1. **CLUSTER :**

A **computer cluster** consists of a set of loosely or tightly connected [computers](https://en.wikipedia.org/wiki/Computers) that work together so that, in many respects, they can be viewed as a single system. Unlike [grid computers](https://en.wikipedia.org/wiki/Grid_computing), computer clusters have each node set to perform the same task, controlled and scheduled by software.

The components of a cluster are usually connected to each other through fast [local area networks](https://en.wikipedia.org/wiki/Local_area_network) ("LAN"), with each *node* (computer used as a server) running its own instance of an [operating system](https://en.wikipedia.org/wiki/Operating_system). In most circumstances, all of the nodes use the same hardware[[1]](https://en.wikipedia.org/wiki/Computer_cluster#cite_note-1) and the same operating system, although in some setups (i.e. using [Open Source Cluster Application Resources](https://en.wikipedia.org/wiki/Open_Source_Cluster_Application_Resources) (OSCAR)), different operating systems can be used on each computer, and/or different hardware.[[2]](https://en.wikipedia.org/wiki/Computer_cluster#cite_note-2)

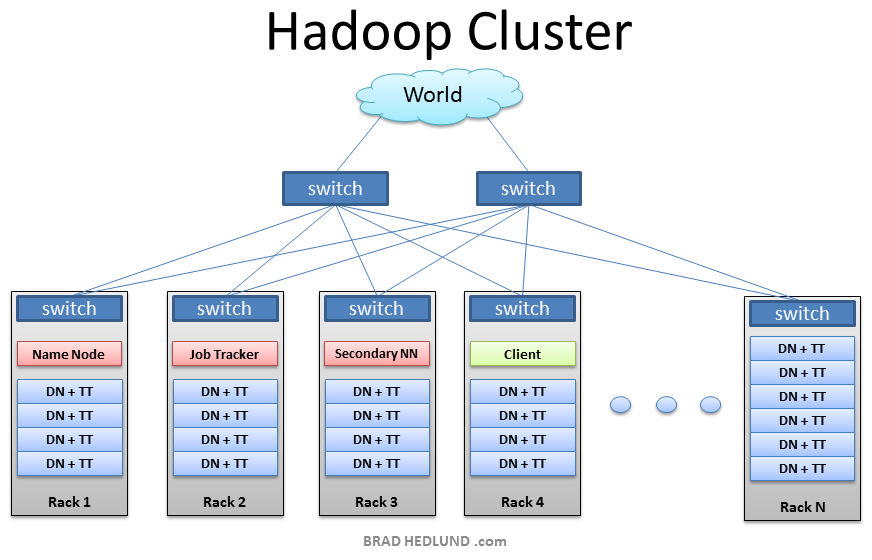
They are usually deployed to improve performance and availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.[[3]](https://en.wikipedia.org/wiki/Computer_cluster#cite_note-3)

Computer clusters emerged as a result of convergence of a number of computing trends including the availability of low-cost microprocessors, high speed networks, and software for high-performance [distributed computing](https://en.wikipedia.org/wiki/Distributed_computing).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] They have a wide range of applicability and deployment, ranging from small business clusters with a handful of [nodes](https://en.wikipedia.org/wiki/Node_(computer_science)) to some of the fastest [supercomputers](https://en.wikipedia.org/wiki/Supercomputer) in the world such as [IBM's Sequoia](https://en.wikipedia.org/wiki/IBM_Sequoia).[[4]](https://en.wikipedia.org/wiki/Computer_cluster#cite_note-4)

**Hadoop Cluster**

This is the typical architecture of a Hadoop cluster. You will have rack servers (Not blades) populated in racks connected to a top of rack sith usualy with 1 or 2 GE boned links. 10 GE nodes are uncommon but gaining interest as machines continue to get more dense with CPU cores and disk drives. The rack switch has uplinks connected to another tier of switches connecting all the other racks with uniform bandwith,forming the cluster. The majority of the server will be slave nodes with lots of local disk storage and moderate amounts of CPU and DRAM. Some of the machines will be Master nodes that might have a slightly different configuration favouring more DRAM and CPU, less local storage.

Hadoop clusters are known for boosting the speed of data analysis applications. They also are highly scalable: If a cluster's processing power is overwhelmed by growing volumes of [data](http://searchdatamanagement.techtarget.com/definition/data), additional cluster nodes can be added to increase throughput. Hadoop clusters also are highly resistant to failure because each piece of data is copied onto other cluster nodes, which ensures that the data is not lost if one node fails.



2.COMPONENTS OF HADOOP 1.X:

* HDFS stands for Hadoop Distributed File System. It is also know as HDFS V1 as it is part of Hadoop 1.x. It is used as a Distributed Storage System in Hadoop Architecture.
* MapReduce is a Batch Processing or Distributed Data Processing Module. It is built by following Google’s MapReduce Algorithm. It is also known as “MR V1” or “Classic MapReduce” as it is part of Hadoop 1.x.

Hadoop 1.x Major Components components are:

* HDFS
* MapReduce.

They are also know as “Two Pillars” of Hadoop 1.x.

HDFS:

It is designed to work with Large DataSets.HDFS component is again divided into two sub-components:

1. Name Node

Name Node is placed in Master Node. It used to store Meta Data about Data Nodes like “How many blocks are stored in Data Nodes, Which Data Nodes have data, Slave Node Details, Data Nodes locations, timestamps etc” .

1. Data Node

Data Nodes are places in Slave Nodes. It is used to store our Application Actual Data. It stores data in Data Slots of size 64MB by default.

**MapReduce:**

MapReduce is a Distributed Data Processing or Batch Processing Programming Model. Like HDFS, MapReduce component also uses Commodity Hardware to process “High Volume of Variety of Data at High Velocity Rate” in a reliable and fault-tolerant manner.

MapReduce component is again divided into two sub-components:

1. Job Tracker

Job Tracker is used to assign MapReduce Tasks to Task Trackers in the Cluster of Nodes. Sometimes, it reassigns same tasks to other Task Trackers as previous Task Trackers are failed or shutdown scenarios.Job Tracker maintains all the Task Trackers status like Up/running, Failed, Recovered etc.

1. Task Tracker

Task Tracker executes the Tasks which are assigned by Job Tracker and sends the status of those tasks to Job Tracker.

Hadoop 1.x components interact each other and to work parallel in a reliable and fault-tolerant manner.

* `Both Master Node and Slave Nodes contain two Hadoop Components:
  1. HDFS Component
  2. MapReduce Component
* Master Node’s HDFS component is also known as “Name Node” is used to store Meta Data.
* Slave Node’s HDFS component is also known as “Data Node” is used to store actual our application Big Data.
* HDFS stores data by using 64MB size of “Data Slots” or “Data Blocks”.
* Master Node’s MapReduce component is also known as “Job Tracker” which take care assigning tasks to “Task Tracker” and receiving results from them.
* Slave Node’s MapReduce component is also known as “Task Tracker” which contains two MapReduce Tasks:
  1. Map Task
  2. Reduce Task
* Clients (one or more) submit their work to Hadoop System.
* When Hadoop System receives a Client Request, first it is received by a Master Node.
* Master Node’s MapReduce component “Job Tracker” is responsible for receiving Client Work and divides into manageable independent Tasks and assign them to Task Trackers.
* Slave Node’s MapReduce component “Task Tracker” receives those Tasks from “Job Tracker” and perform those tasks by using MapReduce components.
* Once all Task Trackers finished their job, Job Tracker takes those results and combines them into final result.
* Finally Hadoop System will send that final result to the Client.