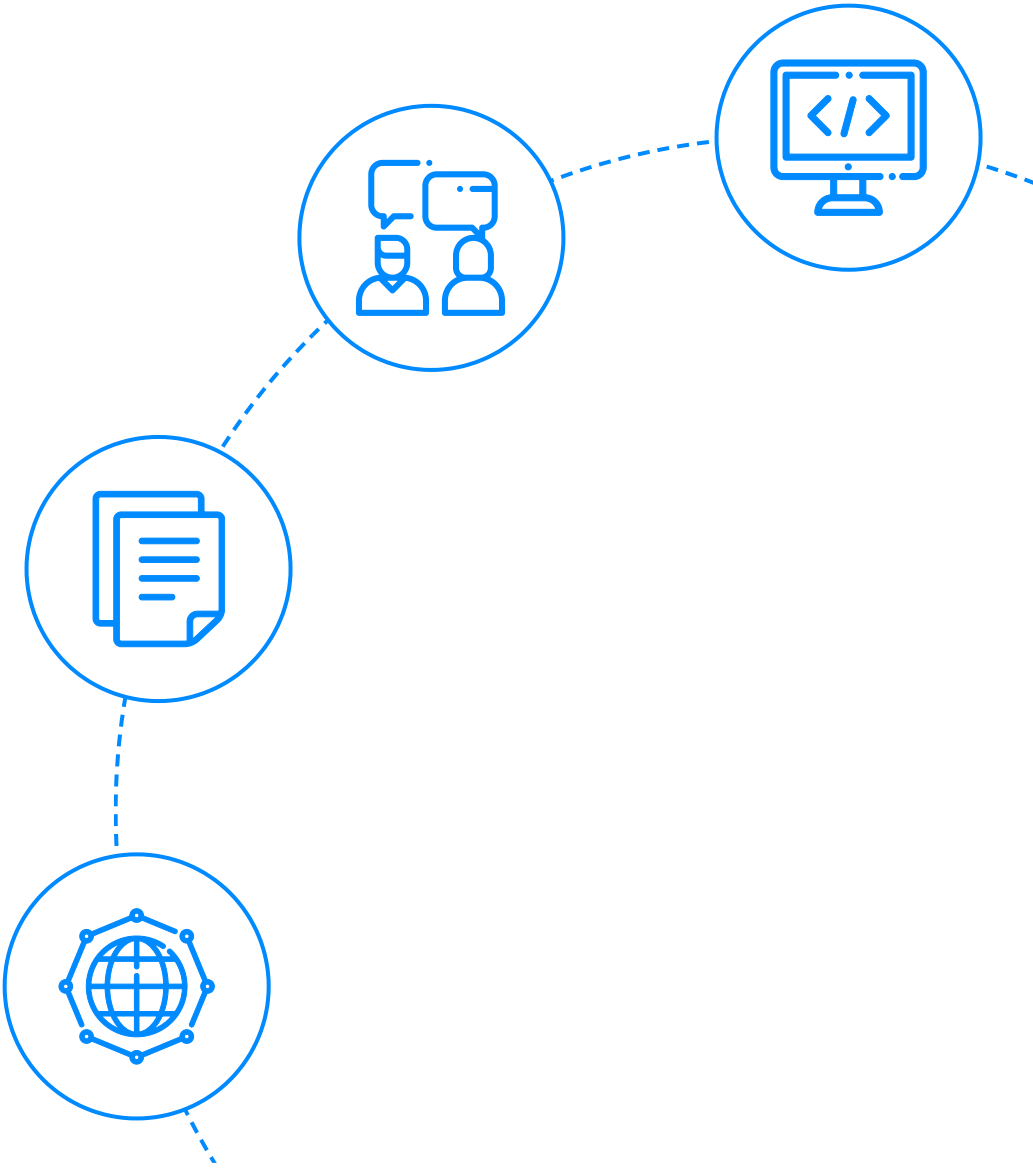
InterviewBit

Hadoop Interview Questions



To view the live version of the page, click here.

Contents

Hadoop Interview Questions for Freshers

Hadoop Interview Questions for Freshers (.....Continued)

21. Mention features of Apache sqoop.

Hadoop Interview Questions for Experienced

22. What is DistCp?
23. Why are blocks in HDFS huge?
24. What is the default replication factor?
25. How can you skip the bad records in Hadoop?
26. Where are the two types of metadata that NameNode server stores?
27. Which Command is used to find the status of the Blocks and File-system health?
28. Write the command used to copy data from the local system onto HDFS?
29. Explain the purpose of the dfsadmin tool?
30. Explain the actions followed by a Jobtracker in Hadoop.
31. Explain the distributed Cache in MapReduce framework.
32. List the actions that happen when a DataNode fails.
33. What are the basic parameters of a mapper?
34. Mention the main Configuration parameters that has to be specified by the user to run MapReduce.
35. Explain the Resilient Distributed Datasets in Spark.
36. Give a brief on how Spark is good at low latency workloads like graph processing and Machine Learning.
37. What applications are supported by Apache Hive?
38. Explain a metastore in Hive?

Hadoop Interview Questions for Experienced

(.....Continued)

39. Compare differences between Local Metastore and Remote Metastore
40. Are Multiline Comments supported in Hive? Why?
41. Why do we need to perform partitioning in Hive?
42. How can you restart NameNode and all the daemons in Hadoop?
43. How do you differentiate inner bag and outer bag in Pig.
44. If the source data gets updated every now and then, how will you synchronize the data in HDFS that is imported by Sqoop?
45. Where is table data stored in Apache Hive by default?
46. What is the default File format to import data using Apache sqoop?
47. What is the use of the -compress-codec parameter?
48. What is Apache Flume in Hadoop ?
49. Explain the architecture of Flume.
50. Mention the consequences of Distributed Applications.

Let's get Started

Apache Hadoop is an open-source software library used to control data processing and storage in big data applications. Hadoop helps to analyze vast amounts of data parallelly and more swiftly. Apache Hadoop was acquainted with the public in 2012 by The Apache Software Foundation(ASF). Hadoop is economical to use as data is stored on affordable commodity Servers that run as clusters.

Before the digital period, the volume of data gathered was slow and could be examined and stored with a single storage format. At the same time, the format of the data received for similar purposes had the same format. However, with the development of the Internet and digital platforms like social media, the data comes in multiple formats (structured, semi-structured, and unstructured), and its velocity also massively grown. A new name was given to this data which is Big data. Then, the need for multiple processors and storage units arose to handle the big data. Therefore, as a solution, Hadoop was introduced.

Hadoop Interview Questions for Freshers

1. Explain big data and list its characteristics.

Gartner defined Big Data as–
"*Big data" is high-volume, velocity, and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.*"

Simply, big data is larger, more complex data sets, particularly from new data sources. These data sets are so large that conventional data processing software can't manage them. But these massive volumes of data can be used to address business problems you wouldn't have been able to tackle before.

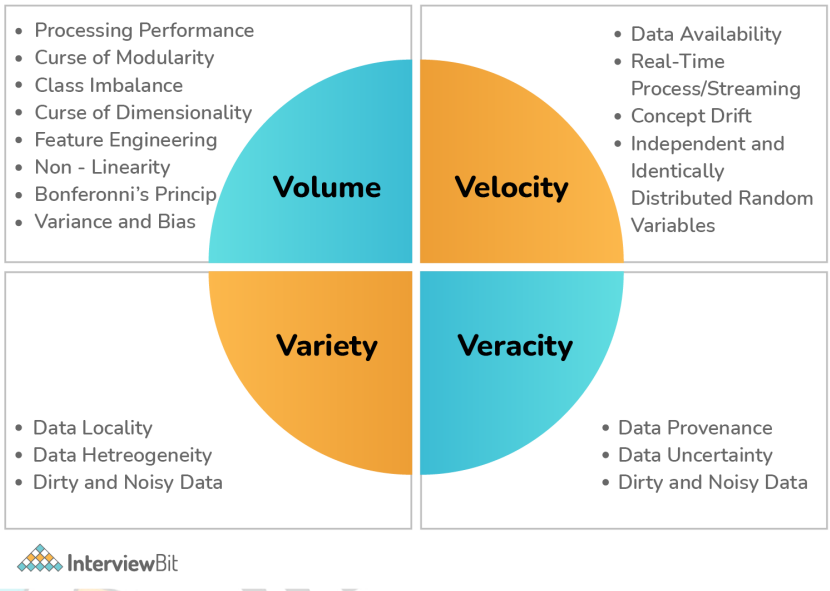


Image Source: ResearchGate

Characteristics of Big Data are:

- **Volume**: A large amount of data stored in data warehouses refers to Volume.
- **Velocity**: Velocity typically refers to the pace at which data is being generated in real-time.
- **Variety**: Variety of Big Data relates to structured, unstructured, and semistructured data that is collected from multiple sources.
- **Veracity**: Data veracity generally refers to how accurate the data is.
- **Value**: No matter how fast the data is produced or its amount, it has to be reliable and valuable. Otherwise, the information is not good enough for processing or analysis.

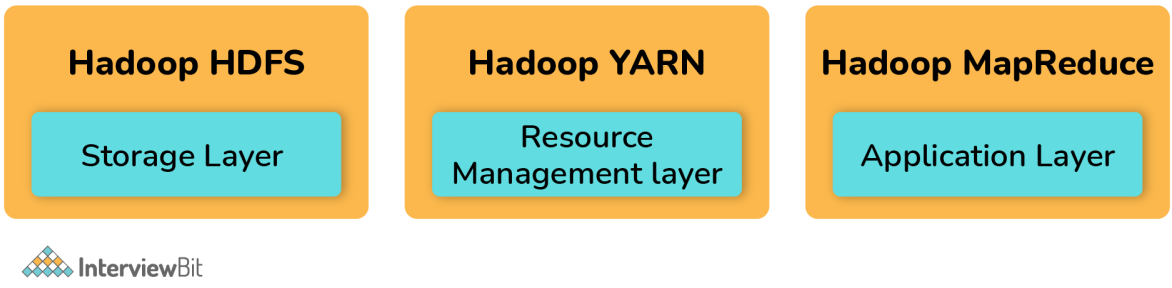
2. Explain Hadoop. List the core components of Hadoop

Hadoop is a famous big data tool utilized by many companies globally. Few successful Hadoop users:

- Uber
- The Bank of Scotland
- Netflix
- The National Security Agency (NSA) of the United States
- Twitter

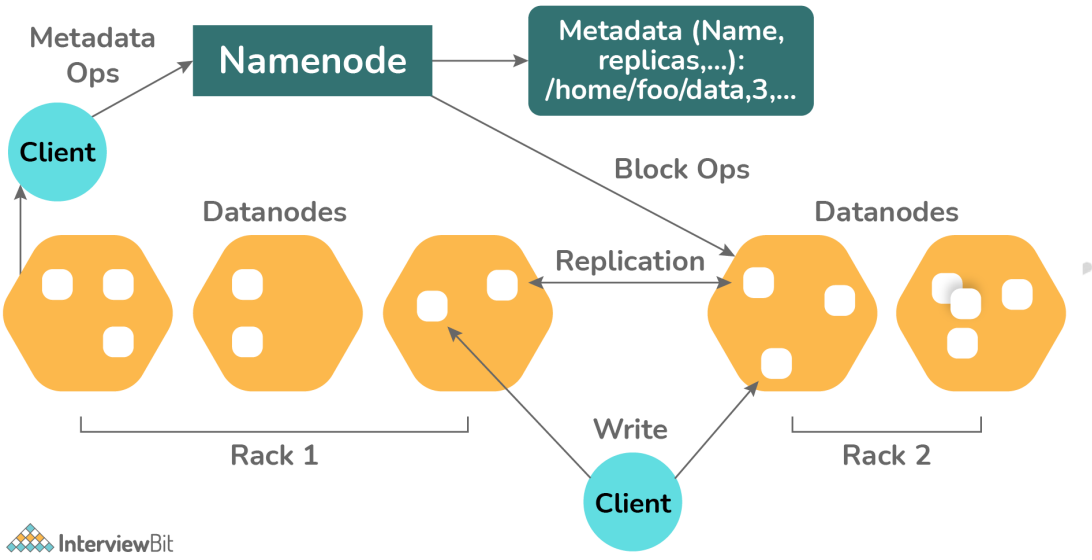
There are three components of Hadoop are:

1. **Hadoop YARN** - It is a resource management unit of Hadoop.
2. **Hadoop Distributed File System (HDFS)** - It is the storage unit of Hadoop.
3. **Hadoop MapReduce** - It is the processing unit of Hadoop.



3. Explain the Storage Unit In Hadoop (HDFS).

HDFS is the Hadoop Distributed File System, is the storage layer for Hadoop. The files in HDFS are split into block-size parts called data blocks. These blocks are saved on the slave nodes in the cluster. By default, the size of the block is 128 MB by default, which can be configured as per our necessities. It follows the master-slave architecture. It contains two daemons- DataNodes and NameNode.



NameNode

The NameNode is the master daemon that operates on the master node. It saves the filesystem metadata, that is, files names, data about blocks of a file, blocks locations, permissions, etc. It manages the Datanodes.

DataNode

The DataNodes are the slave daemon that operates on the slave nodes. It saves the actual business data. It serves the client read/write requests based on the NameNode instructions. It stores the blocks of the files, and NameNode stores the metadata like block locations, permission, etc.

4. Mention different Features of HDFS.

- **Fault Tolerance**
  Hadoop framework divides data into blocks and creates various copies of blocks on several machines in the cluster. So, when any device in the cluster fails, clients can still access their data from the other machine containing the exact copy of data blocks.
- **High Availability**
  In the HDFS environment, the data is duplicated by generating a copy of the blocks. So, whenever a user wants to obtain this data, or in case of an unfortunate situation, users can simply access their data from the other nodes because duplicate images of blocks are already present in the other nodes of the HDFS cluster.
- **High Reliability**
  HDFS splits the data into blocks, these blocks are stored by the Hadoop framework on nodes existing in the cluster. It saves data by generating a duplicate of every block current in the cluster. Hence presents a fault tolerance facility. By default, it creates 3 duplicates of each block containing information present in the nodes. Therefore, the data is promptly obtainable to the users. Hence the user does not face the difficulty of data loss. Therefore, HDFS is very reliable.
- **Replication**
  Replication resolves the problem of data loss in adverse conditions like device failure, crashing of nodes, etc. It manages the process of replication at frequent intervals of time. Thus, there is a low probability of a loss of user data.
- **Scalability**
  HDFS stocks the data on multiple nodes. So, in case of an increase in demand, it can scale the cluster.

5. What are the Limitations of Hadoop 1.0 ?

- Only one NameNode is possible to configure.
- Secondary NameNode was to take hourly backup of MetaData from NameNode.
- It is only suitable for Batch Processing of a vast amount of Data, which is already in the Hadoop System.
- It is not ideal for Real-time Data Processing.
- It supports up to 4000 Nodes per Cluster.
- It has a single component: JobTracker to perform many activities like Resource Management, Job Scheduling, Job Monitoring, Re-scheduling Jobs etc.
- JobTracker is the single point of failure.
- It supports only one Name No and One Namespace per Cluster.
- It does not help the Horizontal Scalability of NameNode.
- It runs only Map/Reduce jobs.

6. Compare the main differences between HDFS (Hadoop Distributed File System) and Network Attached Storage(NAS) ?

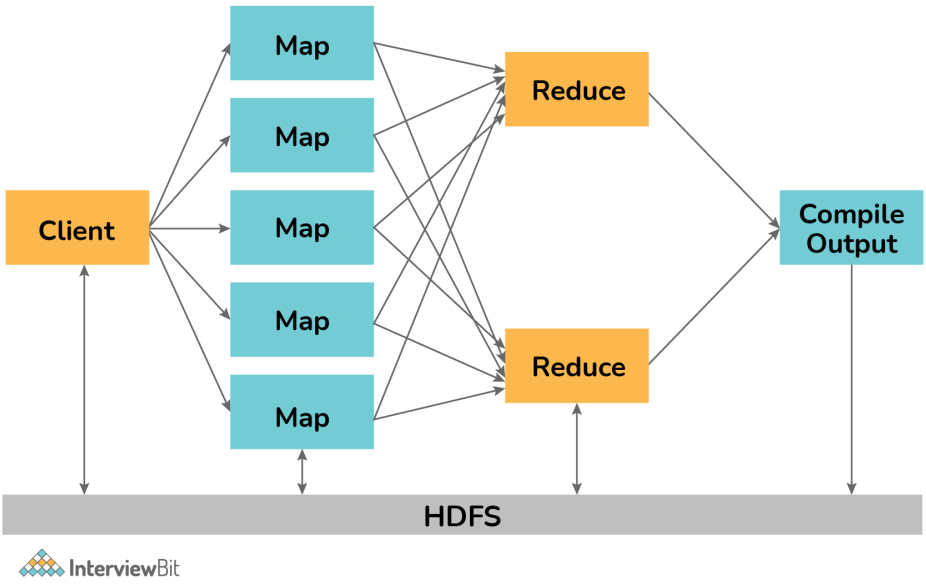
HDFS	NAS
HDFS is a Distributed File system that is mainly used to store data by commodity hardware.	NAS is a file-level computer data storage server connected to a computer network that provides network access to a heterogeneous group of clients.
HDFS is programmed to work with the MapReduce paradigm.	NAS is not suitable to work with a MapReduce paradigm.
HDFS is Cost-effective.	NAS is a high-end storage device that is highly expensive.

7. List Hadoop Configuration files.

Configuration Filenames	Description of log Files
hadoop-env.sh	Environment variables that are used in the scripts to run Hadoop.
core-site.xml	Configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.
hdfs-site.xml	Configuration settings for HDFS daemons, the namenode, the secondary namenode and the data nodes.
mapred-site.xml	Configuration settings for MapReduce daemons : the job-tracker and the task-trackers.
masters	A list of machines (one per line) that each run a secondary namenode.
slaves	A list of machines (one per line) that each run a datanode and a task-tracker.

8. Explain Hadoop MapReduce.

Hadoop MapReduce is a software framework for processing enormous data sets. It is the main component for data processing in the Hadoop framework. It divides the input data into several parts and runs a program on every data component parallel at one. The word MapReduce refers to two separate and different tasks.

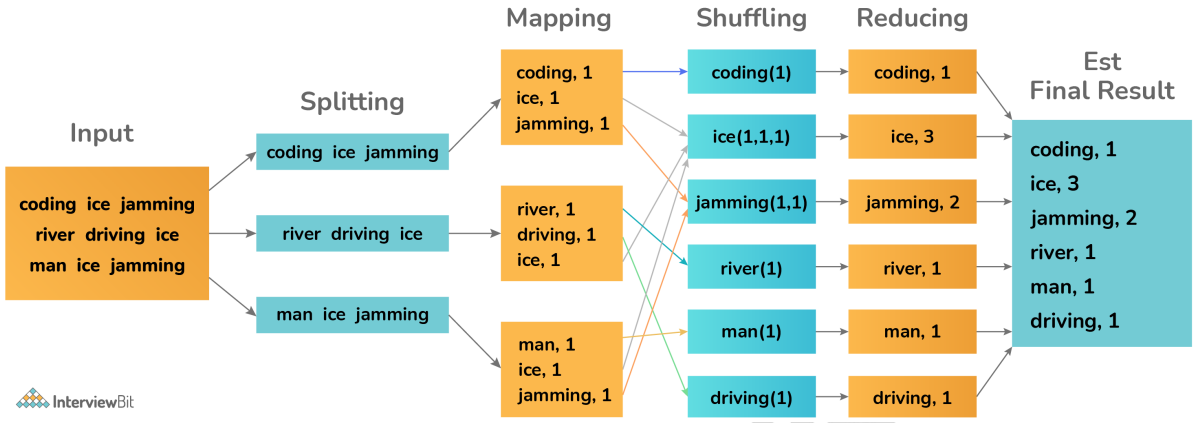


The first is the map operation, which takes a set of data and transforms it into a different collection of data, where individual elements are divided into tuples. The reduce operation consolidates those data tuples based on the key and subsequently modifies the value of the key.

Let us take an example of a text file called example_data.txt and understand how MapReduce works.

The content of the example_data.txt file is:
coding,jamming,ice,river,man,driving

Now, assume we have to find out the word count on the example_data.txt using MapReduce. So, we will be looking for the unique words and the number of times those unique words appeared.



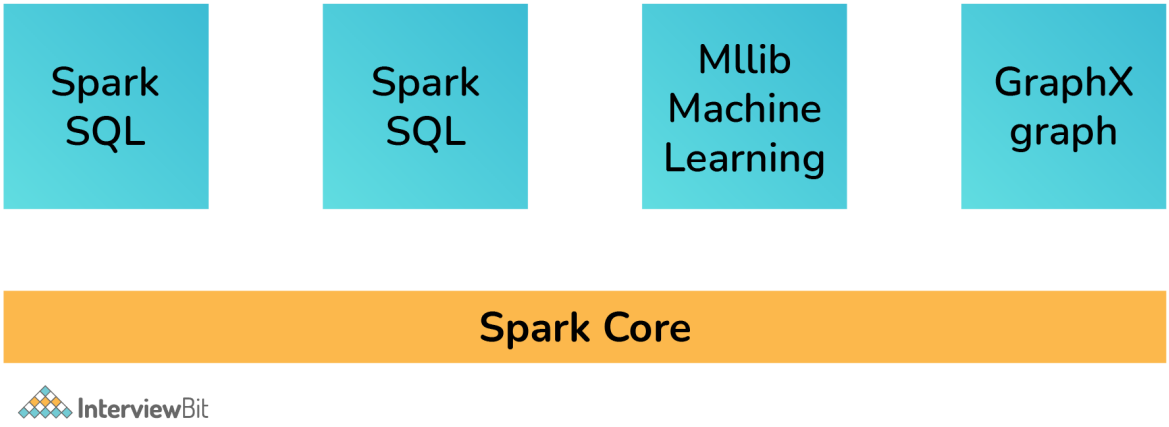
- First, we break the input into three divisions, as seen in the figure. This will share the work among all the map nodes.
- Then, all the words are tokenized in each of the mappers, and a hardcoded value (1) to each of the tokens is given. The reason behind giving a hardcoded value equal to 1 is that every word by itself will, at least, occur once.
- Now, a list of key-value pairs will be created where the key is nothing but the individual words and value is one. So, for the first line (Coding Ice Jamming), we have three key-value pairs – Coding, 1; Ice, 1; Jamming, 1.
- The mapping process persists the same on all the nodes.
- Next, a partition process occurs where sorting and shuffling follow so that all the tuples with the same key are sent to the identical reducer.
- Subsequent to the sorting and shuffling phase, every reducer will have a unique key and a list of values matching that very key. For example, Coding, [1,1]; Ice, [1,1,1].., etc.
- Now, each Reducer adds the values which are present in that list of values. As shown in the example, the reducer gets a list of values [1,1] for the key Jamming. Then, it adds the number of ones in the same list and gives the final output as – Jamming, 2.
- Lastly, all the output key/value pairs are then assembled and written in the output file.

9. What is shuffling in MapReduce?

In Hadoop MapReduce, shuffling is used to transfer data from the mappers to the important reducers. It is the process in which the system sorts the unstructured data and transfers the output of the map as an input to the reducer. It is a significant process for reducers. Otherwise, they would not accept any information. Moreover, since this process can begin even before the map phase is completed, it helps to save time and complete the process in a lesser amount of time.

10. List the components of Apache Spark.

Apache Spark comprises the Spark Core Engine, Spark Streaming, MLlib, GraphX, Spark SQL, and Spark R.



The Spark Core Engine can be used along with any of the other five components specified. It is not required to use all the Spark components collectively. Depending on the use case and request, one or more can be used along with Spark Core.

11. What are the three modes that hadoop can Run?

- **Local Mode or Standalone Mode**
  Hadoop, by default, is configured to run in a no distributed mode. It runs as a single Java process. Instead of HDFS, this mode utilizes the local file system. This mode is more helpful for debugging, and there isn't any requirement to configure core-site.xml, hdfs-site.xml, mapred-site.xml, masters & slaves. Stand-alone mode is ordinarily the quickest mode in Hadoop.
- **Pseudo-distributed Model**
  In this mode, each daemon runs on a separate java process. This mode requires custom configuration (core-site.xml, hdfs-site.xml, mapred-site.xml). The HDFS is used for input and output. This mode of deployment is beneficial for testing and debugging purposes.
- **Fully Distributed Mode**
  It is the production mode of Hadoop. Basically, one machine in the cluster is designated as NameNode and another as Resource Manager exclusively. These are masters. Rest nodes act as Data Node and Node Manager. These are the slaves. Configuration parameters and environment need to be defined for Hadoop Daemons. This mode gives fully distributed computing capacity, security, fault endurance, and scalability.

12. What is an Apache Hive?

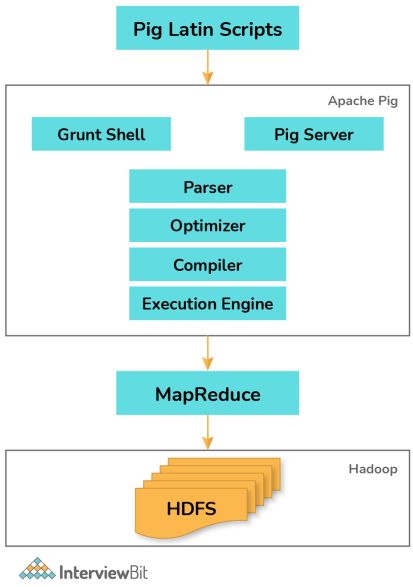
Hive is an open-source system that processes structured data in Hadoop, living on top of the latter for summing Big Data and facilitating analysis and queries. In addition, hive enables SQL developers to write Hive Query Language statements similar to standard SQL statements for data query and analysis. It is created to make MapReduce programming easier because you don't know and write lengthy Java code.

13. What is Apache Pig?

MapReduce needs programs to be translated into map and reduce stages. As not all data analysts are accustomed to MapReduce, Yahoo researchers introduced Apache pig to bridge the gap. Apache Pig was created on top of Hadoop, producing a high level of abstraction and enabling programmers to spend less time writing complex MapReduce programs.

14. Explain the Apache Pig architecture.

Apache Pig architecture includes a Pig Latin interpreter that applies Pig Latin scripts to process and interpret massive datasets. Programmers use Pig Latin language to examine huge datasets in the Hadoop environment. Apache pig has a vibrant set of datasets showing different data operations like join, filter, sort, load, group, etc. Programmers must practice Pig Latin language to address a Pig script to perform a particular task. Pig transforms these Pig scripts into a series of Map-Reduce jobs to reduce programmers' work. Pig Latin programs are performed via various mechanisms such as UDFs, embedded, and Grunt shells.



Apache Pig architecture consists of the following major components:

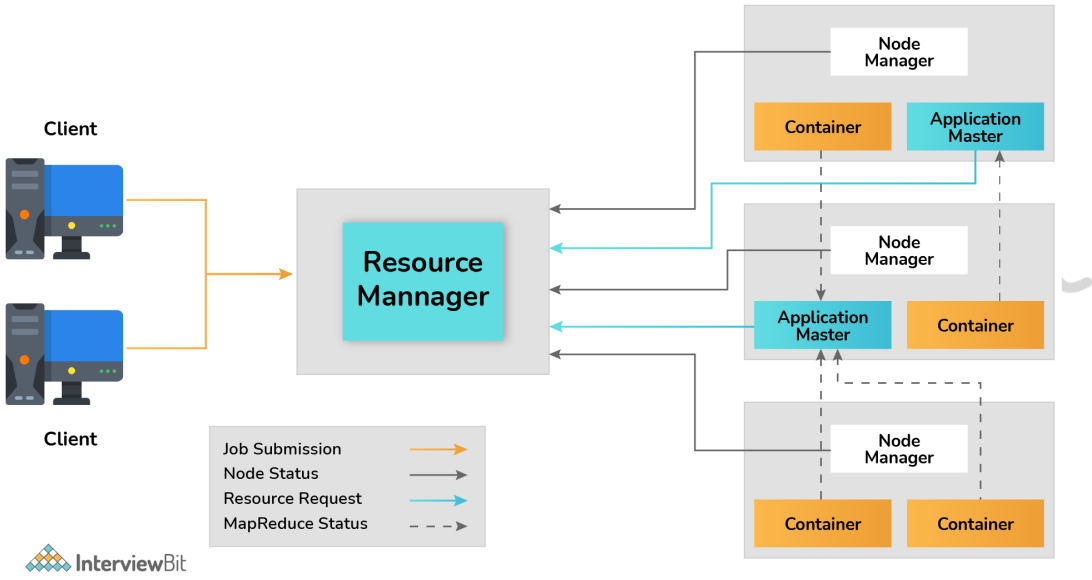
- Parser: The Parser handles the Pig Scripts and checks the syntax of the script.
- Optimizer: The optimizer receives the logical plan (DAG). And carries out the logical optimization such as projection and push down.
- Compiler: The compiler converts the logical plan into a series of MapReduce jobs.
- Execution Engine: In the end, the MapReduce jobs get submitted to Hadoop in sorted order.
- Execution Mode: Apache Pig is executed in local and Map Reduce modes. The selection of execution mode depends on where the data is stored and where you want to run the Pig script.

15. What is Yarn?

Yarn stands for Yet Another Resource Negotiator. It is the resource management layer of Hadoop. The Yarn was launched in Hadoop 2.x. Yarn provides many data processing engines like graph processing, batch processing, interactive processing, and stream processing to execute and process data saved in the Hadoop Distributed File System. Yarn also offers job scheduling. It extends the capability of Hadoop to other evolving technologies so that they can take good advantage of HDFS and economic clusters.
Apache Yarn is the data operating method for Hadoop 2.x. It consists of a master daemon known as “Resource Manager,” a slave daemon called node manager, and Application Master.

16. List the YARN components.

- **Resource Manager**: It runs on a master daemon and controls the resource allocation in the cluster.
- **Node Manager**: It runs on the slave daemons and executes a task on each single Data Node.
- **Application Master**: It controls the user job lifecycle and resource demands of single applications. It works with the Node Manager and monitors the execution of tasks.
- **Container**: It is a combination of resources, including RAM, CPU, Network, HDD, etc., on a single node.



17. What is Apache ZooKeeper?

Apache Zookeeper is an open-source service that supports controlling a huge set of hosts. Management and coordination in a distributed environment are complex. Zookeeper automates this process and enables developers to concentrate on building software features rather than bother about its distributed nature.

Zookeeper helps to maintain configuration knowledge, naming, group services for distributed applications. It implements various protocols on the cluster so that the application should not execute them on its own. It provides a single coherent view of many machines.

18. What are the Benefits of using zookeeper?

- **Simple distributed coordination process**: The coordination process among all nodes in Zookeeper is straightforward.
- **Synchronization**: Mutual exclusion and co-operation among server processes.
- **Ordered Messages**: Zookeeper tracks with a number by denoting its order with the stamping of each update; with the help of all this, messages are ordered here.
- **Serialization**: Encode the data according to specific rules. Ensure your application runs consistently.
- **Reliability**: The zookeeper is very reliable. In case of an update, it keeps all the data until forwarded.
- **Atomicity**: Data transfer either succeeds or fails, but no transaction is partial.

19. Mention the types of Znode.

- **Persistent Znodes:**
  The default znode in ZooKeeper is the Persistent Znode. It permanently stays in the zookeeper server until any other clients leave it apart.
- **Ephemeral Znodes:**
  These are the temporary znodes. It is smashed whenever the creator client logs out of the ZooKeeper server. For example, assume client1 created eznode1. Once client1 logs out of the ZooKeeper server, the eznode1 gets destroyed.
- **Sequential Znodes:**
  Sequential znode is assigned a 10-digit number in numerical order at the end of its name. Assume client1 produced a sznode1. In the ZooKeeper server, the sznode1 will be named like this:
  sznode0000000001
  If the client1 generates another sequential znode, it will bear the following number in a sequence. So the subsequent sequential znode is <znode name>0000000002.

20. List Hadoop HDFS Commands.

A)**version**: hadoop version

```
interviewbit:~$ hadoop version
Hadoop 3.1.2
Source code repository https://github.com/apache/hadoop.git -r
Compiled by sunlig on 2019-01-29T01:39Z
interviewbit:~$
```

B) **mkdir**: Used to create a new directory.

```
interviewbit:~$ hadoop FD -mkdir/interviewbit
interviewbit:~$
```

C) **cat**: We are using the cat command to display the content of the file present in the directory of HDFS.
hadoop fs –cat /path_to_file_in_hdfs

```
interviewbit:~$ hadoop fs -cat/interviewbit/sample
Hello from InterviewBit…
File in HDFS …
interviewbit:~$
```

D)**mv** : The HDFS mv command moves the files or directories from the source to a destination within HDFS.
hadoop fs -mv <src> <dest>

```
interviewbit:~$ hadoop fs -ls/
Found 2 Items
drwxv -xv -x - interviewbit supergroup 0 2020-01-29:11:11/ Intr1
drwxv -xv -x - interviewbit supergroup 0 2020-01-29:11:11/ Interviewbit
interviewbit:~$ hadoop fs -mv/ Intr1/ Interviewbit
interviewbit:~$ hadoop fs -ls/
Found 1 Item
drwxv -xv -x - interviewbit supergroup 0 2020-01-29:11:11/ Interviewbit
```

E) **copyToLocal**: This command copies the file from the file present in the newDataFlair directory of HDFS to the local file system.
hadoop fs -copyToLocal <hdfs source> <localdst>

```
interviewbit:~$ hadoop fs -copyFromLocal -/test1/interviewbit/CopyTest
interviewbit:~$
```

F) **get**: Copies the file from the Hadoop File System to the Local File System.
hadoop fs -get<src> <localdest>

```
interviewbit:~$ hadoop fs - get/testFile ~/copyFromHadoop
interviewbit:~$
```

21. Mention features of Apache sqoop.

- **Robust**: It is highly robust. It even has community support and contribution and is easily usable.
- **Full Load**: Sqoop can load the whole table just by a single Sqoop command. It also allows us to load all the tables of the database by using a single Sqoop command.
- **Incremental Load**: It supports incremental load functionality. Using Sqoop, we can load parts of the table whenever it is updated.
- **Parallel import/export**: It uses the YARN framework for importing and exporting the data. That provides fault tolerance on the top of parallelism.
- **Import results of SQL query**: It allows us to import the output from the SQL query into the Hadoop Distributed File System.

Hadoop Interview Questions for Experienced

22. What is DistCp?

It is a tool that is used for copying a very large amount of data to and from Hadoop file systems in parallel. It uses MapReduce to affect its distribution, error handling, recovery, and reporting. It expands a list of files and directories into input to map tasks, each of which will copy a partition of the files specified in the source list.

23. Why are blocks in HDFS huge?

By default, the size of the HDFS data block is 128 MB. The ideas for the large size of blocks are:

- To reduce the expense of seek: Because of the large size blocks, the time consumed to shift the data from the disk can be longer than the usual time taken to commence the block. As a result, the multiple blocks are transferred at the disk transfer rate.
- If there are small blocks, the number of blocks will be too many in Hadoop HDFS and too much metadata to store. Managing such a vast number of blocks and metadata will create overhead and head to traffic in a network.

24. What is the default replication factor?

By default, the replication factor is 3. There are no two copies that will be on the same data node. Usually, the first two copies will be on the same rack, and the third copy will be off the shelf. It is advised to set the replication factor to at least three so that one copy is always safe, even if something happens to the rack.
We can set the default replication factor of the file system as well as of each file and directory exclusively. For files that are not essential, we can lower the replication factor, and critical files should have a high replication factor.

25. How can you skip the bad records in Hadoop?

Hadoop provides an option where a particular set of lousy input records can be skipped when processing map inputs. Applications can manage this feature through the SkipBadRecords class.
This feature can be used when map tasks fail deterministically on a particular input. This usually happens due to faults in the map function. The user would have to fix these issues.

26. Where are the two types of metadata that NameNode server stores?

The two types of metadata that NameNode server stores are in Disk and RAM. Metadata is linked to two files which are:

- **EditLogs**: It contains all the latest changes in the file system regarding the last FsImage.
- **FsImage**: It contains the whole state of the namespace of the file system from the origination of the NameNode.

Once the file is deleted from HDFS, the NameNode will immediately store this in the EditLog.
All the file systems and metadata which are present in the Namenode's Ram are read by the Secondary NameNode continuously and later get recorded into the file system or hard disk. EditLogs is combined with FsImage in the NameNode. Periodically, Secondary NameNode downloads the EditLogs from the NameNode, and then it is implemented to FsImage. The new FsImage is then copied back into the NameNode and used only after the NameNode has started the subsequent time.

27. Which Command is used to find the status of the Blocks and File-system health?

The command used to find the status of the block is: **hdfs fsck <path> -files –blocks**
And the command used to find File-system health is: **hdfs fsck/ -files –blocks –locations > dfs-fsck.log**

28. Write the command used to copy data from the local system onto HDFS?

The command used for copying data from the Local system to HDFS is: **hadoop fs –copyFromLocal [source][destination]**

29. Explain the purpose of the dfsadmin tool?

The dfsadmin tools are a specific set of tools designed to help you root out information about your Hadoop Distributed File system (HDFS). As a bonus, you can use them to perform some administration operations on HDFS as well.

Option	What It Does
-report	Reports basic file system information and statistics.
-safemode enter I leave I get I wait	Manages safe mode, a NameNode state in which changes to the name space are not accepted and blocks can be neither replicated nor deleted. The NameNode is in safe mode during start-up so that it doesn't prematurely start replicating blocks even though there are already enough replicas in the cluster.
-refreshNodes	Forces the NameNode to reread its configuration, including the dfs.hosts.exclude file. The NameNode decommissions nodes after their blocks have been replicated onto machines that will remain active.
-finalizeUpgrade	Completes the HDFS upgrade process. DataNodes and the NameNode delete working directories from the previous version.
-upgradeProgress status I details I force	Requests the standard or detailed current status of the distributed upgrade, or forces the upgrade to proceed.

InterviewBit

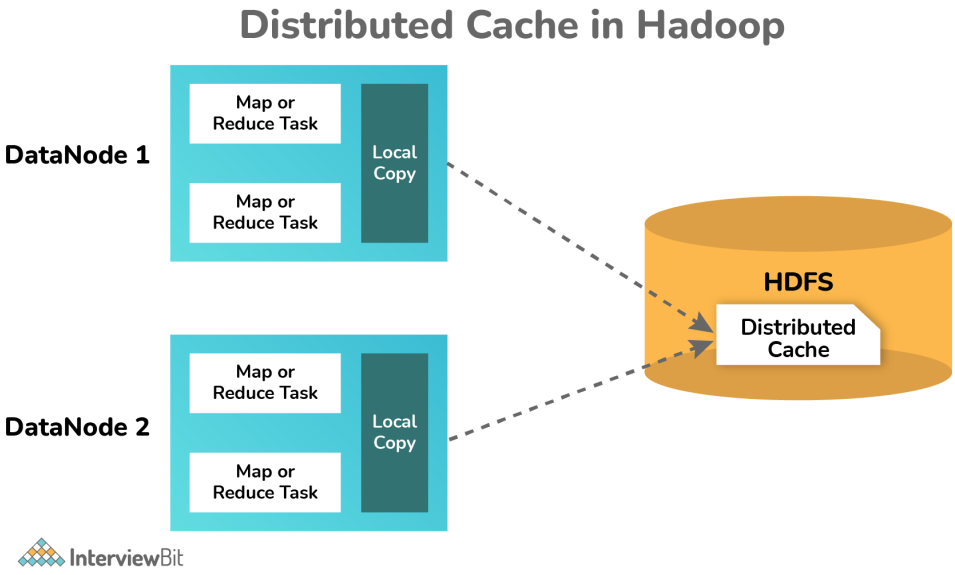
30. Explain the actions followed by a Jobtracker in Hadoop.

- The client application is used to submit the jobs to the Jobtracker.
- The JobTracker associates with the NameNode to determine the data location.
- With the help of available slots and the near the data, JobTracker locates TaskTracker nodes.
- It submits the work on the selected TaskTracker Nodes.
- When a task fails, JobTracker notifies and decides the further steps.
- JobTracker monitors the TaskTracker nodes.

31. Explain the distributed Cache in MapReduce framework.

Distributed Cache is a significant feature provided by the MapReduce Framework, practiced when you want to share the files across all nodes in a Hadoop cluster. These files can be jar files or simple properties files.

Hadoop's MapReduce framework allows the facility to cache small to moderate read-only files such as text files, zip files, jar files, etc., and distribute them to all the Datanodes(worker-nodes) MapReduce jobs are running. All Datanode gets a copy of the file(local-copy), which is sent by Distributed Cache.



32. List the actions that happen when a DataNode fails.

- Both the Jobtracker and the name node detect the failure on which blocks were the DataNode failed.
- On the failed node all the tasks are rescheduled by locating other DataNodes with copies of these blocks
- User's data will be replicated to another node from namenode to maintain the configured replication factor.

33. What are the basic parameters of a mapper?

The primary parameters of a mapper are text, LongWritable, text, and IntWritable. The initial two represent input parameters, and the other two signify intermediate output parameters.

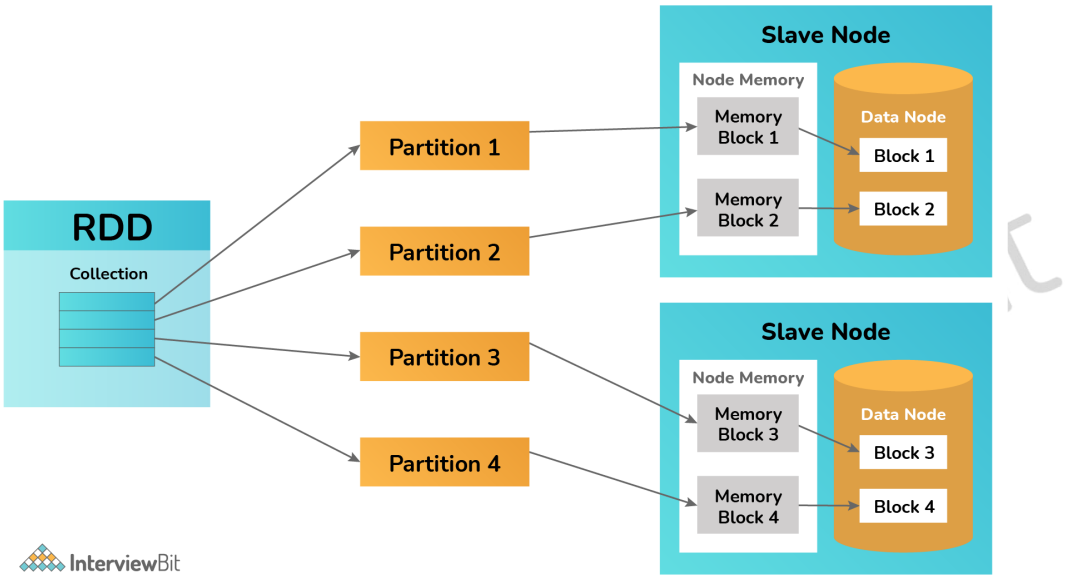
34. Mention the main Configuration parameters that has to be specified by the user to run MapReduce.

The chief configuration parameters that the user of the MapReduce framework needs to mention is:

- Job's input Location
- Job's Output Location
- The Input format
- The Output format
- The Class including the Map function
- The Class including the reduce function
- JAR file, which includes the mapper, the Reducer, and the driver classes.

35. Explain the Resilient Distributed Datasets in Spark.

Resilient Distributed Datasets is the basic data structure of Apache Spark. It is installed in the Spark Core. They are immutable and fault-tolerant. RDDs are generated by transforming already present RDDs or storing an outer dataset from well-built storage like HDFS or HBase.



Since they have distributed collections of objects, they can be operated in parallel. Resilient Distributed Datasets are divided into parts such that they can be executed on various nodes of a cluster.

36. Give a brief on how Spark is good at low latency workloads like graph processing and Machine Learning.

The data is stored in memory by Apache Spark for faster processing and development of machine learning models, which may need a lot of Machine Learning algorithms for multiple repetitions and various conceptual steps to create an optimized model. In the case of Graph algorithms, it moves within all the nodes and edges to make a graph. These low latency workloads, which need many iterations, enhance the performance.

37. What applications are supported by Apache Hive?

The applications that are supported by Apache Hive are,

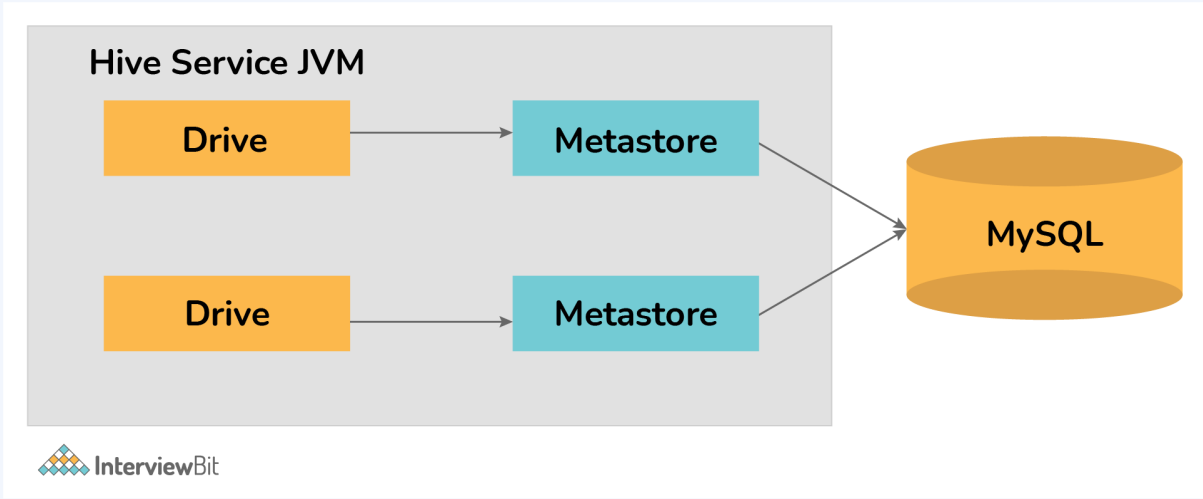
- Java
- PHP
- Python
- C++
- Ruby

38. Explain a metastore in Hive?

Metastore is used to store the metadata information; it's also possible to use RDBMS and the open-source ORM layer, converting object Representation into a relational schema. It's the central repository of Apache Hive metadata. It stores metadata for Hive tables (similar to their schema and location) and partitions in a relational database. It gives the client access to this information by using metastore service API. Disk storage for the Hive metadata is separate from HDFS storage.

39. Compare differences between Local Metastore and Remote Metastore

Local Metastore
Local metastore is a metastore service that runs in the same JVM in which the Hive service is running.
It can also connect to a separate database running in a separate JVM in the same or separate machine.

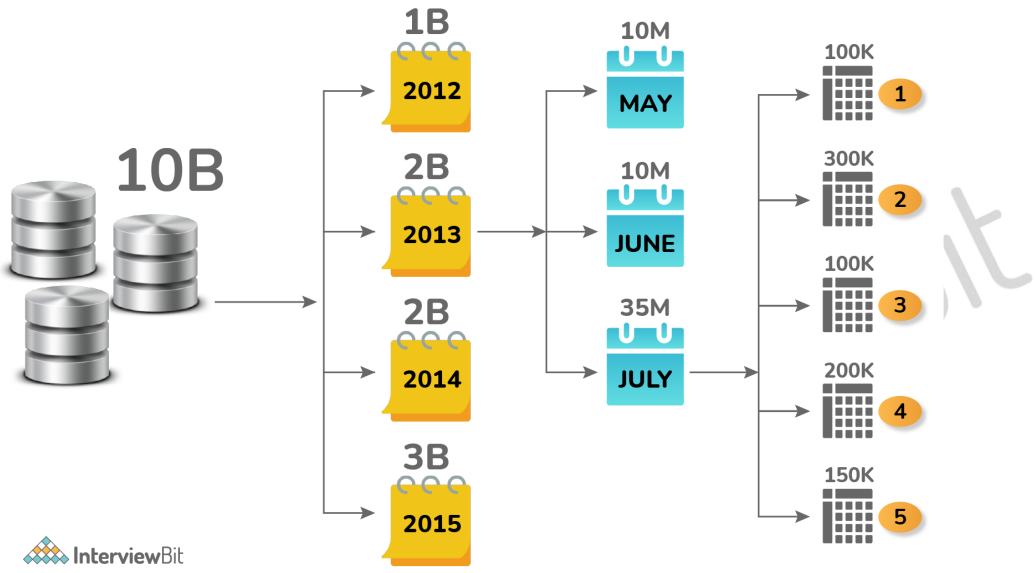


40. Are Multiline Comments supported in Hive? Why?

No, as of now multiline comments are not supported in Hive, only single-line comments are supported.

41. Why do we need to perform partitioning in Hive?

Apache Hive organizes tables into partitions. Partitioning is the manner in which a table is split into related components depending on the values of appropriate columns like date, city, and department.



Every table in the hive can have one or more than one partition keys to recognize a distinct partition. With the help of partitions, it is effortless to do queries on slices of the data.

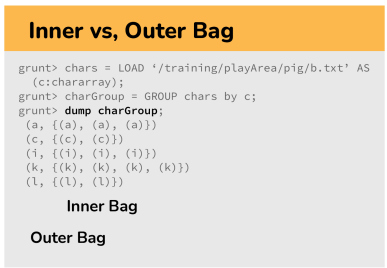
42. How can you restart NameNode and all the daemons in Hadoop?

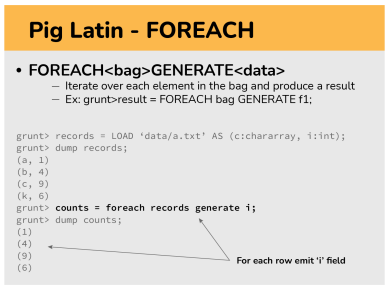
The following commands will help you restart NameNode and all the daemons:

- You can stop the NameNode with **./sbin /Hadoop-daemon.sh** stop NameNode command and then start the NameNode using **./sbin/Hadoop-daemon.sh** start NameNode command.
- You can stop all the daemons with the **./sbin /stop-all.sh** command and then start the daemons using the **./sbin/start-all.sh** command.

43. How do you differentiate inner bag and outer bag in Pig.

Inner Bag	Outer Bag
An inner bag just Contains a bag inside a tuple.	An outer bag which is also called a relation is nothing but a bag of tuples.
Example : (4,{(4,2,1), (4,3,3,)}) In this example the complete relation is an outer bag and {(4,2,1), (4,3,3,)} is an inner bag.	Example:{(park, New York), (Hollywood, Los Angeles)} Which is a bag of tuples, nothing but an outer bag.
An inner bag is a relation inside any other bag.	In an outer bag, relations are similar to relations in relational databases.





44. If the source data gets updated every now and then, how will you synchronize the data in HDFS that is imported by Sqoop?

If the source data gets updated in a very short interval of time, the synchronization of data in HDFS that is imported by Sqoop is done with the help of incremental parameters.

We should use incremental import along with the append choice even when the table is refreshed continuously with new rows. Principally where values of a few columns are examined, and if it encounters any revised value for those columns, only a new row will be inserted. Similar to incremental import, the origin has a date column examined for all the records that have been modified after the last import, depending on the previous revised column in the beginning. The values would be modernized.

45. Where is table data stored in Apache Hive by default?

By default,the is table data in Apache Hive is stored in:
Hdfs://namenode_server/user/hive/warehouse

46. What is the default File format to import data using Apache sqoop?

There are basically two file formats sqoop allos to import data they are:

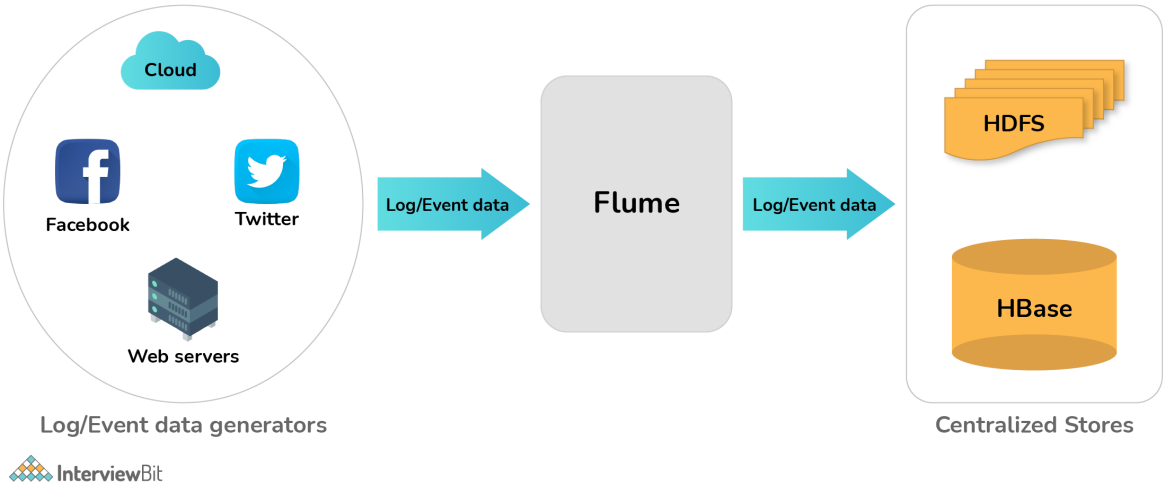
- Delimited Text File format
- Sequence File Format

47. What is the use of the -compress-codec parameter?

-compress-codec parameter is generally used to get the output file of a sqoop import in formats other than .gz.

48. What is Apache Flume in Hadoop ?

Apache Flume is a tool/service/data ingestion mechanism for assembling, aggregating, and carrying huge amounts of streaming data such as record files,

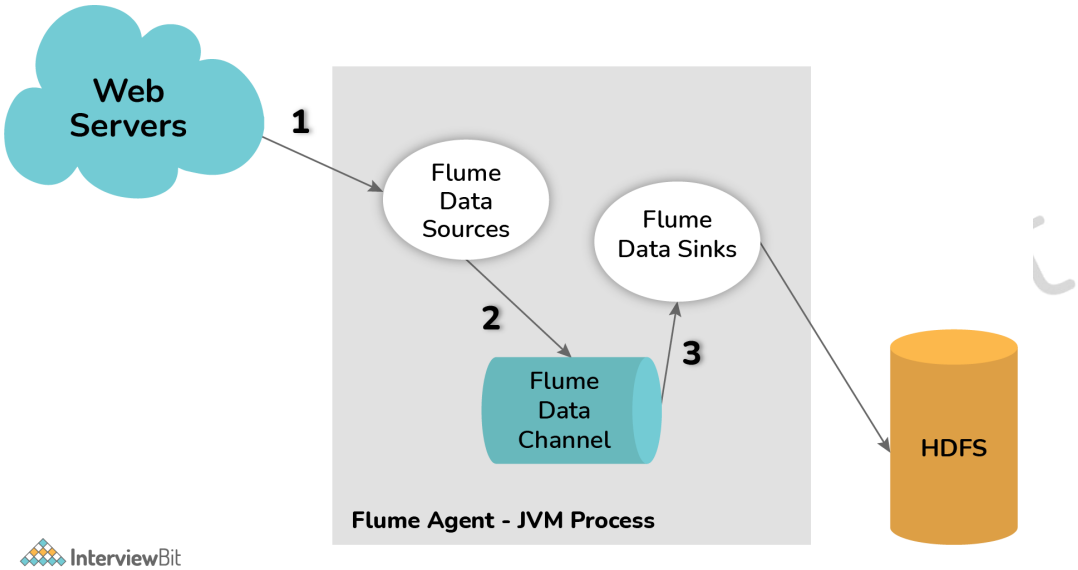


Flume is a very stable, distributed, and configurable tool. It is generally designed to copy streaming data (log data) from various web servers to HDFS.

49. Explain the architecture of Flume.

In general Apache Flume architecture is composed of the following components:

1. Flume Source
2. Flume Channel
3. Flume Sink
4. Flume Agent
5. Flume Event



1. **Flume Source**: Flume Source is available on various networking platforms like Facebook or Instagram. It is a Data generator that collects data from the generator, and then the data is transferred to a Flume Channel in the form of a Flume.
2. **Flume Channel**: The data from the flume source is sent to an Intermediate Store which buffers the events till they get transferred into Sink. The Intermediate Store is called Flume Channel. Channel is an intermediate source. It is a bridge between Source and a Sink Flume channel. It supports both the Memory channel and File channel. The file channel is non-volatile which means once the data is entered into the channel, the data will never be lost unless you delete it. In contrast, in the Memory channel, events are stored in memory, so it's volatile, meaning data may be lost, but Memory Channel is very fast in nature.
3. **Flume sink**: Data repositories like HDFS, have Flume Sink. Which takes Flume events from the Flume Channel and stores them into the Destination specified like HDFS. It is done in such a way where it should deliver the events to the Store or another agent. Various sinks like Hive Sink, Thrift Sink, etc are supported by the Flume.
4. **Flume Agent**: A Java process that works on Source, Channel, Sink combination is called Flume Agent. One or more than one agent is possible in Flume. Connected Flume agents which are distributed in nature can also be collectively called Flume.
5. **Flume Event**: An Event is the unit of data transported in Flume. The general representation of the Data Object in Flume is called Event. The event is made up of a payload of a byte array with optional headers.

50. Mention the consequences of Distributed Applications.

- **Heterogeneity**: The design of applications should allow the users to access services and run applications over a heterogeneous collection of computers and networks taking into consideration Hardware devices, OS, networks, Programming languages.
- **Transparency**: Distributed system Designers must hide the complexity of the system as much as they can. Some Terms of transparency are location, access, migration, Relocation, and so on.
- **Openness**: It is a characteristic that determines whether the system can be extended and reimplemented in various ways.
- **Security**: Distributed system Designers must take care of confidentiality, integrity, and availability.
- **Scalability**: A system is said to be scalable if it can handle the addition of users and resources without suffering a noticeable loss of performance.

Recommended Resource:
Spark Interview

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions