Hadoop interview questions on Hadoop & HDFS

1) List the various Hadoop daemons and their roles in a Hadoop cluster.

**Namenode:** It is the Master node which is responsible for storing the meta data for all the files and directories. It has information around blocks that make a file, and where those blocks are located in the cluster.

**Datanode:** It is the Slave node that contains the actual data. It reports information of the blocks it contains to the NameNode in a periodic fashion.

**Secondary Namenode:** It periodically merges changes in the NameNode with the edit log so that it doesn’t grow too large in size. It also keeps a copy of the image which can be used in case of failure of NameNode.

**JobTracker:** This is a daemon that runs on a Namenode for submitting and tracking MapReduce jobs in Hadoop. It assigns the tasks to different task trackers.

**TaskTracker:** This is a daemon that runs on Datanodes. Task Trackers manage the execution of individual tasks on the slave node.

**ResourceManager** (Hadoop 2.x): It is the central authority that manages resources and schedules applications running on top of YARN.

**NodeManager (Hadoop 2.x):** It runs on slave machines, and is responsible for launching the application’s containers, monitoring their resource usage (CPU, memory, disk, network) and reporting these to the ResourceManager.

**JobHistoryServer (Hadoop 2.x):** It maintains information about MapReduce jobs after the Application Master terminates.

2) Name some Hadoop tools that are required to work on Big Data.

“Hive”, “HBase, Ambari and many more. There are many Hadoop tools for Big Data. To learn more about these tools, please visit the Edureka blog on Essential Hadoop Tools for Big Data: <http://www.edureka.co/blog/essential-hadoop-tools-for-big-data>

3) List the difference between Hadoop 1 and Hadoop 2.

In Hadoop 1.x, “Namenode” is the single point of failure. In Hadoop 2.x, we have Active and Passive “Namenodes”. If the active “Namenode” fails, the passive “Namenode” takes charge. Because of this, high availability can be achieved in Hadoop 2.x.

Also, in Hadoop 2.x, YARN provides a central resource manager. With YARN, you can now run multiple applications in Hadoop, all sharing a common resource. MR2 is a particular type of distributed application that runs the MapReduce framework on top of YARN. Other tools can also perform data processing via YARN, which was a problem in Hadoop-1.x.

To learn more about the advantages of Hadoop 2.x, read this blog: <http://www.edureka.co/blog/introduction-to-hadoop-2-0-and-advantages-of-hadoop-2-0/>

4) What are active and passive “Namenodes”?

In Hadoop-2.x, we have two Namenodes – Active “Namenode” and Passive “Namenode”. Active “Namenode” is the “Namenode” which works and runs in the cluster. Passive “Namenode” is a standby “Namenode”, which has similar data as active “Namenode”. When the active “Namenode” fails, the passive “Namenode” replaces the active “Namenode” in the cluster. Hence, the cluster is never without a “Namenode” and so it never fails.

5) How does one remove or add nodes in a Hadoop cluster?

One of the most attractive features of the Hadoop framework is its utilization of commodity hardware. However, this leads to frequent “DataNode” crashes in a Hadoop cluster. Another striking feature of Hadoop Framework is the ease of scale in accordance to the rapid growth in data volume. Because of these two reasons, one of the most common task of a Hadoop administrator is to commission (Add) and decommission (Remove) “Data Nodes” in a Hadoop Cluster.

Read this Edureka blog to get a detailed understanding on commissioning and decommissioning nodes in a Hadoop cluster: <http://www.edureka.co/blog/commissioning-and-decommissioning-nodes-in-a-hadoop-cluster/>

6) What happens when two clients try to access the same file on the HDFS?

HDFS supports exclusive writes only.

When the first client contacts the “Namenode” to open the file for writing, the “Namenode” grants a lease to the client to create this file. When the second client tries to open the same file for writing, the “Namenode” will notice that the lease for the file is already granted to another client, and will reject the open request for the second client

7) Why do we sometimes get a “file could only be replicated to 0 nodes, instead of 1” error?

This happens because the “Namenode” does not have any available DataNodes.

8) How does one switch off the “SAFEMODE” in HDFS?

You use the command: ***hadoop dfsadmin –safemode leave***

9) How do you define “block” in HDFS? What is the block size in Hadoop 1 and in Hadoop 2? Can it be changed?

A “block” is the minimum amount of data that can be read or written. Files in HDFS are broken down into block-sized chunks, which are stored as independent units.

Hadoop 1 default block size: 64 MB

Hadoop 2 default block size:  128 MB

Yes, blocks can be configured. The ***dfs.block.size*** parameter can be used in the ***hdfs-site.xml*** file to set the size of a block in a Hadoop environment.

10) How do you define “rack awareness” in Hadoop?

It is the manner in which the “Namenode” decides how blocks are placed, based on rack definitions to minimize network traffic between “DataNodes” within the same rack. Let’s say we consider replication factor 3 (default), the policy is that “for every block of data, two copies will exist in one rack, third copy in a different rack”. This rule is known as the “Replica Placement Policy”.

11) Why do we use HDFS for applications having large data sets and not when there are a lot of small files?

HDFS is more suitable for large amount of data sets in a single file as compared to small amount of data spread across multiple files. This is because “Namenode” is a very expensive, high performance system, and it is not prudent to occupy the space in the “Namenode” by unnecessary amounts of metadata that are generated for multiple small files. So, when there is a large amount of data in a single file, “Namenode” will occupy less space. Hence, for getting optimized performance, HDFS supports large data sets instead of multiple small files.

12) What is the basic difference between traditional RDBMS and Hadoop?

Traditional RDBMS is used for transactional systems to report and archive the data, whereas Hadoop is an approach to store and process huge amounts of data in the distributed file system. RDBMS will be useful when you want to seek one record from Big data, whereas Hadoop will be useful when you want Big Data in one shot and perform analysis on that later.

13) Explain the indexing process in HDFS.

Hadoop has its own way of indexing data. Depending on the block size, HDFS will continue storing the last part of the data. It will also tell you where the next part of the data is located.

14) What is “speculative execution” in Hadoop?

If a node appears to be running a task slower, the master node can redundantly execute another instance of the same task on another node. Here, the task which finishes first will be accepted and the other one is killed. This process is called “speculative execution”.

15) How do you achieve HA (High Availability) in a Hadoop cluster?

You can set up HA in two different ways; Using the Quorum Journal Manager (QJM), or with NFS for the shared storage. To understand this in detail, do read this Edureka blog: <http://www.edureka.co/blog/how-to-set-up-hadoop-cluster-with-hdfs-high-availability/>

16) Why is it that in HDFS, “Reading” is performed in parallel but “Writing” is not?

Using the MapReduce program, the file can be read by splitting its blocks. But while writing, MapReduce cannot be applied and no parallel writing is possible. Hence, the incoming values are not yet known to the system.

*Its always good to have a comprehensive list of all the topics which might be asked in a Hadoop interview. Edureka has a specially curated Hadoop course that will prepare you for a Hadoop career.*

[Check-out-Hot-Hadoop-Topics](http://www.edureka.co/big-data-and-hadoop?utm_source=blog&utm_medium=button&utm_campaign=hadoop-interview-questions-check-out-hot-hadoop-topics)

17) How can I restart “Namenode”?

* Click on ***stop-all.sh*** and then click on ***start-all.sh*** **OR**
* Write ***sudo hdfs*** (press enter), ***su-hdfs*** (press enter), ***/etc/init.d/ha*** (press enter) and then ***/etc/init.d/hadoop-x.x-namenode start*** (press enter).

18) What happens if you get a “Connection refused java exception’ error when you try to access HDFS or its corresponding files?

It could mean that the “Namenode” is not working on your VM. The “Namenode” may be in “Safemode” or the IP address of “Namenode” may have changed.

19) What is “commodity hardware”? Does it include RAM?

“Commodity hardware” is a non-expensive system which doesn’t have high quality or high availability. Hadoop can be installed in any average commodity hardware. We don’t need super computers or high-end hardware to work on Hadoop. Yes, commodity hardware includes RAM because some services might still be running on RAM.

20) What is the difference between an “HDFS Block” and an “Input Split”?

“HDFS Block” is the physical division of the data while “Input Split” is the logical division of the data.

21) What are the features of a “Fully Distributed” mode?

“Fully Distributed” mode is used in the production environment, where we have ‘n’ number of machines forming a Hadoop cluster. Hadoop daemons run on a cluster of machines. There is one host on which “Namenode” runs and another host on which “Datanode” runs, and then there are machines on which “TaskTracker/NodeManager” runs. We have separate masters and slaves in this sort of a distribution.

22) Name the three modes in which Hadoop can be run.

The three modes in which Hadoop can be run are:

1. Standalone (local) mode

2. Pseudo distributed mode

3. Fully distributed mode

23) What is the role of “ZooKeeper” in a Hadoop cluster?

The purpose of “ZooKeeper” is cluster management. “ZooKeeper” will help you achieve coordination between Hadoop nodes. “ZooKeeper” also helps to:

* Manage configuration across nodes
* Implement reliable messaging
* Implement redundant services
* Synchronize process execution

Hadoop interview questions around MapReduce

24) What is “MapReduce”? What is the syntax to run a “MapReduce” program?

It is a framework or a programming model that is used for processing large data sets over clusters of computers using distributed programming. The syntax to run a MapReduce program is ***hadoop jar file.jar /input\_path /output\_path***

25) MapReduce Vs Pig. Explain the key differences.

Java is a first-class language in Hadoop and will always give the developer more options. However, Pig is written in Java and allows for developers to write User Defined functions in Java that leverage Java Libraries. So, we can call Pig Latin a second-class language in MapReduce, and other languages like Python, Bash and C# are 3rd class.

Pig is largely used for prototyping and quick and dirty analysis. Pig uses a high level scripting language called Pig Latin for writing prototypes and internally converts into MapReduce. This makes it slower than MapReduce. Once the prototype is validated in Pig, then the code can be moved to the lower level i.e. MapReduce which provides a lot more flexibility for optimization and increasing the speed of execution. Writing MapReduce code takes more time but the time of execution is much faster in MapReduce.

26) How would you debug a Hadoop code?

There are many ways to debug Hadoop codes but the most popular methods are:

* Using Counters.
* Using the web interface provided by the Hadoop framework.

27) What are the main configuration parameters in a “MapReduce” program?

Users of the “MapReduce” framework need to specify these parameters:

* Job’s input locations in the distributed file system
* Job’s output location in the distributed file system
* Input format
* Output format
* Class containing the “map” function
* Class containing the “reduce” function

28) What is the default input type/format in “MapReduce”?

By default, the type input type in MapReduce is “text”.

29) State the reason why we can’t perform “aggregation” (addition) in a mapper? Why do we need the “reducer” for this?

We cannot perform “aggregation” (addition) in a mapper because sorting does not occur in the “mapper”. Sorting occurs only on the reducer side. The “Mapper” method initialization depends on each input split. During “aggregation”, we will lose the value of the previous instance. For each row, a new “mapper” will get initialized. For each row, “input split” again gets divided into the “mapper”. Hence, we cannot have a track of the previous row value.

30) What is the purpose of “RecordReader” in Hadoop?

The “InputSplit” has defined a slice of work, but does not describe how to access it. The “RecordReader” class loads the data from its source and converts it into (***key, value***) pairs suitable for reading by the “Mapper”. The “RecordReader” instance is defined by the “Input Format”.

31) Explain “Distributed Cache” in a “MapReduce Framework”

“Distributed Cache” is an important feature provided by the “MapReduce Framework”.  “Distributed Cache” is used when you want to share files across multiple nodes in a Hadoop Cluster. The files could reside as executable “jar” files or simple “properties” files.

Check out this Edureka blog for a webinar recording on Distributed Cache with MapReduce: <http://www.edureka.co/blog/distributed-cache-with-mapreduce>

32) What mechanism does the “Hadoop Framework” provide to synchronize changes made in the “Distribution Cache” during runtime of the application?

This is a tricky question. There is no such mechanism. “Distributed Cache” by design is “read only” during the time of job execution.

33) How do “reducers” communicate with each other?

This is another tricky question. The “MapReduce” programming model does not allow “reducers” to communicate with each other. “Reducers” run in isolation.

Explain the role of “Reduce Side Join” in “MapReduce”.

Joining data is one of the most attractive aspects of Hadoop. Organizing data is a breeze with “Reduce Side Join”. To learn more, do check out this Edureka blog with a webinar recording of “Reduce Side Join in MapReduce: <http://www.edureka.co/blog/reduce-side-joins-with-mapreduce>

34) What does a “MapReduce Partitioner” do?

A “MapReduce Partitioner” makes sure that all the values of a single key goes to the same “reducer”, thus allowing even distribution of the map output over the “reducers”. It redirects the “mapper” output to the “reducer” by determining which “reducer” is responsible for the particular key.

35) What is a “Combiner”?

A “Combiner” is a mini “reducer” that performs the local “reduce” task. It receives the input from the “mapper” on a particular “node” and sends the output to the “reducer”. “Combiners” help in enhancing the efficiency of “MapReduce” by reducing the quantum of data that is required to be sent to the “reducers”.

36) What do you know about “SequenceFileInputFormat”?

“SequenceFileInputFormat” is an input format for reading within sequence files. Key and value are user defined. It is a specific compressed binary file format which is optimized for passing the data between the output of one “MapReduce” job to the input of some other “MapReduce” job.

Hadoop interview questions around Pig

37) What is a “Bag”?

A Bag is one of the data models present in “Pig”. It is an unordered collection of tuples with possible duplicates. “Bags” are used to store collections while grouping. The size of “Bag” is the size of the local disk, which means that the size of the “Bag” is limited. When a “Bag” is full, “Pig” will spill this “Bag” into the local disk and keep only some parts of the “Bag” in memory. It is not necessary that the complete “Bag” fit into the memory. We represent “Bag” with “***{}***”.

38) What does “FOREACH” do?

“FOREACH” is used to apply transformations to the data and to generate new data items. The name itself indicates that for each element of a data “Bag”, the respective action will be performed.

Syntax: ***FOREACH bagname GENERATE expression1, expression2, …..***

The meaning of this statement is that the expressions mentioned after “GENERATE” will be applied to the current record of the data “Bag”.

39) Why do we need “MapReduce” during “Pig” programming?

“Pig” is a high level platform that makes Hadoop data analysis issues easier to execute. The language we use for this platform is “Pig Latin”. A program written in “Pig Latin” is like a query written in SQL, where we need an execution engine to execute the query. So, when a program is written in “Pig Latin”, the “Pig compiler” will convert the program into “MapReduce” jobs. Here, “MapReduce” acts as the execution engine.

40) What is the role of a “co-group” in “Pig”?

“Co-group” joins the data set by grouping one particular data set only. It groups the elements by their common field and then returns a set of records containing two separate “bags”. The first “bag” consists of records from the first data set with the common data set, while the second “bag” consists of records from the second data set along with the common data set.

41) What are the different relational operations in “Pig Latin”?

They are:

i. for each

ii. order by

iii. filters

iv. group

v. distinct

vi.  join

vii. limit

Hadoop interview questions around Hive

42) What is “SerDe” in “Hive”?

The “SerDe” interface allows you to instruct “Hive” about how a record should be processed. A “SerDe” is a combination of a “Serializer” and a “Deserializer”. “Hive” uses “SerDe” (and “FileFormat”) to read and write table rows.

43) Can the default “Hive Metastore” be used by multiple users (processes) at the same time?

“Derby database” is the default “Hive Metastore”. Multiple users (processes) cannot access it at the same time. It is mainly used to perform unit tests.

44) What is the default location where “Hive” stores table data?

***hdfs://namenode/user/hive/warehouse***

45) What is a “generic UDF” in “Hive”?

It is a UDF which is created using a Java program to serve some specific need not covered under the existing functions in “Hive”. It can detect the type of input argument programmatically and provide appropriate responses.

Hadoop interview questions around Oozie

46) How do you configure an “Oozie” job in Hadoop?

“Oozie” is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs such as “Java MapReduce”, “Streaming MapReduce”, “Pig”, “Hive” and “Sqoop”. To understand “Oozie” in detail and learn how to configure am “Oozie” job, do check out this Edureka blog: <http://www.edureka.co/blog/brief-introduction-to-oozie/>

Hadoop interview questions around Sqoop

47) Explain “Sqoop” in Hadoop.

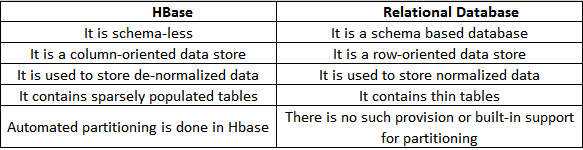
“Sqoop” is a tool used to transfer data between an RDBMS and a Hadoop HDFS. Using “Sqoop”, data can be transferred from an RDBMS (such as MySQL or Oracle) into the HDFS as well as export data from HDFS file to RDBMS.

Hadoop interview questionsaround HBase

48) Explain “WAL” and “Hlog” in “HBase”?

“WAL” (Write Ahead Log) is similar to the “MySQL BIN” log; it records all the changes that occur in the data. It is a standard sequence file by Hadoop and stores “HLogkeys”.  These keys consist of a sequential number as well as actual data and are used to replay not yet persisted data after a server crash. So, in case of server failure, the “WAL” works as the lifeline and retrieves the lost data.

49) Mention the differences between “HBase” and “Relational Databases”?

[](http://www.edureka.co/big-data-and-hadoop?utm_source=blog&utm_medium=image&utm_campaign=top-50-hadoop-interview-questions-you-should-prepare-for-in-2016-22dec15)

Hadoop interview questions around Spark

50) Can you build “Spark” with any particular Hadoop version and “Hive”?

Yes, you can build “Spark” for a specific Hadoop version. Check out this Edureka blog to learn more: <http://www.edureka.co/blog/yarn-hive-get-electrified-by-spark/>

Feeling overwhelmed with all the questions the interviewer might ask in your Hadoop interview? It’s never too late to strengthen your basics. Learn Hadoop from industry experts while working with real-life use cases. Click below to know more.