What are the different vendor-specific distributions of Hadoop?

The different vendor-specific distributions of Hadoop are Cloudera, MAPR, Amazon EMR, Microsoft Azure, IBM InfoSphere, and Hortonworks (Cloudera).

What are the different Hadoop configuration files?

The different Hadoop configuration files include:

- hadoop-env.sh
- mapred-site.xml
- core-site.xml
- yarn-site.xml
- hdfs-site.xml
- Master and Slaves

What are the three modes in which Hadoop can run

The three modes in which Hadoop can run are :

1.Standalone mode: This is the default mode. It uses the local FileSystem and a single Java process to run the Hadoop services.

2.Pseudo-distributed mode: This uses a single-node Hadoop deployment to execute all Hadoop services.

3.Fully-distributed mode: This uses separate nodes to run Hadoop master and slave services.

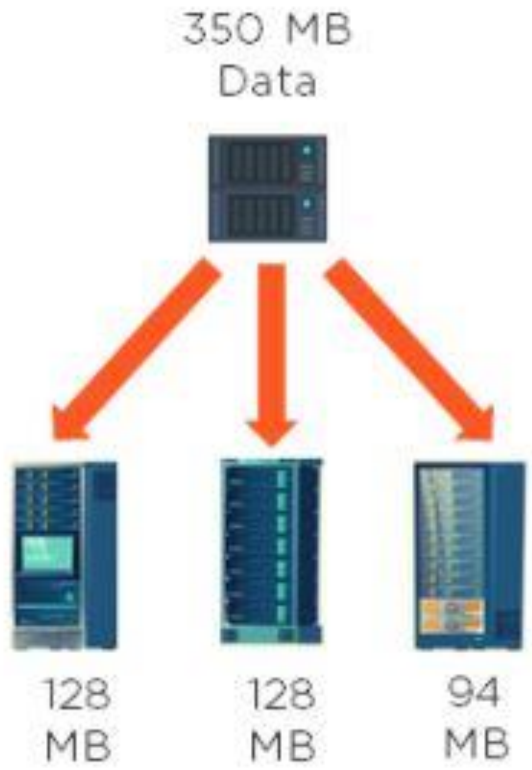
What are the differences between regular FileSystem and HDFS?

1.Regular FileSystem: In regular FileSystem, data is maintained in a single system. If the machine crashes, data recovery is challenging due to low fault tolerance. Seek time is more and hence it takes more time to process the data.

2.HDFS: Data is distributed and maintained on multiple systems. If a DataNode crashes, data can still be recovered from other nodes in the cluster. Time taken to read data is comparatively more, as there is local data read to the disc and coordination of data from multiple systems.

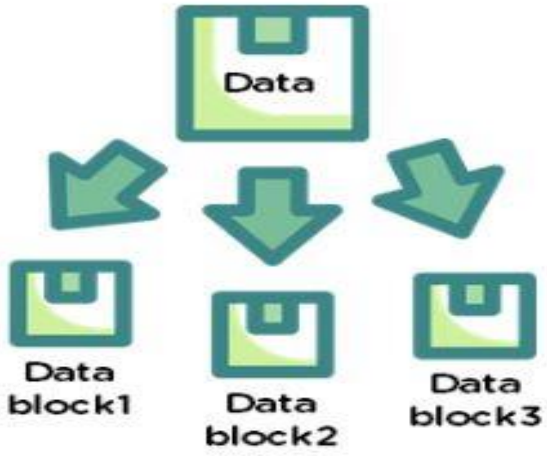
If you have an input file of 350 MB, how many input splits would HDFS create and what would be the size of each input split?

By default, each block in HDFS is divided into 128 MB. The size of all the blocks, except the last block, will be 128 MB. For an input file of 350 MB, there are three input splits in total. The size of each split is 128 MB, 128MB, and 94 MB.

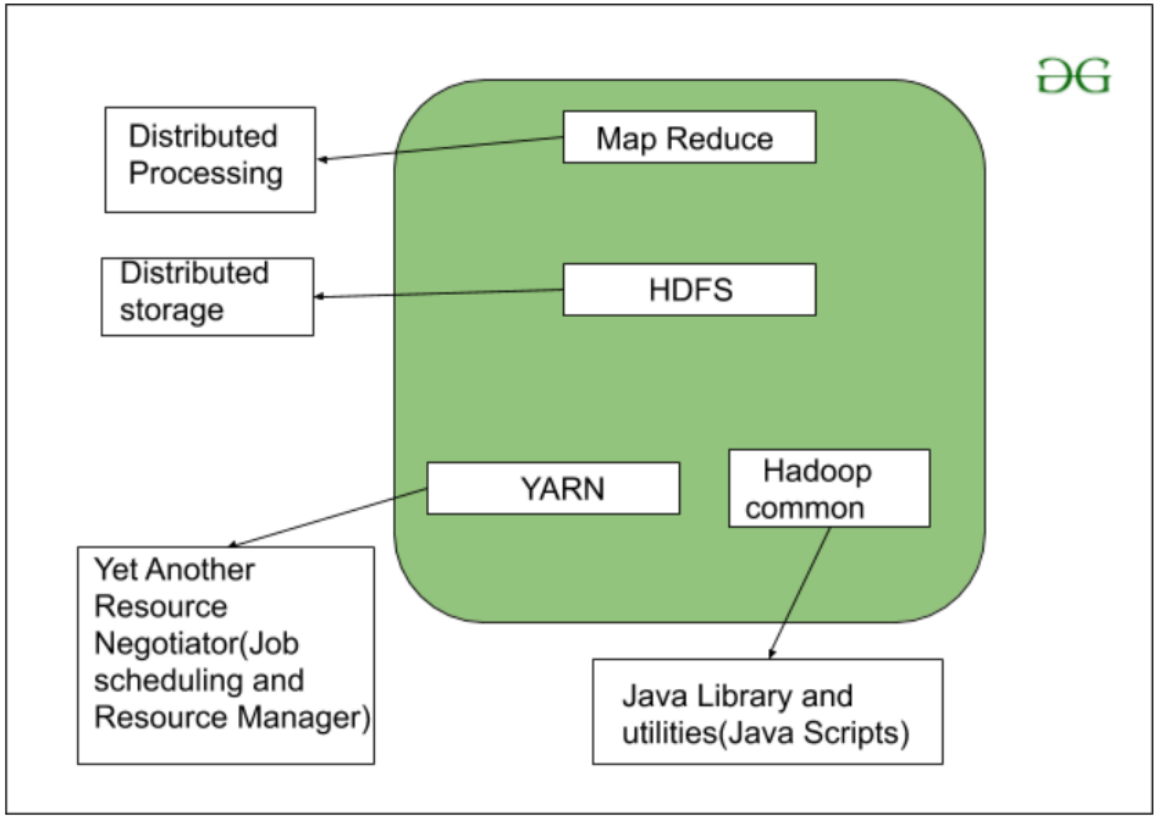


Why is HDFS fault-tolerant?

HDFS is fault-tolerant because it replicates data on different DataNodes. By default, a block of data is replicated on three DataNodes. The data blocks are stored in different DataNodes. If one node crashes, the data can still be retrieved from other DataNodes.



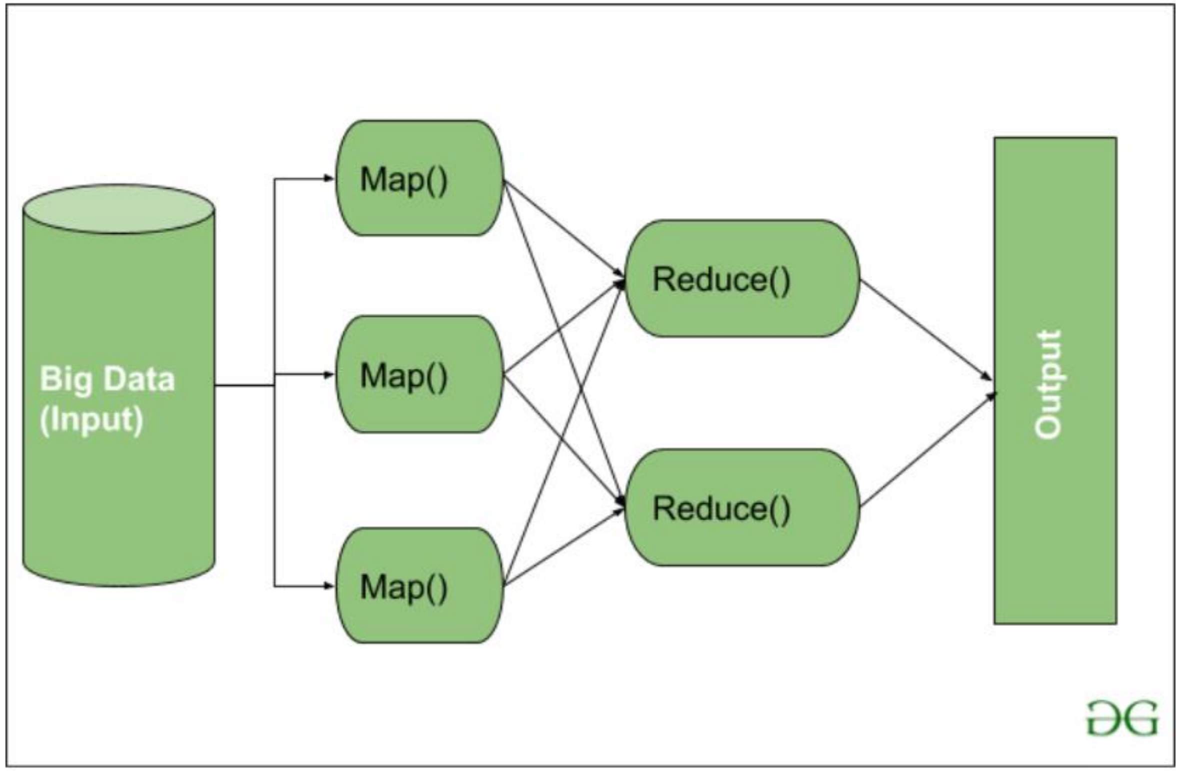
Explain the architecture of HDFS.


ƎG
Distributed Processing
Map Reduce
Distributed storage
HDFS
YARN
Hadoop common
Yet Another Resource Negotiator(Job scheduling and Resource Manager)
Java Library and utilities(Java Scripts)

MapReduce

MapReduce nothing but just like an Algorithm or a data structure that is based on the YARN framework. The major feature of MapReduce is to perform the distributed processing in parallel in a Hadoop cluster which Makes Hadoop working so fast. When you are dealing with Big Data, serial processing is no more of any use. MapReduce has mainly 2 tasks which are divided phase-wise:

In first phase, Map is utilized and in next phase Reduce is utilized.



As we can see that an Input is provided to the Map(), now as we are using Big Data. The Input is a set of Data. The Map() function here breaks this DataBlocks into **Tuples** that are nothing but a key-value pair. These key-value pairs are now sent as input to the Reduce(). The Reduce() function then combines this broken Tuples or key-value pair based on its Key value and form set of Tuples, and perform some operation like sorting, summation type job, etc. which is then sent to the final Output Node. Finally, the Output is Obtained.

HDFS

HDFS(Hadoop Distributed File System) is utilized for storage permission. It is mainly designed for working on commodity Hardware devices(inexpensive devices), working on a distributed file system design. HDFS is designed in such a way that it believes more in storing the data in a large chunk of blocks rather than storing small data blocks.

HDFS in Hadoop provides Fault-tolerance and High availability to the storage layer and the other devices present in that Hadoop cluster. Data storage Nodes in HDFS.

- NameNode(Master)
- DataNode(Slave)

NameNode: NameNode works as a Master in a Hadoop cluster that guides the Datanode(Slaves). Namenode is mainly used for storing the Metadata i.e. the data about the data. Meta Data can be the transaction logs that keep track of the user's activity in a Hadoop cluster.

Meta Data can also be the name of the file, size, and the information about the location(Block number, Block ids) of Datanode that Namenode stores to find the closest DataNode for Faster Communication. Namenode instructs the DataNodes with the operation like delete, create, Replicate, etc.

DataNode: DataNodes works as a Slave DataNodes are mainly utilized for storing the data in a Hadoop cluster, the number of DataNodes can be from 1 to 500 or even more than that. The more number of DataNode, the Hadoop cluster will be able to store more data. So it is advised that the DataNode should have High storing capacity to store a large number of file blocks.

YARN(Yet Another Resource Negotiator)

YARN is a Framework on which MapReduce works. YARN performs 2 operations that are Job scheduling and Resource Management. The Purpose of Job schedular is to divide a big task into small jobs so that each job can be assigned to various slaves in a Hadoop cluster and Processing can be Maximized. Job Scheduler also keeps track of which job is important, which job has more priority, dependencies between the jobs and all the other information like job timing, etc. And the use of Resource Manager is to manage all the resources that are made available for running a Hadoop cluster.

Features of YARN

Multi-Tenancy
Scalability
Cluster-Utilization
Compatibility

Hadoop common or Common Utilities

Hadoop common or Common utilities are nothing but our java library and java files or we can say the java scripts that we need for all the other components present in a Hadoop cluster. these utilities are used by HDFS, YARN, and MapReduce for running the cluster. Hadoop Common verify that Hardware failure in a Hadoop cluster is common so it needs to be solved automatically in software by Hadoop Framework.

For an HDFS service, we have a NameNode that has the master process running on one of the machines and DataNodes, which are the slave nodes.

NameNode

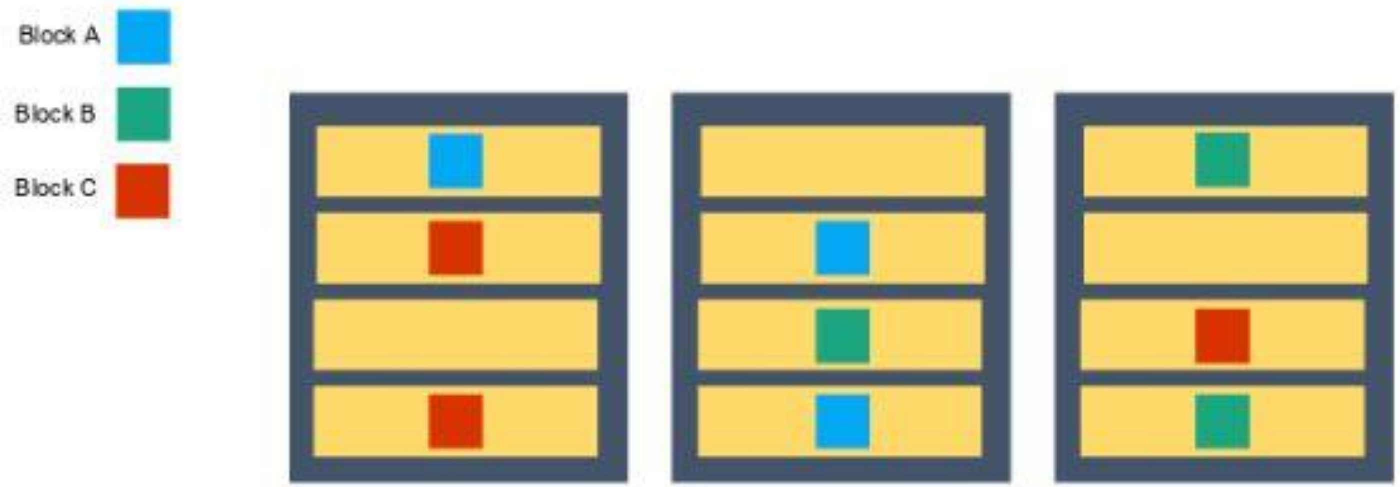
NameNode is the master service that hosts metadata in disk and RAM. It holds information about the various DataNodes, their location, the size of each block, etc.

DataNode

DataNodes hold the actual data blocks and send block reports to the NameNode every 10 seconds. The DataNode stores and retrieves the blocks when the NameNode asks. It reads and writes the client’s request and performs block creation, deletion, and replication based on instructions from the NameNode.

How does rack awareness work in HDFS?

HDFS Rack Awareness refers to the knowledge of different DataNodes and how it is distributed across the racks of a Hadoop Cluster.


Block A
Block B
Block C