

What is the Hadoop Ecosystem? The Hadoop Ecosystem is an open-source framework designed for distributed storage and parallel processing of very large datasets across clusters of computers. It is developed by the Apache Software Foundation and aims to provide reliable, scalable, and distributed computing.

How does the Hadoop Ecosystem compare to traditional systems? Compared to traditional systems, the Hadoop Ecosystem is highly scalable, uses horizontal scaling, is more fault-tolerant, and can handle any type of data (structured and unstructured). Traditional systems often require more investment and typically handle only structured data.

What are the main challenges organizations face with big data that Hadoop aims to solve? Organizations struggle with acquiring and storing massive amounts of data, analyzing large datasets for insights, searching and sharing data, and synchronizing and controlling activities. The Hadoop Ecosystem provides tools to address these challenges.

What are the layers of the Hadoop Ecosystem based on functionality? The Hadoop Ecosystem can be conceptually divided into four layers:

- **Data Storage:** Facilitates storing large volumes of data.
- **Data Processing:** Provides tools for parallel processing of big data (e.g., YARN, MapReduce).
- **Data Access:** Offers tools to access the big data repository for information retrieval (e.g., HIVE, PIG, SQOOP, Mahout, Avro).
- **Data Management:** Includes components for managing the ecosystem (e.g., Ambari, ZooKeeper).

What is the Google File System (GFS) and its architecture? GFS is a distributed file system developed by Google for handling large data files. It uses a master-slave architecture with one master server and multiple chunk servers. Files are divided into chunks and stored on chunk servers, with replicas for reliability. The master server manages metadata. Google Bigtable is the database used in the GFS framework.

What is the Hadoop Distributed File System (HDFS) and its architecture? HDFS is Hadoop's primary storage system, a distributed file system designed for large data files on commodity hardware. It follows a master-slave architecture with a single NameNode (master) and multiple DataNodes (slaves). The NameNode manages the file system namespace and metadata, while DataNodes store data chunks and handle read/write requests.

What are the key features of HDFS? Key features of HDFS include:

- **Hierarchical file organization:** Allows for creating directories and managing files (create, move, rename, delete).
- **Reliability:** Achieved through data replication across multiple machines (replication factor is configurable). Rack-aware replication is a standard feature.
- **Data integrity:** Uses checksums to verify data. Corrupted data can be retrieved from replicas.
- **Robustness:** Employs Heartbeats from DataNodes to the NameNode to detect failures and initiate re-replication. A secondary NameNode provides checkpointing for recovery.

What are some key differences between GFS and HDFS?

- **Proprietary vs. Open-Source:** GFS is proprietary to Google, while HDFS is open-source.
- **Chunk/Block Size:** GFS divides files into 64 MB chunks, while HDFS uses 128 MB blocks.
- **Communication Protocol:** GFS uses TCP connections and pipelining. HDFS uses TCP/IP and Remote Procedure Call (RPC).
- **Cache Management:** GFS saves cache metadata in client memory. HDFS uses distributable cache files (private or public).

What is Apache HBase and its features? Apache HBase is a non-relational, scalable, distributed data management system that runs on top of HDFS, providing Bigtable-like capabilities. It's designed for fault-tolerant storage of sparse datasets and real-time big data management. Features include:

- High linear and modular scalability
- Consistent read and write operations
- Automatic and configurable splitting of large tables
- Robustness
- Support for real-time queries using block caching and Bloom Filters.

What are some key differences between Google Bigtable and Apache HBase?

- **Proprietary vs. Open-Source:** Bigtable is proprietary and a Google cloud service. HBase is open-source and can be installed in various environments.
- **Underlying Storage:** Bigtable uses Google's Colossus. HBase uses HDFS.

- **Concept:** Bigtable is a three-dimensional mapping database. HBase is a column-oriented database.
- **Security:** Bigtable uses Google's Cloud Identity and Access Management. HBase uses Access Control Lists.

What are some key differences between HDFS and HBase?

- **Concept:** HDFS is a file distribution system with a rigid architecture. HBase runs on HDFS and supports dynamic changes.
- **Data Access:** HDFS supports sequential read/write (write-once, read-many). HBase allows random read/write access.
- **Data Processing:** HDFS is suited for offline batch processing. HBase is efficient for real-time processing.
- **Latency:** HDFS has high latency for access operations. HBase provides low latency access to small amounts of data.

What is MapReduce? Hadoop MapReduce is the primary processing component of the Hadoop framework, based on Google's MapReduce algorithm. It facilitates distributed parallel processing of large volumes of structured and unstructured data by dividing tasks into independent jobs. It uses a divide and conquer approach.

What are the stages in MapReduce?

- **Map stage:** Input data is split, and map tasks perform operations like filtering, grouping, shuffling, and sorting, generating key-value pairs.
- **Reduce stage:** The output from the map stage is processed by reduce tasks, which perform aggregation and summarization to produce the final result, stored in HDFS.

What is YARN (Yet Another Resource Negotiator)? YARN is a resource management component in Hadoop (introduced in Hadoop 2.0), acting like an operating system on top of HDFS. It manages and allocates cluster resources and schedules jobs, decentralizing tasks previously handled by a single JobTracker in Hadoop 1.0.

What are the components of YARN?

- **ResourceManager:** Submits jobs, schedules them, and allocates resources to applications.
- **NodeManager:** Installed on each node, acts as a monitoring and reporting agent for the ResourceManager, tracking resource usage (CPU, memory, disk, network).

- **ApplicationMaster:** A framework-specific library for each application that negotiates resources from the ResourceManager and works with NodeManagers to execute and monitor tasks.
- **ResourceContainers:** Collections of resources (processing units, memory, disk) on each NodeManager, controlled by NodeManagers and assigned to individual applications.

What are the features of YARN?

- Dynamic allocation of system resources
- Enables real-time data processing alongside MapReduce batch processing
- Enhanced resource utilization and application performance
- Supports various scheduling algorithms
- Reservation feature for reserving cluster resources in advance
- Increased scalability (supports tens of thousands of nodes).

What is Hive? Apache Hive is an ETL (Extract, Transform, Load) and data warehousing tool for querying and managing large datasets stored in multiple locations within the Hadoop Ecosystem. It supports summarization, querying, and analysis of unstructured and semi-structured data using Hive Query Language (HQL), which is similar to SQL and translates queries into MapReduce jobs.

What are the components of Hive?

- **Hive command line interface:** For executing HQL commands.
- **JDBC/ODBC driver:** For database connectivity.

What are the features of Hive?

- High scalability
- Supports real-time processing and batch query processing (interactive query processing)
- Compatibility with SQL primitive data types
- Support for user-defined functions (UDFs)
- Automatic translation of HQL queries into map and reduce functions
- Efficient extensive sequential scans.

What are the limitations of Hive?

- Not effective for scenarios requiring fast response times due to potential high query delays.
- Primarily a read-based system, not well-suited for jobs with many write operations.