What is Hadoop

Hadoop is an open source framework from Apache and is used to store process and analyze data which are very huge in volume. Hadoop is written in Java and is not OLAP (online analytical processing). It is used for batch/offline processing.It is being used by Facebook, Yahoo, Google, Twitter, LinkedIn and many more. Moreover it can be scaled up just by adding nodes in the cluster.

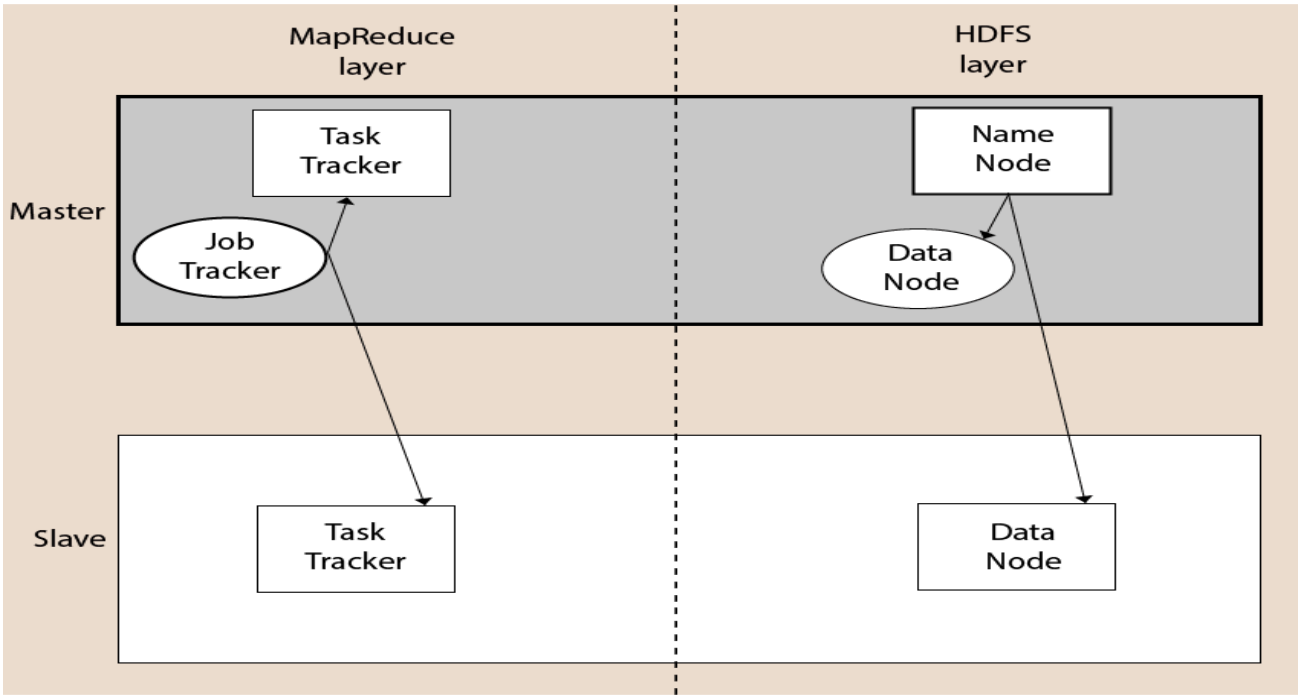
Modules of Hadoop

1. **HDFS: Hadoop Distributed File System. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.**
2. **Yarn: Yet another Resource Negotiator is used for job scheduling and manage the cluster.**
3. **Map Reduce: This is a framework which helps Java programs to do the parallel computation on data using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair. The output of Map task is consumed by reduce task and then the out of reducer gives the desired result.**
4. **Hadoop Common: These Java libraries are used to start Hadoop and are used by other Hadoop modules.**

Hadoop Architecture

The Hadoop architecture is a package of the file system, MapReduce engine and the HDFS (Hadoop Distributed File System). The MapReduce engine can be MapReduce/MR1 or YARN/MR2.

A Hadoop cluster consists of a single master and multiple slave nodes. The master node includes Job Tracker, Task Tracker, NameNode, and DataNode whereas the slave node includes DataNode and TaskTracker.



Hadoop Distributed File System(HDFS)

The Hadoop Distributed File System (HDFS) is a distributed file system for Hadoop. It contains a master/slave architecture. This architecture consist of a single NameNode performs the role of master, and multiple DataNodes performs the role of a slave.

Both NameNode and DataNode are capable enough to run on commodity machines. The Java language is used to develop HDFS. So any machine that supports Java language can easily run the NameNode and DataNode software.

NameNode

- It is a single master server exist in the HDFS cluster.
- As it is a single node, it may become the reason of single point failure.
- It manages the file system namespace by executing an operation like the opening, renaming and closing the files.
- It simplifies the architecture of the system.

DataNode

- The HDFS cluster contains multiple DataNodes.
- Each DataNode contains multiple data blocks.
- These data blocks are used to store data.
- It is the responsibility of DataNode to read and write requests from the file system's clients.
- It performs block creation, deletion, and replication upon instruction from the NameNode.

Job Tracker

- The role of Job Tracker is to accept the MapReduce jobs from client and process the data by using NameNode.
- In response, NameNode provides metadata to Job Tracker.

Task Tracker

- It works as a slave node for Job Tracker.
- It receives task and code from Job Tracker and applies that code on the file. This process can also be called as a Mapper.

MapReduce Layer

The MapReduce comes into existence when the client application submits the MapReduce job to Job Tracker. In response, the Job Tracker sends the request to the appropriate Task Trackers. Sometimes, the TaskTracker fails or time out. In such a case, that part of the job is rescheduled.

Advantages of Hadoop

- **Fast:** In HDFS the data is distributed over the cluster and is mapped which helps in faster retrieval. Even the tools to process the data are often on the

same servers, thus reducing the processing time. It is able to process terabytes of data in minutes and Peta bytes in hours.

- **Scalable:** Hadoop cluster can be extended by just adding nodes in the cluster.
- **Cost Effective:** Hadoop is open source and uses commodity hardware to store data so it is really cost effective as compared to traditional relational database management systems.
- **Resilient to failure:** HDFS has the property with which it can replicate data over the network, so if one node is down or some other network failure happens, then Hadoop takes the other copy of data and uses it. Normally, data are replicated thrice but the replication factor is configurable.

**** What is Hadoop Distributed File System (HDFS) and What are the Design Features HDFS?**

The Hadoop Distributed File System (HDFS) was designed for Big Data storage and processing. HDFS is a core part of Hadoop which is used for data storage. It is designed to run on commodity hardware (low-cost and easily available hardware).

It involves the concept of blocks, data nodes and node name.

Following are the features of Hadoop Distributed File System (HDFS)

- **Highly Scalable -** HDFS is highly scalable as it can scale hundreds of nodes in a single cluster.

- **Replication -** Due to some unfavourable conditions, the node containing the data may be lost. So, to overcome such problems, HDFS always maintains the copy of data on a different machine.
- **Fault tolerance -** In HDFS, the fault tolerance signifies the robustness of the system in the event of failure. The HDFS is highly fault-tolerant that if any machine fails, the other machine containing the copy of that data automatically becomes active.
- **Distributed data storage -** This is one of the most important features of HDFS that makes Hadoop very powerful. Here, data is divided into multiple blocks and stored into nodes.
- **Portable -** HDFS is designed in such a way that it can easily be portable from platform to another.

Goals of HDFS

- **Handling the hardware failure -** The HDFS contains multiple server machines. Anyhow, if any machine fails, the HDFS goal is to recover it quickly.
- **Streaming data access -** The HDFS applications usually run on the general-purpose file system. This application requires streaming access to their data sets.
- **Coherence Model -** The application that runs on HDFS requires to follow the write-once-read-many approach. So, a file once created need not to be changed. However, it can be appended and truncated.

HDFS Concepts

1. **Blocks:** A Block is the minimum amount of data that it can read or write.HDFS blocks are 128 MB by default and this is configurable.Files in HDFS are broken into block-sized chunks,which are stored as independent units.Unlike a file

system, if the file is in HDFS is smaller than block size, then it does not occupy full block?s size, i.e. 5 MB of file stored in HDFS of block size 128 MB takes 5MB of space only.The HDFS block size is large just to minimise the cost of seek.

2. **Name Node:** HDFS works in master-worker pattern where the name node acts as master.Name Node is controller and manager of HDFS as it knows the status and the metadata of all the files in HDFS; the metadata information being file permission, names and location of each block.The metadata are small, so it is stored in the memory of name node,allowing faster access to data. Moreover the HDFS cluster is accessed by multiple clients concurrently,so all this information is handled by a single machine. The file system operations like opening, closing, renaming etc. are executed by it.
3. **Data Node:** They store and retrieve blocks when they are told to; by client or name node. They report back to the name node periodically, with a list of blocks that they are storing. The data node being a commodity hardware also does the work of block creation, deletion and replication as stated by the name node.

Hadoop MapReduce Parallel Data Flow Model Big Data

The Map-Reduce data flow model is a very powerful computational method for big data applications. The underlying idea in the MapReduce computational model is very simple.

There are two stages in the Hadoop MapReduce model, the first stage is the **mapping stage**, and the second stage is the **reducing stage**.

In the mapping stage, a mapping procedure is applied to the given input data. For example, assume that one wants to count how many times each word appears in the novel. One solution is to divide the novel into 20 sections and assign the task of counting the frequency of each word to 20 people. This step is known as the mapping stage.

Let the content of the text file contains two lines:

Line1: Hello World Hadoop World

Line 2: Hello Hadoop Goodbye Hadoop

In this case, each line of the text file is assigned to one person to count the frequency of each word. Hence the output of the mapping stage

First Person:

<Hello, 1>

<World, 2>

< Hadoop, 1>

Second Person:

<Hello, 1>

<Hadoop, 2>

<Goodbye, 1>

The reducing stage starts when everyone is finished counting their assigned text file. The reducer calculates the sum of each word as each one of them tells their counts. The output of the reducing stage is:

<Hello, 2>

<World, 2>

< Hadoop, 3>

<Goodbye, 1>

Following are the steps in Hadoop MapReduce Parallel Data Flow Model

1. Input Splits

Hadoop Distributed File Systems (HDFS) divides the data into multiple blocks. These data blocks are distributed and replicated over multiple storage devices called DatNodes. The default size of the data block is 64MB. Thus, the data with 150MB file size would be divided into 3 data blocks and they will be written into different machines in the Hadoop cluster. Also, they will be replicated in the Hadoop cluster based on the replication factor.

2. Map Step

The mapping stage is where the parallel nature of Hadoop comes into the picture. When we want to process a large amount of data, many mappers can operate at the same time in parallel. The user provides the specific task to the mapper. Mapreduce model executes the mapper process where the data resides. As the data block is replicated on multiple data nodes, the least busy node is selected for the mapper process. If all data nodes holding the data blocks are too busy, the MapReduce model will try to select a data node that is closest to the name node (a characteristic called rack awareness). The last option is any data node in the cluster is chosen that has a data block.

3. Combiner step

It is possible to provide optimization or pre-reduction as part of the mapping stage where key-value pairs are combined prior to the next stage. The combiner stage is optional.

Let the text at mapper is: Hello World Hadoop World

Output of Mapper

<Hello, 1>

<World, 1>

< Hadoop, 1>

<World, 1>

Output of Combiner

<Hello, 1>

<World, 2>

< Hadoop, 1>

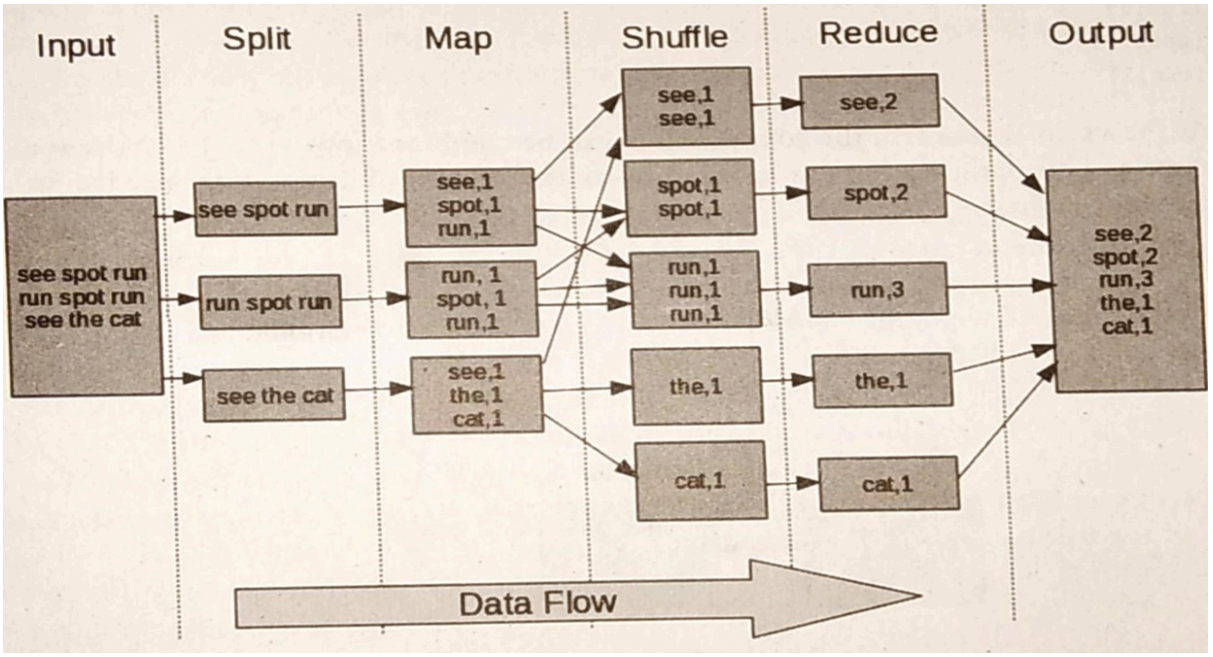
4. Shuffle step

Each reducer process should count similar key-value pairs. Hence, results of the mapping stage must be collected by key-value pairs, then they are shuffled and sent to the same reducing process. If only a single reducer process is used, the Shuffle stage is not needed.

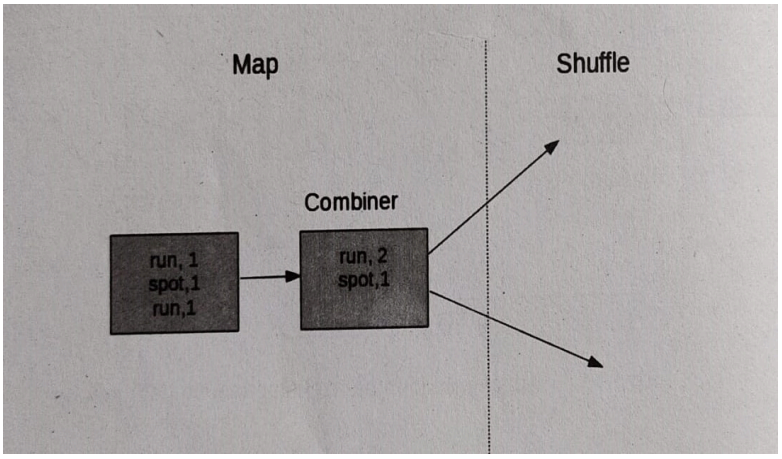
5. Reduce Step

The final step is the actual reduction stage. In this reducing stage, the data reduction is performed based on the programmer's design. The reducing stage is also optional. The results of the reducer stage are written to HDFS. Each reducer process will write an output file. For instance, a MapReduce job with two reducers will create two output files with names part-0000 and part-0001.

Sample Example: Hadoop MapReduce Parallel Data Flow Model



Combiner Stage in MapReduce Model



Usage of MapReduce

- It can be used in various applications like document clustering, distributed sorting, and web link-graph reversal.
- It can be used for distributed pattern-based searching.
- We can also use MapReduce in machine learning.
- It was used by Google to regenerate Google's index of the World Wide Web.

- It can be used in multiple computing environments such as multi-cluster, multi-core, and mobile environments.

Benefits of Hadoop MapReduce

There are numerous benefits of MapReduce; some of them are listed below.

- Speed: It can process large amounts of random data in a short period.
- Fault tolerance: The MapReduce framework can manage failures.
- Most expensive: Hadoop has a rating feature that allows users to process or store data cost-effectively.
- Scalability: Hadoop provides an excellent framework. MapReduce allows users to run applications on multiple nodes.
- Data availability: Data matches are sent to various locations within the network. This ensures that copies of the data are available in case of failure.
- Parallel Processing: On MapReduce, many parts of the same database functions can be processed similarly. This reduces the time taken to complete the task.