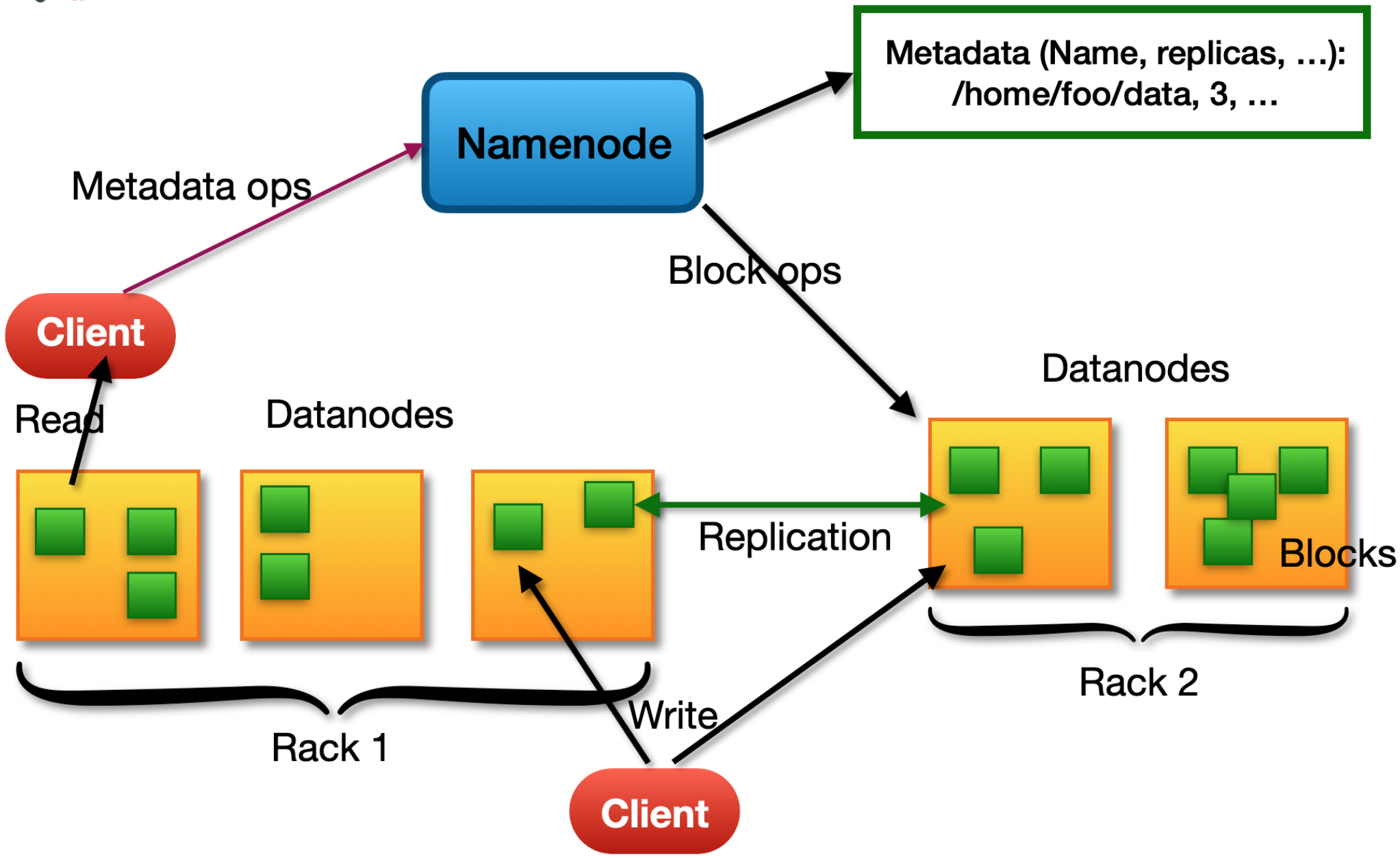
Преимущества Hadoop:

- Масштабируемость: Hadoop может легко масштабироваться от одного сервера до тысяч узлов, что позволяет обрабатывать все больший объем данных.
- Отказоустойчивость: Данные автоматически реплицируются на нескольких узлах, что защищает их от потерь при сбоях оборудования.
- Гибкость: Hadoop поддерживает работу с различными типами данных (структурированные, полуструктурированные и неструктурированные), что позволяет использовать его для широкого спектра задач.
- Кост-эффективность: Hadoop может работать на недорогом оборудовании, что позволяет снижать затраты на инфраструктуру.

Основные принципы работы HDFS включают:

- Распределённое хранение: Данные разбиваются на блоки фиксированного размера (обычно 128 МБ или 256 МБ) и распределяются по узлам кластера. Это обеспечивает высокую доступность и устойчивость к сбоям.
- Репликация: Каждый блок данных реплицируется (по умолчанию три копии) на разных узлах для защиты от потери данных в случае сбоя узла.
- Мастер-слейв архитектура: HDFS использует архитектуру, состоящую из одного NameNode (главный сервер, управляющий метаданными) и множества DataNode (рабочие узлы, на которых хранятся данные).
- Схема на запись: HDFS ориентирован на операции записи: данные записываются в файловую систему только один раз и могут читаться многократно.

HDFS Architecture



MapReduce — это модель программирования, предназначенная для обработки и анализа больших объёмов данных. Основные этапы работы MapReduce:

- Map (отображение): На этом этапе входные данные разбиваются на наборы ключ-значение, и функция map обрабатывает каждый набор, создавая промежуточные пары ключ-значение.
- Shuffle (перетасовка): На этом этапе промежуточные данные, созданные функцией map, группируются по ключам. Этот процесс включает в себя передачу данных между узлами, чтобы все значения, относящиеся к одному ключу, находились на одном узле.
- Reduce (уменьшение): На последнем этапе функция reduce принимает сгруппированные данные и обрабатывает их, производя итоговые результаты. Каждый ключ и соответствующий ему набор значений обрабатываются для создания финальной пары ключ-значение.

4. Компоненты экосистемы Hadoop

Экосистема Hadoop состоит из различных компонентов, каждый из которых выполняет специфические функции для обработки и хранения больших данных. Вот основные компоненты экосистемы Hadoop:

1. Hadoop Common. Библиотеки и инструменты, необходимые для других модулей Hadoop. Они обеспечивают общий функционал, такой как управление файлами и настройка среды.

2. Hadoop Distributed File System (HDFS). Распределённая файловая система, предназначенная для хранения больших объёмов данных. HDFS разбивает файлы на блоки и реплицирует их на различных узлах кластера для обеспечения отказоустойчивости и доступности.

3. MapReduce. Модель программирования и фреймворк для обработки больших объёмов данных с использованием распределённых вычислений. Она делит задачу на этапы: Map, Shuffle и Reduce.

4. YARN (Yet Another Resource Negotiator). Система управления ресурсами, которая управляет вычислительными ресурсами в кластере. YARN позволяет различным приложениям и фреймворкам использовать ресурсы кластера эффективно.

5. Apache Hive. Инструмент для анализа данных, который предоставляет SQL-подобный интерфейс (HiveQL) для выполнения запросов к данным, хранящимся в HDFS. Hive упрощает анализ больших объёмов данных для пользователей, знакомых с SQL.

6. Apache Hbase. Распределённая, масштабируемая база данных NoSQL, работающая поверх HDFS. HBase позволяет быстро получать доступ к данным в реальном времени и подходит для хранения структурированных данных.

7. Apache Pig. Высокоуровневый язык сценариев для обработки и анализа больших данных. Pig Latin позволяет пользователям писать программы, которые затем компилируются в задачи MapReduce.

8. Apache Spark. Быстрый фреймворк для обработки больших данных, который может работать как в пакетном, так и в потоковом режимах. Spark часто используется вместе с Hadoop и может работать на кластере Hadoop через YARN.

9. Apache Flume. Инструмент для сбора и передачи больших объёмов данных в HDFS. Flume используется для сбора данных в реальном времени, таких как логи веб-сайтов и данные с IoT-устройств.

10. Apache Sqoop. Инструмент для передачи данных между Hadoop и реляционными базами данных. Sqoop позволяет импортировать данные из SQL-баз данных в HDFS и экспортировать их обратно.

11. Apache Zookeeper. Система управления координацией, используемая для обеспечения распределённых приложений. Zookeeper помогает управлять конфигурацией, синхронизацией и предоставлением имен.

12. Apache Oozie. Система управления рабочими процессами для координации задач обработки данных в Hadoop. Oozie позволяет создавать, планировать и управлять зависимостями между заданиями MapReduce, Pig, Hive и другими задачами.

5. Применение Hadoop в реальном мире

Hadoop находит широкое применение в различных отраслях благодаря своей способности обрабатывать и хранить большие объемы данных.

Вот несколько кейсов успешного применения Hadoop в реальном мире:

1. Обработка логов и анализ пользовательского поведения.

Компания Yahoo! использует Hadoop для обработки огромных объемов логов, которые генерируются пользователями их сервисов. С помощью Hadoop компания может анализировать поведение пользователей, выявлять тренды и улучшать качество своих услуг.

2. Анализ данных в социальной сети

Facebook применяет Hadoop для обработки и анализа данных о пользователях, включая посты, комментарии и взаимодействия. Это позволяет социальной сети лучше понять предпочтения пользователей и предлагать персонализированный контент и рекламу.

3. Рекомендательные системы

Компания Netflix использует Hadoop для анализа данных о просмотрах, оценках и предпочтениях пользователей. На основе этой информации платформа разрабатывает и улучшает свои рекомендательные системы, что способствует повышению уровня удовлетворенности пользователей.

4. Финансовый анализ и управление рисками

Компания American Express использует Hadoop для анализа транзакционных данных и выявления подозрительных операций. Это позволяет компании улучшить управление рисками и предотвратить мошенничество.

5. Обработка данных о здоровье

Компания: CERN

Описание: CERN использует Hadoop для обработки и анализа данных, полученных от большого адронного коллайдера (LHC). Hadoop помогает исследователям управлять и анализировать массивные объемы научных данных, чтобы делать открытия в области физики элементарных частиц.

6. Анализ медицинских данных

Компания: MD Anderson Cancer Center

Описание: Центр рака MD Anderson использует Hadoop для анализа медицинских данных пациентов. Это позволяет им разрабатывать персонализированные методы лечения и проводить исследования по эффективным стратегиям борьбы с раком.

7. Обработка больших данных для анализа рынка

Компания Home Depot применяет Hadoop для анализа больших объемов данных о продажах, запасах и предпочтениях покупателей. Это помогает компании оптимизировать свои запасы и планировать маркетинговые стратегии.

8. Анализ данных о транспортировке и логистике

Компания UPS использует Hadoop для оптимизации логистики и маршрутов доставки. Анализ данных позволяет компании сократить время и затраты на доставку, улучшая общую эффективность операций.

Заключение

Hadoop применяется в различных отраслях, включая финансы, здравоохранение, ритейл, социальные сети и науку. Он помогает компаниям анализировать большие объемы данных, выявлять инсайты, оптимизировать процессы и улучшать качество услуг, что делает его незаменимым инструментом в современном бизнесе.

Домашнее задание:

1. Повторить материал лекции.

Список литературы:

1. В. Ю. Кара-ушанов SQL — язык реляционных баз данных
2. А. Б. ГРАДУСОВ. Введение в технологию баз данных
3. А.Мотеев. Уроки MySQL

Материалы лекций:

https://github.com/ShViktor72/Education

Обратная связь:

colledge20education23@gmail.com