

1. What is Hadoop Cluster?

A Hadoop cluster is nothing but a group of computers connected together via LAN.

We use it for storing and processing large data sets. Hadoop clusters have a number of commodity hardware connected together.

They communicate with a high-end machine which acts as a master. These master and slaves implement distributed computing over distributed data storage.

It runs open source software for providing distributed functionality.

2. What is the Basic Architecture of Hadoop Cluster?

Hadoop cluster has [master-slave architecture](#).

i. Master in Hadoop Cluster

It is a machine with a good configuration of memory and CPU. There are two daemons running on the master and they are NameNode and Resource Manager.

a. Functions of NameNode

- Manages file system namespace

- Regulates access to files by clients

- Stores metadata of actual data For example – file path, number of blocks, block id, the location of blocks etc.

- Executes file system namespace operations like opening, closing, renaming files and directories

The NameNode stores the metadata in the memory for fast retrieval. Hence we should configure it on a high-end machine.

b. Functions of Resource Manager

It arbitrates resources among competing nodes
Keeps track of live and dead nodes

ii. Slaves in the Hadoop Cluster

It is a machine with a normal configuration. There are two daemons running on Slave machines and they are – DataNode and Node Manager

a. Functions of DataNode

It stores the business data
It does read, write and data processing operations
Upon instruction from a master, it does creation, deletion, and replication of data blocks.

b. Functions of NodeManager

It runs services on the node to check its health and reports the same to ResourceManager.

We can easily scale Hadoop cluster by adding more nodes to it. Hence we call it a linearly scaled cluster. Each node added increases the throughput of the cluster.

Client nodes in Hadoop cluster – We [install Hadoop](#) and configure it on client nodes.

c. Functions of the client node

To load the data on the Hadoop cluster.
Tells how to process the data by submitting MapReduce job.
Collects the output from a specified location.

3. Single Node Cluster VS Multi-Node Cluster

As the name suggests, single node cluster gets deployed over a *single machine*. And multi-node clusters gets deployed on *several machines*.

In **single-node Hadoop clusters**, all the daemons like NameNode, DataNode run on the same machine. In a single node Hadoop cluster, all the processes run on one JVM instance. The user need not make any configuration setting. The Hadoop user only needs to set JAVA_HOME variable. The default factor for single node Hadoop cluster is one.

In **multi-node Hadoop clusters**, the daemons run on separate host or machine. A multi-node Hadoop cluster has master-slave architecture. In this NameNode daemon run on the master machine. And DataNode daemon runs on the slave machines. In multi-node Hadoop cluster, the slave daemons like DataNode and NodeManager run on cheap machines. On the other hand, master daemons like NameNode and ResourceManager run on powerful servers. In a multi-node Hadoop cluster, slave machines can be present in any location irrespective of the physical location of the master server.

4. Communication Protocols Used in Hadoop Clusters

The **HDFS** communication protocol works on the top of TCP/IP protocol. The client establishes a connection with NameNode using configurable TCP port. Hadoop cluster establishes the connection to the client using client protocol. DataNode talks to NameNode using the DataNode Protocol. A Remote Procedure Call (RPC) abstraction wraps both Client protocol and DataNode protocol. NameNode does not initiate any RPC instead it responds to RPC from the DataNode.

5. How to Build a Cluster in Hadoop

Building a Hadoop cluster is a non-trivial job. Ultimately the performance of our system will depend upon how we have configured our cluster. In this section, we will discuss various parameters one should take into consideration while setting up a Hadoop cluster.

For choosing the right hardware one must consider the following points

- Understand the kind of workloads, the cluster will be dealing with. The volume of data which cluster need to handle. And kind of processing required like CPU bound, I/O bound etc

- Data storage methodology like data compression technique used if any.

- Data retention policy like how frequently we need to flush.

Sizing the Hadoop Cluster

For determining the size of Hadoop clusters we need to look at how much data is in hand. We should also examine the daily data generation. Based on these factors we can decide the requirements of a number of machines and their configuration. There should be a balance between performance and cost of the hardware approved.

Configuring Hadoop Cluster

For deciding the configuration of Hadoop cluster, run typical **Hadoop jobs** on the default configuration to get the baseline. We can analyze job history log files to check if a job takes more time than expected. If so then change the configuration. After that repeat the same process to fine tune the Hadoop cluster configuration so that it meets the business requirement. Performance of the cluster greatly depends upon resources allocated to the daemons. The Hadoop cluster allocates one CPU core for small to medium data volume to each DataNode. And for large data sets, it allocates two CPU cores to the HDFS daemons.

6. Hadoop Cluster Management

When you deploy your Hadoop cluster in production it is apparent that it would scale along all dimensions. They are volume, velocity, and variety. Various features that it should have to become production-ready are – robust, round the clock availability, performance and manageability. Hadoop cluster management is the main aspect of your big data initiative.

A good cluster management tool should have the following features:-

- It should provide diverse work-load management, security, resource provisioning, performance optimization, health monitoring. Also, it needs to provide policy management, job scheduling, back up and recovery across one or more nodes.
- Implement [NameNode high availability](#) with load balancing, auto-failover, and hot standbys
- Enabling policy-based controls that prevent any application from gulping more resources than others.
- Managing the deployment of any layers of software over Hadoop clusters by performing regression testing. This is to make sure that any jobs or data won't crash or encounter any bottlenecks in daily operations.

7. Benefits of Hadoop Clusters

Here is a list of benefits provided by Clusters in Hadoop –

- Robustness
- Data disks failures, heartbeats and re-replication
- Data integrity
- Metadata disk failure
- Snapshot

i. Robustness

The **main objective of Hadoop** is to store data reliably even in the event of failures. Various kind of failure is NameNode failure, DataNode failure, and network partition. DataNode periodically sends a heartbeat signal to NameNode. In network partition, a set of DataNodes gets disconnected with the NameNode. Thus NameNode does not receive any heartbeat from these DataNodes. It marks these DataNodes as dead. Also, Namenode does not forward any I/O

request to them. The replication factor of the blocks stored in these DataNodes falls below their specified value. As a result, NameNode initiates replication of these blocks. In this way, NameNode recovers from the failure.

ii. Data Disks Failure, Heartbeats, and Re-replication

NameNode receives a heartbeat from each DataNode. NameNode may fail to receive heartbeat because of certain reasons like network partition. In this case, it marks these nodes as dead. This decreases the replication factor of the data present in the dead nodes. Hence NameNode initiates replication for these blocks thereby making the cluster fault tolerant.

iii. Cluster Rebalancing

The [HDFS architecture](#) automatically does cluster rebalancing. Suppose the free space in a DataNode falls below a threshold level. Then it automatically moves some data to another DataNode where enough space is available.

iv. Data Integrity

Hadoop [cluster](#) implements checksum on each block of the file. It does so to see if there is any corruption due to buggy software, faults in storage device etc. If it finds the block corrupted it seeks it from another DataNode that has a replica of the block.

v. Metadata Disk Failure

FSImage and Editlog are the central data structures of HDFS. Corruption of these files can stop the **functioning of HDFS**. For this reason, we can configure NameNode to maintain multiple copies of FSImage and EditLog. Updation of multiple copies of FSImage and EditLog can degrade the performance of Namespace operations. But it is fine as Hadoop deals more with the data-intensive application rather than metadata intensive operation.

vi. Snapshot

Snapshot is nothing but storing a copy of data at a particular instance of time. One of the usages of the snapshot is to rollback a failed HDFS instance to a good point in time. We can take Snapshots of the sub-tree of the file system or entire file system. Some of the uses of snapshots are disaster recovery, data backup, and protection against user error. We can take snapshots of any directory. Only the particular directory should be set as Snapshottable. The administrators can set any directory as snapshottable. We cannot rename or delete a snapshottable directory if there are snapshots in it. After removing all the snapshots from the directory, we can rename or delete it.

8. Summary

There are several options to manage a Hadoop cluster. One of them is [Ambari](#). Hortonworks promote Ambari and many other players. We can manage more than one Hadoop cluster at a time using Ambari. Cloudera Manager is one more tool for Hadoop cluster management. Cloudera manager permits us to deploy and operate complete Hadoop stack very easily. It provides us with many features like performance and health monitoring of the cluster. Hope this helped. Share your feedback through comments.