Week2_Introduction_to_Hadoop

Overview

Hadoop is an open-source framework used to process enormous data sets. It was designed to help organizations manage terabytes of data.

- Set of open-source programs and procedures
- Used for processing large amounts of data
- Servers runs applications on clusters (a collection of computers working together to perform tasks)
- Handles parallel jobs or processes
- Can handle structured, semi-structured and unstructured data

How does Hadoop work?

- **Hadoop Common** is an essential part of the Apache Hadoop Framework that refers to the collection of common utilities and libraries that support other Hadoop modules.
- **HDFS** is the Hadoop Distributed File System, and handles and stores large data running in a commodity hardware (low-specifications industry-grade hardware). It scales a single hadoop cluster as much as thousand clusters
- **MapReduce** is known as Hadoop's processing unit. It processes Big Data by splitting the data into smaller units. The first method used to query data stored in HDFS
- **Yarn** Yet Another Resource Negotiator prepares hadoop for batch, stream, interactive and graph processing. Prepares the RAM.

The challenges of Hadoop
Hadoop is not good for:

- Processing transactions (random access)
- When work cannot be parallelized
- When there are dependencies in the data
- Low latency data access
- Processing lots of small files
- Intensive calculations with little data

Intro to MapReduce

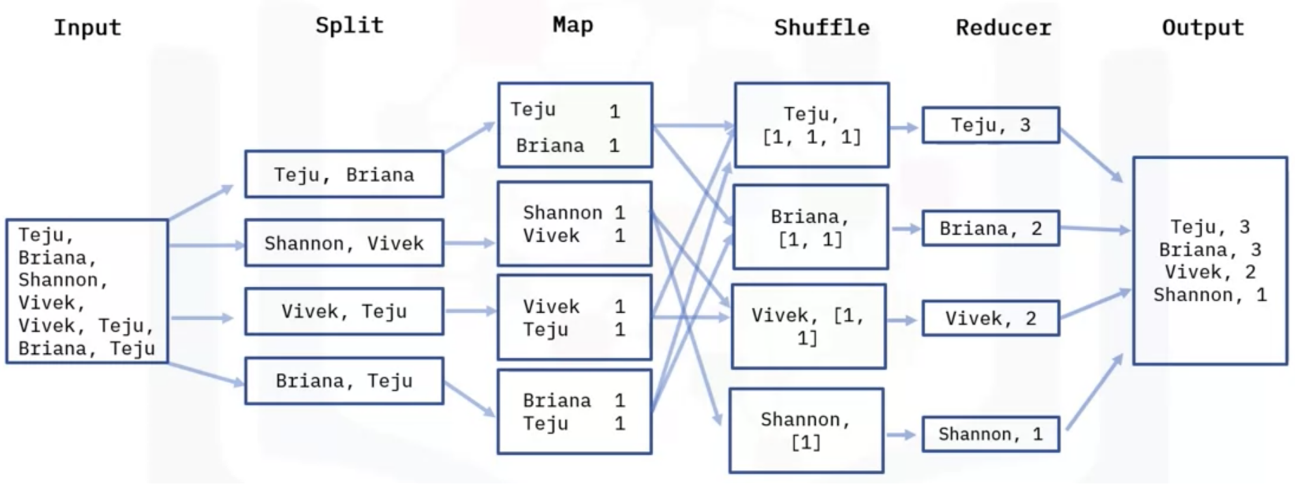
What is?

- Programming model used in Hadoop for processing Big Data

- Processing technique for distributed computing (a system with multiple components located on different machines that communicate actions in one view to the end user), based on Java
- Consists of a Map task and a Reduce tasks
- Can be coded in many programming languages like Java, C++, Python, Ruby, R..

Map and Reduce

1. Input file
2. Map: Processes data into key value pairs
3. Further data sorting and organizing
4. Reducer: Aggregates and computes a set of result and produces a final output
5. MapReduce keeps track of its task by creating a unique key



Why use MapReduce

- Parallel Computing
- Divide -> Run tasks -> Done
- Process data in tabular and non tabular forms, such as videos
- Support for multiple languages
- Platform for analysis and data warehousing

Common Use Cases

- Social Media
  - Social media platforms can use MapReduce to abalyze who visited your profile and who viewed your posts.
- Recommendations
  - Create a recommender's system for users and provide suggestions for them based on their interest.
- Financial Industries

    - Can be used for fraud detection by analyzing behaviors of buyers and tracking down anomalies.
- Advertisment
    - Can be used to analyze and understand the interaction with ads and the engagement levels.

Hadoop Ecosystem

The Hadoop ecosystem is made up of components that support one another

Ingest Data > Store Data > Process and Analyze Data > Access Data
(Flume & Sqoop) > (HDFS & HBase) > (Pig & Hive) > (Impala & Hue)

Ingest

- Flume
    - Collects, aggregates, and transfers big data
    - Has a simple and flexible archutecture based on streaming data flows
    - Uses a simple extensible data model that allows for online analytic application
- Sqoop
    - Designed to transfer data between relational database systems and Hadoop
    - Accesses the database to understand the schema of the data
    - Generates a MapReduce application to import or export data

Store

- HBase
    - A non-relational database that runs on top of HDFS
    - Provides real time wrangling on data
    - Stores data as indexes to allow for random and faster access to data
- Cassandra
    - A scalable, NoSQL database designed to have no single point of failure

Analyze

- Pig
    - Analyzes large amounts of data
    - Operates on the client side of cluster
    - A procedural data flow language
- Hive

    - Used for creating reports
    - Operates on the server side of a cluster
    - A declarative programming language (allows users express which data they wish to receive)

Access

- Impala
    - Scalable and easy to use platform for everyone
    - No programming skills required
- Hue
    - Stands for Hadoop user experience
    - Allows you to upload, browse, and query data
    - Runs Pig jobs and workflow
    - Provides editors for several SQL query languages like Hive and MySQL

HDFS

- Hadoop Distributed File System
- It is the storage layer of Hadoop
- Splites the files into blocks, creates replicas of the blocks, and stores them on different machines
- Provides access to streaming data (a constant bitrate when transferring data rather than having the data being transferred in waves)
- uses a command line interface to interact with hadoop

Key Features

- Cost Efficient
    - The storage hardware is not expensive
- Large amounts of data
    - HDFS can store up to petabytes of data
- Replication
    - Makes copies of the data on multiple machines
- Fault tolerant
    - If one machine crashes, a copy of the data can be found somewhere else and work continues
- Scalable
    - One cluster can be scaled ito hundreds of nodes
- Portable

    - Can easily move across multiple platforms

HDFS concepts

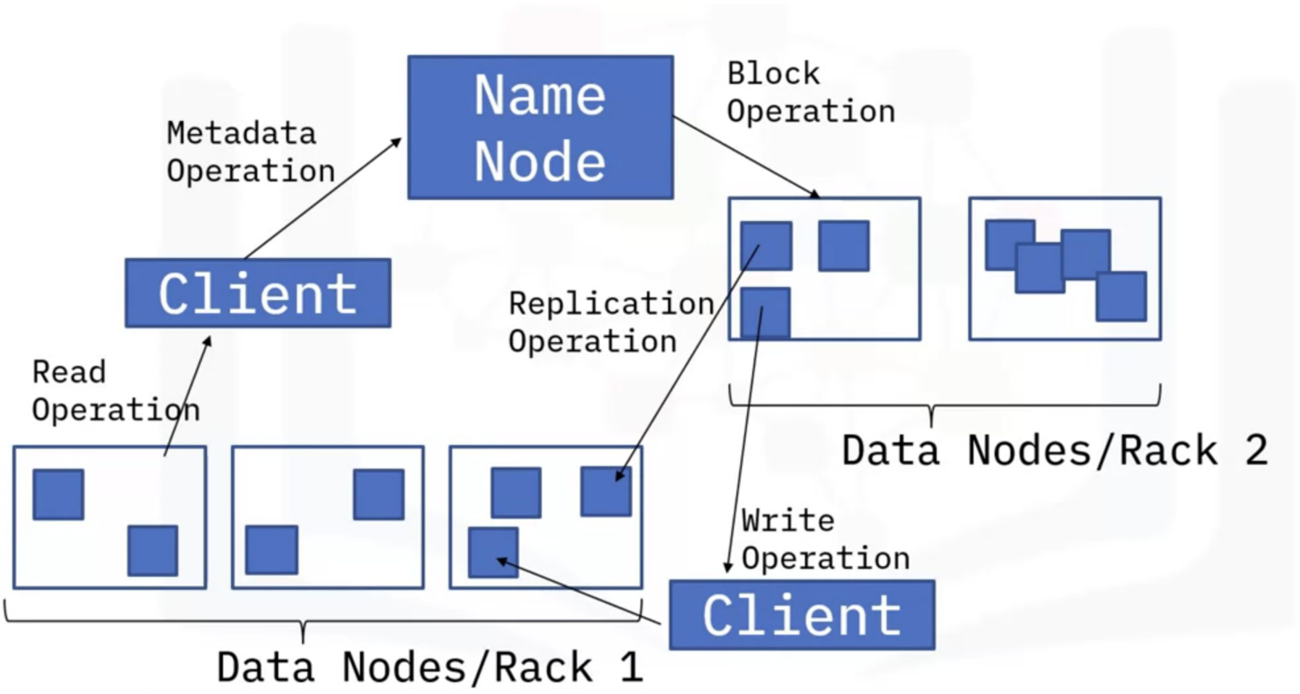
- Blocks
    - Minimum amount of data that can be read or written
    - Provides fault tolerance
    - Default size is 64 or 128 MB
    - Each file stored does'nt have to take up the configured space size
- Nodes
    - A single system which is responsible to store and process data
    - **Primary node (name node)**
        - This node regulates file access to the clients and maintains, manages, and assigns tasks to the secondary node
    - **Secondary node (data node)**
        - These nodes are the actual workers in the HDFS system and take instructions from the primary node
- **Rack awareness in HDFS**
    - Choosing data node racks that are closes to each other
    - Improves cluster performance by reducing network traffic
    - Name node keeps the rack ID information
    - Replication can be done through rack awareness

> A rack is the collection of about 40 to 50 data nodes using the same network switch

- **Replication**
    - Creating a copy of the data block
    - Copies are created for backup purposes
    - Replication factor: Number of times the data block was copied
- **Read and Write Operations**
    - HDFS allows **write once read many operations**
    - **Read**:
        - Client will send a request to the primary node to get the location of the data nodes containing blocks;
        - Client will read files closest to the data nodes
    - **Write**:

  - The name node makes sure that the file does not exists
  - If the file excists client gets an IO Exception message
  - If the file does not exist, the client is given access to start the write

> A client fulfills a user's request by interacting with the Name node and Data nodes



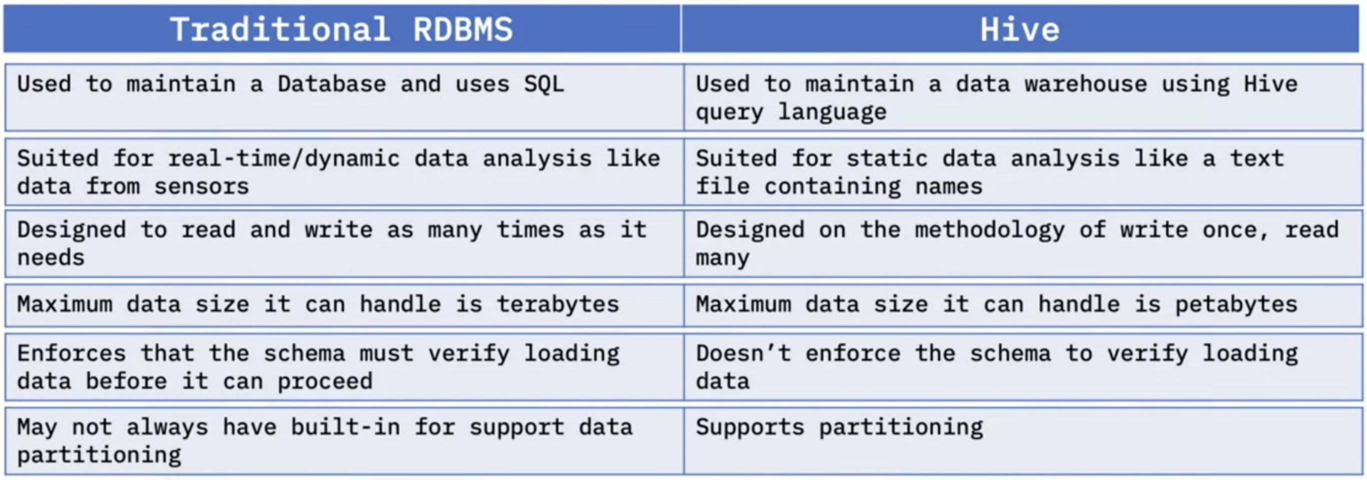
HIVE

Hive is a data warehouse software within Hadoop that is designed for reading, writing, and managing tabular-type datasets and data analysis:

- It is scalable, fast, and easy to use
- Hive query language (HiveQL) is inspired by SQL, making it easier for users to grasp concepts
- It supports data cleansing and filtering daepending on users requirements

> A data warehouse stores historical data from many different sources so that you can analyze and extract insights from it.

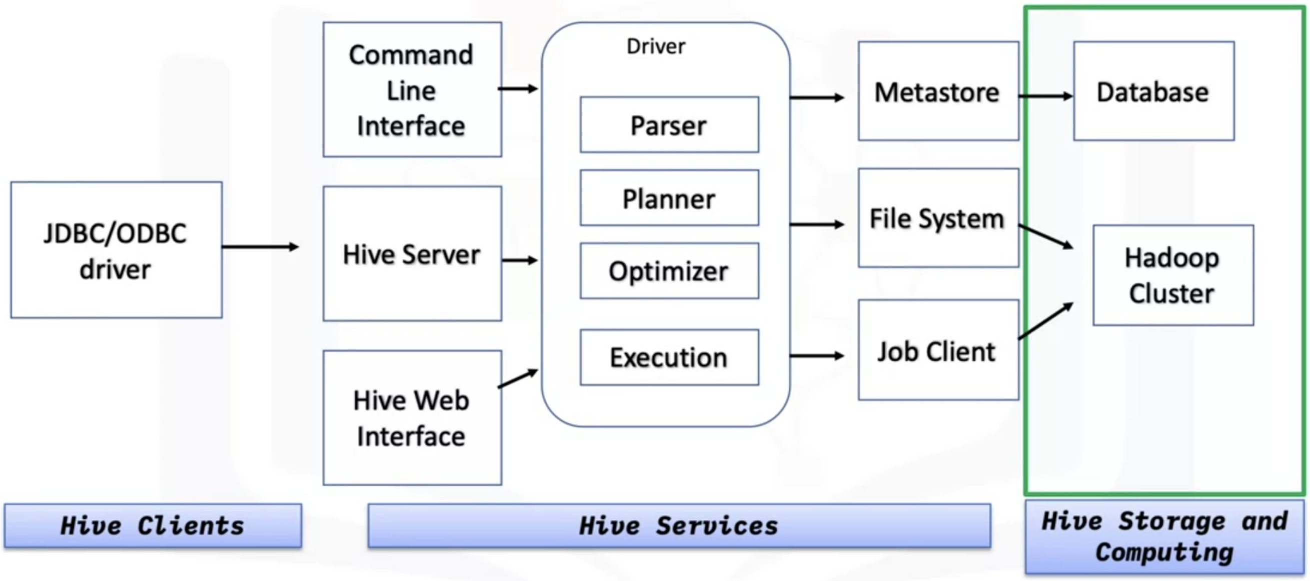
Hive and traditional RDMBS compared



Traditional RDBMS	Hive
Used to maintain a Database and uses SQL	Used to maintain a data warehouse using Hive query language
Suited for real-time/dynamic data analysis like data from sensors	Suited for static data analysis like a text file containing names
Designed to read and write as many times as it needs	Designed on the methodology of write once, read many
Maximum data size it can handle is terabytes	Maximum data size it can handle is petabytes
Enforces that the schema must verify loading data before it can proceed	Doesn’t enforce the schema to verify loading data
May not always have built-in for support data partitioning	Supports partitioning

> Partitioning means dividing the table into parts based on the values of a particular column, such as data or city

Hive architecture



HBASE

HBase is a column-opriented non-relational datbase management system that runs on top of HDFS. It provides a fault-tolerant way of storing sparse datasets and works well with real-time data and random read and write access to Big Data

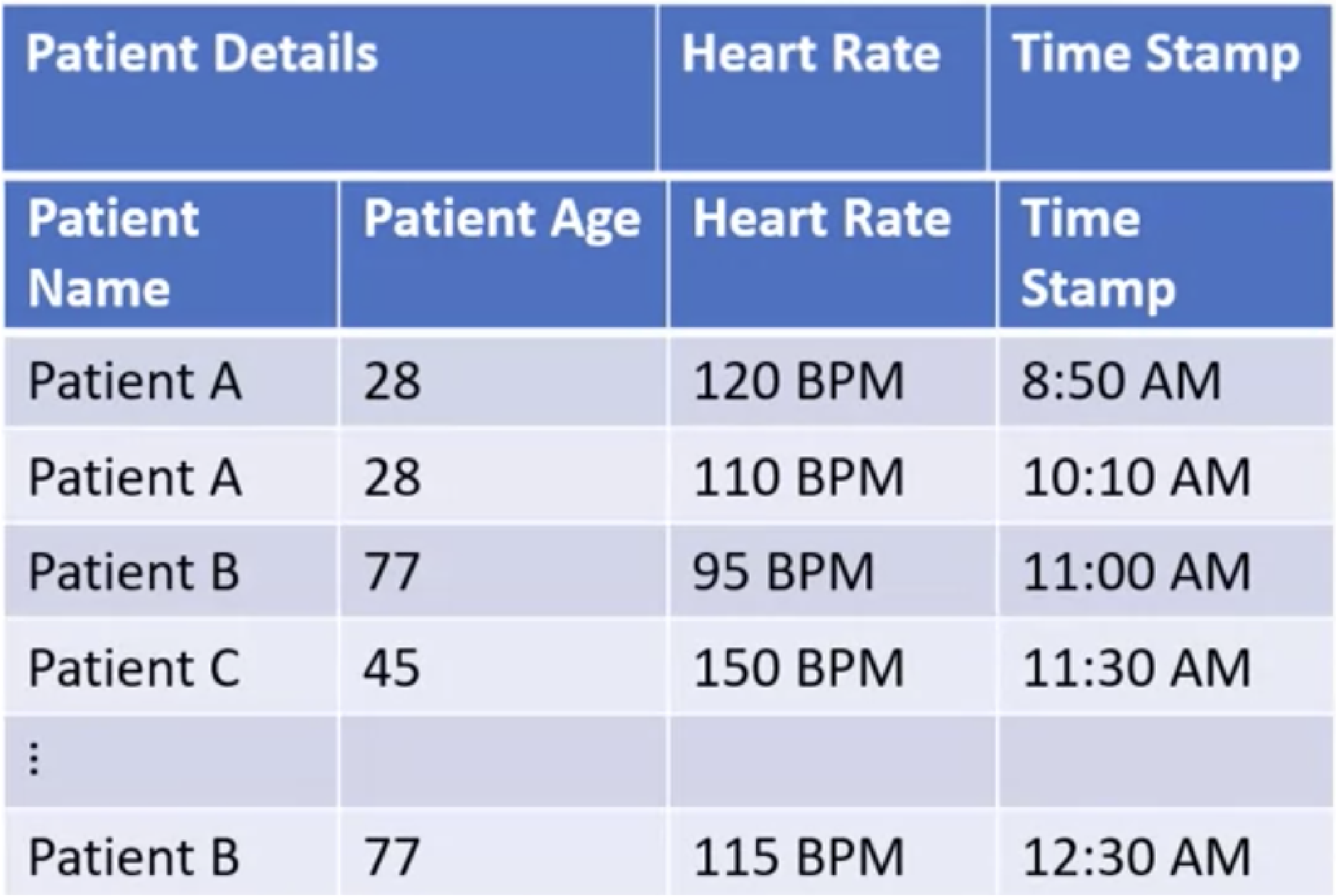
> Fault tolerance refers to the working ability of a system or computer to continue working even in unfavorable conditions such as when a server crashes

Features

- HBase is used for write-heavy applications
- Is linearly and modularly scalable

- It is a backup support for MapReduce jobs
- It provides consistent reads and writes
- It has no fixed column schema
- It is an easy-to-use Java API for client access
- It provides data replication across clusters

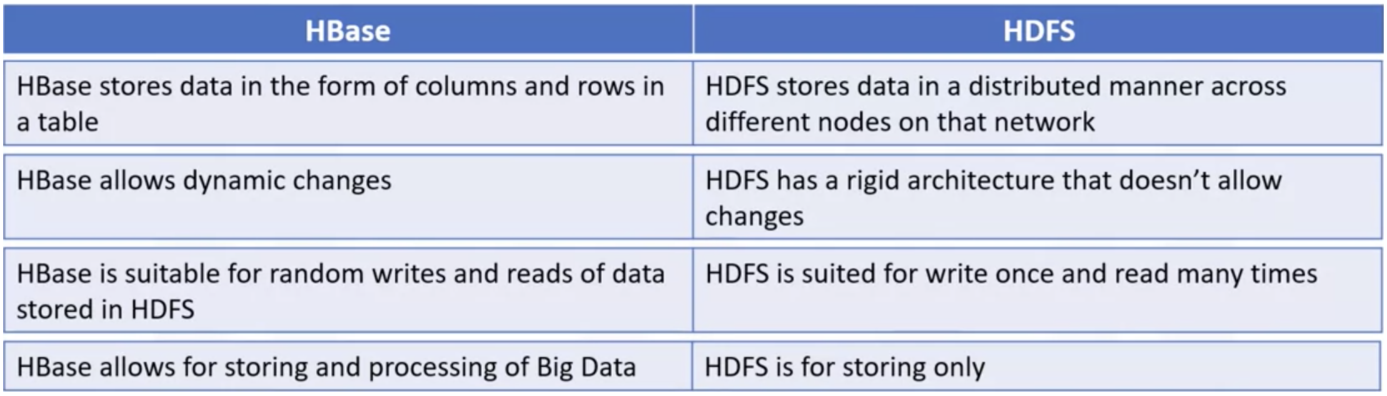
An example



Patient Details		Heart Rate	Time Stamp
Patient Name	Patient Age	Heart Rate	Time Stamp
Patient A	28	120 BPM	8:50 AM
Patient A	28	110 BPM	10:10 AM
Patient B	77	95 BPM	11:00 AM
Patient C	45	150 BPM	11:30 AM
⋮			
Patient B	77	115 BPM	12:30 AM

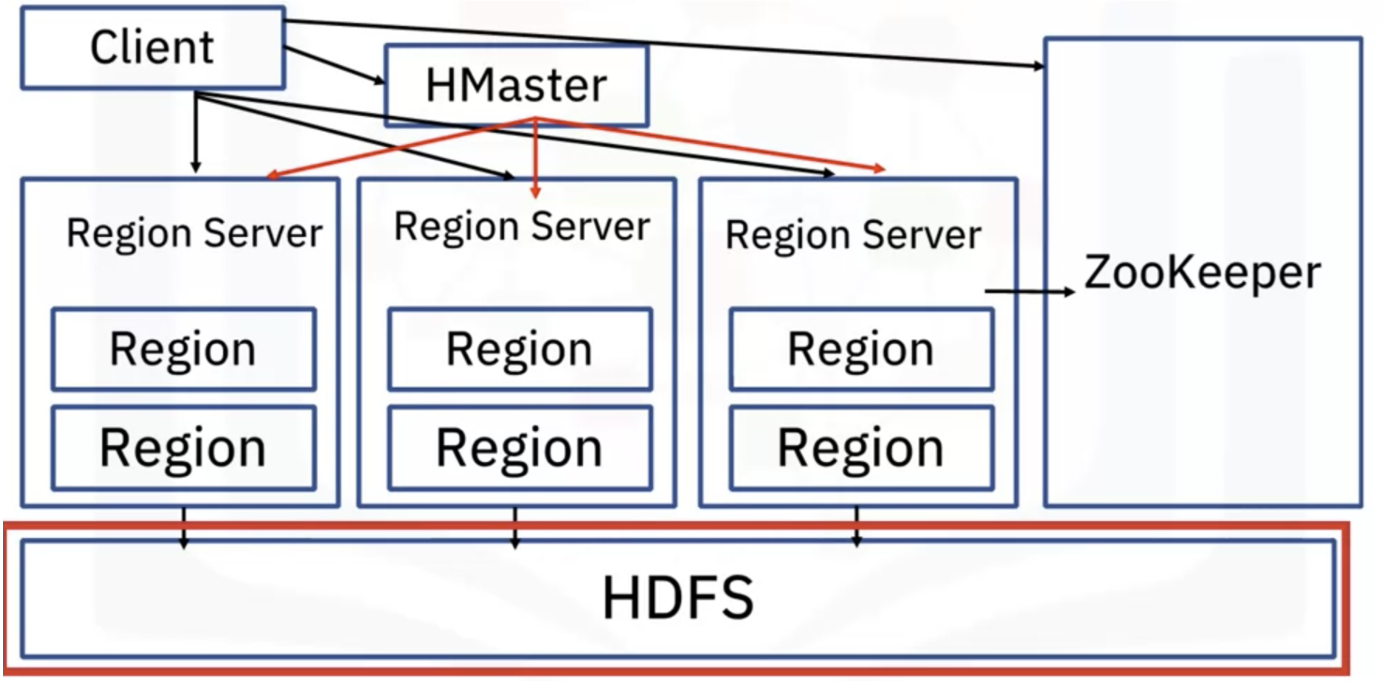
- Predefine the table schema and specify column families
- New columns can be added to column families at any time
- HBase schema is very flexible
- HBase has master nodes to manage the cluster and region servers to perform the work

Differences between HBase and HDFS



HBase	HDFS
HBase stores data in the form of columns and rows in a table	HDFS stores data in a distributed manner across different nodes on that network
HBase allows dynamic changes	HDFS has a rigid architecture that doesn't allow changes
HBase is suitable for random writes and reads of data stored in HDFS	HDFS is suited for write once and read many times
HBase allows for storing and processing of Big Data	HDFS is for storing only

HBase Architecture



- HMaster
  - Monitors the region server instances
  - Assign regions to region servers
  - Manages any changes that are made to the schema
- Region Servers
  - Receives and assigns requests to regions
  - Responsible for managing regions
  - Communicates directly with the client
- Region
  - Smallest unit of HBase cluster
  - Contains multiple stores
  - Two components - HFile and Memstore
- Zookeper
  - Maintains healthy links between nodes
  - Provides distributed synchronization
  - Tracks server failure

Summary

Hadoop is an open-source framework for Big Data that faced challenges when encountering dependencies and low-level latency.

MapReduce, a parallel computing framework used in parallel computing, is flexible for all data types, addresses parallel processing needs for multiple industries and contains two major tasks, “map” and “reduce.”

The four main stages of the Hadoop Ecosystem are Ingest, Store, Process and Analyze, and Access.

Key HDFS benefits include its cost efficiency, scalability, data storage expansion and data replication capabilities. Rack awareness helps reduce the network traffic and improve cluster performance. HDFS enables “write once, read many” operations.

Suited for static data analysis and built to handle petabytes of data, Hive is a data warehouse software for reading, writing, and managing datasets. Hive is based on the “write once, read many” methodology, doesn’t enforce the schema to verify loading data and has built-in partitioning support.

Linearly scalable and highly efficient, HBase is a column-oriented non-relational database management system that runs on HDFS and provides an easy-to-use Java API for client access. HBase architecture consists of HMaster, Region servers, Region, Zookeeper and HDFS. A key difference between HDFS and HBase is that HBase allows dynamic changes compared to the rigid architecture of HDFS.