

Internal Working Process of Hadoop

Apache Hadoop stores data in a distributed manner and process that data in parallel.

Hadoop Components:

HDFS - Storage layer. It stores the data in a distributed manner. The file gets divided into several blocks which is spreads across the cluster of commodity hardware.

MapReduce - Processing engine of Hadoop. It divides the task submitted by the user into several independent subtasks. This sub-task executes in parallel thereby increasing the throughput.

YARN - A resource management layer. There are two daemons running for Yarn. One is NodeManager on the slave machines and other is the Resource Manager on the master node.

Hadoop Daemons:

Daemons are the processes that run in the background. The Hadoop Daemons are: -

- **Namenode** - It runs on master node for HDFS.
- **Datanode** - It runs on slave nodes for HDFS.
- **Resource Manager** – It runs on YARN master node for MapReduce.
- **Node Manager** – It runs on YARN slave node for MapReduce.

Hadoop Work Process:

- Hadoop does distributed processing for huge data sets across the cluster of commodity servers and works on multiple machines simultaneously.
- To process any data, the client submits data and program to Hadoop.
- HDFS stores the data while MapReduce process the data and Yarn divide the tasks.

HDFS

- Follows Master-Slave topology. It has got two daemons running, they are NameNode and DataNode.

NameNode

- NameNode is the daemon running on the master machine. It tracks where across the cluster the file data resides. It does not store the data contained in these files.

- When the client applications want to add/copy/move/delete a file, they interact with NameNode.
- The NameNode responds to the request from client by returning a list of relevant DataNode servers where the data lives.

DataNode

- DataNode daemon runs on the slave nodes.
- It stores data in the HadoopFileSystem.
- In functional file system data replicates across many DataNodes.
- On startup, a DataNode connects to the NameNode.
- It keeps on looking for the request from NameNode to access data.
- Once the NameNode provides the location of the data, client applications can talk directly to a DataNode.
- while replicating the data, DataNode instances can talk to each other.

Replica Placement

- The rack awareness algorithm determines the rack id of each DataNode.
- Under a simple policy, the replicas get placed on unique racks. This prevents data loss in the event of rack failure.
- For replication factor is three HDFS's placement policy places one replica on a local rack and other two replicas on the remote but same rack. This policy cuts the inter-rack write traffic thereby improving the write performance.

MapReduce

MapReduce algorithm is to process the data in parallel on distributed cluster. It subsequently combines it into the desired result or output.

- In the first step, the program locates and reads the « input file » containing the raw data.
- As the file format is arbitrary, there is a need to convert data into something the program can process. The « InputFormat » and « RecordReader » (RR) does this job.
- InputFormat uses InputSplit function to split the file into smaller pieces
- Then the RecordReader transforms the raw data for processing by the map. It outputs a list of key-value pairs.
- Once the mapper process these key-value pairs the result goes to « OutputCollector ».
- There is another function called « Reporter » which intimates the user when the mapping task finishes.
- In the next step, the Reduce function performs its task on each key-value pair from the mapper.

- Finally, OutputFormat organizes the key-value pairs from Reducer for writing it on HDFS.

YARN

- Yarn divides the task on resource management and job scheduling/monitoring into separate daemons.
- There is one ResourceManager and per-application ApplicationMaster.
- An application can be either a job or a DAG of jobs.
- The ResourceManager has two components – Scheduler and ApplicationManager.
- The scheduler does not track the status of running application. It only allocates resources to various competing applications.
- The scheduler allocates the resources based on fraction of resources like CPU, memory, disk, network etc.

Yarn supports the concept of Resource Reservation via ReservationSystem.

In this, a user can fix several resources for execution of a particular job over time and temporal constraints.

The ReservationSystem makes sure that the resources are available to the job until its completion. It also performs admission control for reservation.

SUMMARY

Hadoop works step by step:

- Input data is broken into blocks of size 128 Mb and then blocks are moved to different nodes.
- Once all the blocks of the data are stored on data-nodes, the user can process the data.
- Resource Manager then schedules the program (submitted by the user) on individual nodes.
- Once all the nodes process the data, the output is written back to HDFS.