



INTERVIEW QUESTIONS AND ANSWERS

1. What is the difference between Hadoop and Traditional RDBMS?

Criteria	Hadoop	RDBMS
Datatypes	Processes semi-structured and unstructured data.	Processes structured data.
Schema	Schema on Read	Schema on Write
Best Fit for Applications	Data discovery and Massive Storage/Processing of Unstructured data.	Best suited for OLTP and complex ACID transactions.
Speed	Writes are Fast	Reads are Fast

RDBMS vs. Hadoop

	RDBMS	HADOOP
Data size	Gigabytes (<i>Terabytes</i>)	Petabytes (<i>Hexabytes</i>)
Access	Interactive and Batch	Batch
Updates	Read / Write many times	Hadoop vs RDBMS times
Structure	Static Schema	Dynamic Schema
Integrity	High (ACID)	Low
Scaling	Nonlinear 	Linear 

2. What do the four V's of Big Data denote?

IBM has a nice, simple explanation for the four critical features of big data:

- a) Volume –Scale of data
- b) Velocity –Analysis of streaming data
- c) Variety – Different forms of data
- d) Veracity –Uncertainty of data

Here is an explanatory video on the four V's of Big Data

3. How big data analysis helps businesses increase their revenue? Give example.

Big data analysis is helping businesses differentiate themselves – for example Walmart the world's largest retailer in 2014 in terms of revenue - is using big data analytics to increase its sales through better predictive analytics, providing customized recommendations and launching new products based on customer preferences and needs. Walmart observed a significant 10% to 15% increase in online sales for \$1 billion in incremental revenue. There are many more companies like Facebook, Twitter, LinkedIn, Pandora, JPMorgan Chase, Bank of America, etc. using big data analytics to boost their revenue.

4. Name some companies that use Hadoop.

- Yahoo (One of the biggest user & more than 80% code contributor to Hadoop)
- Facebook
- Netflix
- Amazon
- Adobe

- eBay
- Hulu
- Spotify
- Rubikloud
- Twitter

5. Differentiate between Structured and Unstructured data.

Data that can be stored in traditional database systems in the form of rows and columns, for example, the online purchase transactions can be referred to as Structured Data. Data that can be stored only partially in traditional database systems, for example, data in XML records can be referred to as semi-structured data. Unorganized and raw data that cannot be categorized as semi-structured or structured data is referred to as unstructured data. Facebook updates, tweets on Twitter, Reviews, weblogs, etc. are all examples of unstructured data.

Structured data: Schema-based data, Datastore in SQL, Postgresql databases etc

Semi-structured data : Json objects , json arrays, csv , txt ,xlsx files,web logs ,tweets etc

Unstructured data : Audio, Video files, etc

6. On what concept the Hadoop framework works?

Hadoop Framework works on the following two core components-

1)HDFS – Hadoop Distributed File System is the java based file system for scalable and reliable storage of large datasets. Data in HDFS is stored in the form of blocks and it operates on the Master-Slave Architecture.

2)Hadoop MapReduce-This is a java based programming paradigm of the Hadoop framework that provides scalability across various Hadoop clusters. MapReduce distributes the workload into various tasks that can run in parallel. Hadoop jobs perform 2 separate tasks- job. The map job breaks down the data sets into key-value pairs or tuples. The reduce job then takes the output of the map job and combines the data tuples to into smaller set of tuples. The reduce job is always performed after the map job is executed.

7. What are the main components of a Hadoop Application?

Hadoop applications have wide range of technologies that provide great advantage in solving complex business problems.

Core components of a Hadoop application are-

- 1) Hadoop Common
- 2) HDFS
- 3) Hadoop MapReduce
- 4) YARN

Data Access Components are - Pig and Hive

Data Storage Component is - HBase

Data Integration Components are - Apache Flume, Sqoop, Chukwa

Data Management and Monitoring Components are - Ambari, Oozie and Zookeeper.

Data Serialization Components are - Thrift and Avro

Data Intelligence Components are - Apache Mahout and Drill.

8. What is Hadoop streaming?

Hadoop distribution has a generic application programming interface for writing Map and Reduce jobs in any desired programming language like Python, Perl, Ruby, etc. This is referred to as Hadoop Streaming. Users can create and run jobs with any kind of shell scripts or executable as the Mapper or Reducers. The latest tool for Hadoop streaming is Spark.

9. What is the best hardware configuration to run Hadoop?

The best configuration for executing Hadoop jobs is dual core machines or dual processors with 4GB or 8GB RAM that use ECC memory. Hadoop highly benefits from using ECC memory though it is not low - end. ECC memory is recommended for running Hadoop because most of the Hadoop users have experienced various checksum errors by using non ECC memory. However, the hardware configuration also depends on the workflow requirements and can change accordingly.

10. What are the most commonly defined input formats in Hadoop?

The most common Input Formats defined in Hadoop are:

- Text Input Format- This is the default input format defined in Hadoop.
- Key Value Input Format- This input format is used for plain text files wherein the files are broken down into lines.
- Sequence File Input Format- This input format is used for reading files in sequence.

11. What are the steps involved in deploying a big data solution?

i) Data Ingestion – The foremost step in deploying big data solutions is to extract data from different sources which could be an Enterprise Resource

Planning System like SAP, any CRM like Salesforce or Siebel , RDBMS like MySQL or Oracle, or could be the log files, flat files, documents, images, social media feeds. This data needs to be stored in HDFS. Data can either be ingested through batch jobs that run every 15 minutes, once every night and so on or through streaming in real-time from 100 ms to 120 seconds.

ii) Data Storage – The subsequent step after ingesting data is to store it either in HDFS or NoSQL database like HBase. HBase storage works well for random read/write access whereas HDFS is optimized for sequential access.

iii) Data Processing – The ultimate step is to process the data using one of the processing frameworks like mapreduce, spark, pig, hive, etc.

12. How will you choose various file formats for storing and processing data using Apache Hadoop ?

The decision to choose a particular file format is based on the following factors-

- i) Schema evolution to add, alter and rename fields.
- ii) Usage pattern like accessing 5 columns out of 50 columns vs accessing most of the columns.
- iii) Splittability to be processed in parallel.
- iv) Read/Write/Transfer performance vs block compression saving storage space

File Formats that can be used with Hadoop - CSV, JSON, Columnar, Sequence files, AVRO, and Parquet file.

CSV Files

CSV files are an ideal fit for exchanging data between hadoop and external systems. It is advisable not to use header and footer lines when using CSV files.

JSON Files

Every JSON File has its own record. JSON stores both data and schema together in a record and also enables complete schema evolution and splitability. However, JSON files do not support block level compression.

Avro Files

This kind of file format is best suited for long term storage with Schema. Avro files store metadata with data and also let you specify independent schema for reading the files.

Parquet Files

A columnar file format that supports block level compression and is optimized for query performance as it allows selection of 10 or less columns from from 50+ columns records.

13. What makes Hadoop Fault tolerant?

Hadoop is said to be highly fault tolerant. Hadoop achieves this feat through the process of replication. Data is replicated across multiple nodes in a Hadoop cluster. The data is associated with a replication factor, which indicates the number of copies of the data that are present across the various nodes in a Hadoop cluster. For example, if the replication factor is 3, the data will be present in three different nodes of the Hadoop cluster, where each node will contain one copy each. In this manner, if there is a failure in any one of the nodes, the data will not be lost, but can be recovered from one of the other nodes which contains copies or replicas of the data.

Hadoop 3.0, however, makes use of the method of Erasure Coding (EC). EC is implemented by means of Redundant Array of Inexpensive Disks (RAID) by striping, where logically sequential data is divided into smaller units and these smaller units are then stored as consecutive units on different disks. Replication results in a storage overhead of 200% in case of a replication factor of 3 (which is the default replication factor). The use of EC in Hadoop improves storage efficiency when compared to replication, but still maintains the same level of fault tolerance.

Another feature of Hadoop 3.0 which ensures more fault tolerance is that Hadoop 3.0 supports two or more Standby nodes, unlike Hadoop that supports only two NameNodes.

14. How can you restart the NameNode in Hadoop?

The following commands can be used to restart the NameNode:

First, stop the NameNode using:

```
./sbin /Hadoop-daemon.sh stop NameNode
```

After this, start the NameNode using this command:

```
./sbin/Hadoop-daemon.sh start NameNode
```

15. What is meant by over/under- replicated blocks in Hadoop?

The NameNode is responsible for ensuring that the number of copies of the data across the cluster is equal to the replication factor.

- In some cases, maybe due to a failure in one of the nodes, the number of copies of the data is less than the replication factor. In such cases, the block is said to be under-replicated. The nodes are required to send updates to the NameNode regarding their health. In such cases,

if the NameNode does not receive any updates from a particular node, it will ensure that the replication factor for a block is reached by starting re-replication of the blocks from the available nodes onto a new node.

- Blocks are said to be over-replicated In cases where the number of copies of the data exceeds the replication factor. The name node fixed this issue by automatically deleting the extra copies of the blocks. Over-replication may occur in cases when after the shutdown of one particular node, the NameNode starts re-replication of data across new nodes, following which the node which was previously not available is restored.

Scenario Based Hadoop Interview Question -

16. What is Big Data?

Big data is defined as the voluminous amount of structured, unstructured or semi-structured data that has huge potential for mining but is so large that it cannot be processed using traditional database systems. Big data is characterized by its high velocity, volume and variety that requires cost effective and innovative methods for information processing to draw meaningful business insights. More than the volume of the data – it is the nature of the data that defines whether it is considered as Big Data or not.

17. Mention a business use case where you worked with the Hadoop Ecosystem

You can share details on how you deployed Hadoop distributions like Cloudera and Hortonworks in your organization either in a standalone environment or on the cloud. Mention how you configured the number of

required nodes , tools, services, security features such as SSL, SASL, Kerberos, etc. Having set up the Hadoop cluster, talk about how you initially extracted the data from data sources like APIs, SQL based databases, etc and stored it in HDFS(storage layer) , how you performed data cleaning and validation, and the series of ETLs you performed to extract the data in the given format to extract KPIs.

Some of the ETLs tasks include :

- Date format parsing
- The casting of data type values
- Deriving calculated fields

Hadoop HDFS Interview Questions and Answers

18. What is a block and block scanner in HDFS?

Block - The minimum amount of data that can be read or written is generally referred to as a “block” in HDFS. The de

size of a block in HDFS is 64MB.

Block Scanner - Block Scanner tracks the list of blocks present on a DataNode and verifies them to find any kind of checksum errors. Block Scanners use a throttling mechanism to reserve disk bandwidth on the datanode.

19. Explain the difference between NameNode, Backup Node and Checkpoint NameNode.

NameNode: NameNode is at the heart of the HDFS file system which manages the metadata i.e. the data of the files is not stored on the NameNode but rather it has the directory tree of all the files present in the HDFS file system on a hadoop cluster. NameNode uses two files for the namespace-

fsimage file- It keeps track of the latest checkpoint of the namespace.

edits file-It is a log of changes that have been made to the namespace since checkpoint.

Checkpoint Node-

Checkpoint Node keeps track of the latest checkpoint in a directory that has same structure as that of NameNode's directory. Checkpoint node creates checkpoints for the namespace at regular intervals by downloading the edits and fsimage file from the NameNode and merging it locally. The new image is then again updated back to the active NameNode.

BackupNode:

Backup Node also provides check pointing functionality like that of the checkpoint node but it also maintains its up-to-date in-memory copy of the file system namespace that is in sync with the active NameNode.

20. What is commodity hardware?

Commodity Hardware refers to inexpensive systems that do not have high availability or high quality. Commodity Hardware consists of RAM because there are specific services that need to be executed on RAM. Hadoop can be run on any commodity hardware and does not require any super computer s or high end hardware configuration to execute jobs.

21. What is the port number for NameNode, Task Tracker and Job Tracker?

NameNode 50070

Job Tracker 50030

22. Explain about the process of inter cluster data copying.

HDFS provides a distributed data copying facility through the DistCP from source to destination. If this data copying is within the hadoop cluster then it is referred to as inter cluster data copying. DistCP requires both source and destination to have a compatible or same version of hadoop.

23. How can you overwrite the replication factors in HDFS?

The replication factor in HDFS can be modified or overwritten in 2 ways-

1) Using the Hadoop FS Shell, replication factor can be changed per file basis using the below command-

```
$hadoop fs -setrep -w 2 /my/test_file (test_file is the filename whose replication factor will be set to 2)
```

2) Using the Hadoop FS Shell, replication factor of all files under a given directory can be modified using the below command-

```
3)$hadoop fs -setrep -w 5 /my/test_dir (test_dir is the name of the directory and all the files in this directory will have a replication factor set to 5)
```

24. Explain the difference between NAS and HDFS.

- NAS runs on a single machine and thus there is no probability of data redundancy whereas HDFS runs on a cluster of different machines thus there is data redundancy because of the replication protocol.
- NAS stores data on a dedicated hardware whereas in HDFS all the data blocks are distributed across local drives of the machines.

- In NAS data is stored independent of the computation and hence Hadoop MapReduce cannot be used for processing whereas HDFS works with Hadoop MapReduce as the computations in HDFS are moved to data.

25. Explain what happens if during the PUT operation, HDFS block is assigned a replication factor 1 instead of the default value 3.

Replication factor is a property of HDFS that can be set accordingly for the entire cluster to adjust the number of times the blocks are to be replicated to ensure high data availability. For every block that is stored in HDFS, the cluster will have $n-1$ duplicated blocks. So, if the replication factor during the PUT operation is set to 1 instead of the default value 3, then it will have a single copy of data. Under these circumstances when the replication factor is set to 1, if the DataNode crashes under any circumstances, then only single copy of the data would be lost.

26. What is the process to change the files at arbitrary locations in HDFS?

HDFS does not support modifications at arbitrary offsets in the file or multiple writers but files are written by a single writer in append only format i.e. writes to a file in HDFS are always made at the end of the file.

27. Explain about the indexing process in HDFS.

Indexing process in HDFS depends on the block size. HDFS stores the last part of the data that further points to the address where the next part of data chunk is stored.

28. What is a rack awareness and on what basis is data stored in a rack?

All the data nodes put together form a storage area i.e. the physical location of the data nodes is referred to as Rack in HDFS. The rack information i.e. the rack id of each data node is acquired by the NameNode. The process of selecting closer data nodes depending on the rack information is known as Rack Awareness.

The contents present in the file are divided into data block as soon as the client is ready to load the file into the hadoop cluster. After consulting with the NameNode, client allocates 3 data nodes for each data block. For each data block, there exists 2 copies in one rack and the third copy is present in another rack. This is generally referred to as the Replica Placement Policy.

29. What happens to a NameNode that has no data?

There does not exist any NameNode without data. If it is a NameNode then it should have some sort of data in it.

30. What happens when a user submits a Hadoop job when the NameNode is down- does the job get in to hold or does it fail.

The Hadoop job fails when the NameNode is down.

31. What happens when a user submits a Hadoop job when the Job Tracker is down- does the job get in to hold or does it fail.

The Hadoop job fails when the Job Tracker is down.

32. Whenever a client submits a hadoop job, who receives it?

NameNode receives the Hadoop job which then looks for the data requested by the client and provides the block information. JobTracker takes care of resource allocation of the hadoop job to ensure timely completion.

33. What do you understand by edge nodes in Hadoop?

Edges nodes are the interface between hadoop cluster and the external network. Edge nodes are used for running cluster administration tools and client applications. Edge nodes are also referred to as gateway nodes.

We have further categorized Hadoop HDFS Interview Questions for Freshers and Experienced-

- Hadoop Interview Questions and Answers for Freshers - Q.Nos- 2,3,7,9,10,11,13,14
- Hadoop Interview Questions and Answers for Experienced - Q.Nos- 1,2, 4,5,6,7,8,12,15

34. Mention some differences between HDFS high availability and HDFS federation.

HFS High Availability	HDFS Federation
There are two NameNodes, one active and one on standby.	In the HDFS Federation, the NameNodes are not related to each other.
If the primary NameNode goes down, the standby will take its place using the most recent metadata that it has.	Even if one NameNode goes down, there is no effect on the other NameNodes.
At a given time, only the active NameNode will be running, while the standby NameNode remains idle and only updates its metadata to stay up to date.	There is a pool of metadata which is shared by all the NameNodes. In addition, each NameNode will have its own metadata.
This requires two separate machines. One machine is configured to be the primary NameNode and the other as the Standby NameNode.	There is no limit on the number of machines that can be configured as the NameNodes.

Hadoop MapReduce Interview Questions and Answers

35. Explain the usage of Context Object.

Context Object is used to help the mapper interact with other Hadoop systems. Context Object can be used for updating counters, to report the progress and to provide any application level status updates. ContextObject has the configuration details for the job and also interfaces, that helps it to generating the output.

36. What are the core methods of a Reducer?

The 3 core methods of a reducer are –

1)setup () – This method of the reducer is used for configuring various parameters like the input data size, distributed cache, heap size, etc.

Function Definition- *public void setup (context)*

2)reduce () it is heart of the reducer which is called once per key with the associated reduce task.

Function Definition -*public void reduce (Key,Value,context)*

3)cleanup () - This method is called only once at the end of reduce task for clearing all the temporary files.

Function Definition -*public void cleanup (context)*

37. Explain about the partitioning, shuffle and sort phase

Shuffle Phase-Once the first map tasks are completed, the nodes continue to perform several other map tasks and also exchange the intermediate outputs

with the reducers as required. This process of moving the intermediate outputs of map tasks to the reducer is referred to as Shuffling.

Sort Phase- Hadoop MapReduce automatically sorts the set of intermediate keys on a single node before they are given as input to the reducer.

Partitioning Phase-The process that determines which intermediate keys and value will be received by each reducer instance is referred to as partitioning. The destination partition is same for any key irrespective of the mapper instance that generated it.

38. How to write a custom partitioner for a Hadoop MapReduce job?

Steps to write a Custom Partitioner for a Hadoop MapReduce Job-

- A new class must be created that extends the pre-defined Partitioner Class.
- getPartition method of the Partitioner class must be overridden.
- The custom partitioner to the job can be added as a config file in the wrapper which runs Hadoop MapReduce or the custom partitioner can be added to the job by using the set method of the partitioner class.

39. What are side data distribution techniques in Hadoop?

The extra read only data required by a hadoop job to process the main dataset is referred to as side data. Hadoop has two side data distribution techniques

-

i) Using the job configuration - This technique should not be used for transferring more than few kilobytes of data as it can pressurize the memory usage of hadoop daemons, particularly if your system is running several hadoop jobs.

ii) Distributed Cache - Rather than serializing side data using the job configuration, it is suggested to distribute data using hadoop's distributed cache mechanism.

40. What does the jps command in hadoop do?

A. JPS is short for Java Virtual Machine Process Status Tool. it gives the status of the daemons running the Hadoop cluster. It is a tool which is used to check if Hadoop processes are running or not. JPS gives an output which mentions the status of namenode, datanode, secondary namenode, Jobtracker and Task tracker.

Hadoop HBase Interview Questions and Answers

41. When should you use HBase and what are the key components of HBase?

HBase should be used when the big data application has –

- 1)A variable schema
- 2)When data is stored in the form of collections
- 3)If the application demands key based access to data while retrieving.

Key components of HBase are –

Region- This component contains memory data store and Hfile.

Region Server-This monitors the Region.

HBase Master-It is responsible for monitoring the region server.

Zookeeper- It takes care of the coordination between the HBase Master component and the client.

Catalog Tables-The two important catalog tables are ROOT and META.ROOT table tracks where the META table is and META table stores all the regions in the system.

42. What are the different operational commands in HBase at record level and table level?

Record Level Operational Commands in HBase are -put, get, increment, scan and delete.

Table Level Operational Commands in HBase are-describe, list, drop, disable and scan.

43. What is Row Key?

Every row in an HBase table has a unique identifier known as RowKey. It is used for grouping cells logically and it ensures that all cells that have the same RowKeys are co-located on the same server. RowKey is internally regarded as a byte array.

44. Explain the difference between RDBMS data model and HBase data model.

RDBMS is a schema based database whereas HBase is schema less data model.

RDBMS does not have support for in-built partitioning whereas in HBase there is automated partitioning.

RDBMS stores normalized data whereas HBase stores de-normalized data.

45. Explain about the different catalog tables in HBase?

The two important catalog tables in HBase, are ROOT and META. ROOT table tracks where the META table is and META table stores all the regions in the system.

46. What is column families? What happens if you alter the block size of ColumnFamily on an already populated database?

The logical deviation of data is represented through a key known as column Family. Column families consist of the basic unit of physical storage on which compression features can be applied. In an already populated database, when the block size of column family is altered, the old data will remain within the old block size whereas the new data that comes in will take the new block size. When compaction takes place, the old data will take the new block size so that the existing data is read correctly.

47. Explain the difference between HBase and Hive.

HBase and Hive both are completely different hadoop based technologies-Hive is a data warehouse infrastructure on top of Hadoop whereas HBase is a NoSQL key value store that runs on top of Hadoop. Hive helps SQL savvy people to run MapReduce jobs whereas HBase supports 4 primary operations-put, get, scan and delete. HBase is ideal for real time querying of big data where Hive is an ideal choice for analytical querying of data collected over period of time.

48. Explain the process of row deletion in HBase.

On issuing a delete command in HBase through the HBase client, data is not actually deleted from the cells but rather the cells are made invisible by setting a tombstone marker. The deleted cells are removed at regular intervals during compaction.

49. What are the different types of tombstone markers in HBase for deletion?

There are 3 different types of tombstone markers in HBase for deletion-

- 1)Family Delete Marker- This markers marks all columns for a column family.
- 2)Version Delete Marker-This marker marks a single version of a column.
- 3)Column Delete Marker-This markers marks all the versions of a column.

50. Explain about HLog and WAL in HBase.

All edits in the HStore are stored in the HLog. Every region server has one HLog. HLog contains entries for edits of all regions performed by a particular Region Server.WAL abbreviates to Write Ahead Log (WAL) in which all the HLog edits are written immediately.WAL edits remain in the memory till the flush period in case of deferred log flush.

51. What are the difficulties associated with hotspotting in HBase?

- A. Hotspotting arises when a large amount of client traffic is directed at a single node or a certain few nodes of a cluster. This traffic may be in the form of reads, writes or other database operations. Large amounts of traffic directed at just a few nodes overwhelms the machines

responsible for hosting their corresponding regions and results in performance degradation. It could also potentially lead to unavailability of some of the regions as well.

Hadoop Sqoop Interview Questions and Answers

52. Explain about some important Sqoop commands other than import and export.

Create Job (--create)

Here we are creating a job with the name my job, which can import the table data from RDBMS table to HDFS. The following command is used to create a job that is importing data from the employee table in the db database to the HDFS file.

```
$ Sqoop job --create myjob \  
  
--import \  
  
--connect jdbc:mysql://localhost/db \  
  
--username root \  
  
--table employee --m 1
```

Verify Job (--list)

‘--list’ argument is used to verify the saved jobs. The following command is used to verify the list of saved Sqoop jobs.

```
$ Sqoop job --list
```

Inspect Job (--show)

‘--show’ argument is used to inspect or verify particular jobs and their details. The following command and sample output is used to verify a job called myjob.

```
$ Sqoop job --show myjob
```

Execute Job (--exec)

'--exec' option is used to execute a saved job. The following command is used to execute a saved job called myjob.

```
$ Sqoop job --exec myjob
```

53. How Sqoop can be used in a Java program?

The Sqoop jar in classpath should be included in the java code. After this the method Sqoop.runTool () method must be invoked. The necessary parameters should be created to Sqoop programmatically just like for command line.

54. What is the process to perform an incremental data load in Sqoop?

The process to perform incremental data load in Sqoop is to synchronize the modified or updated data (often referred as delta data) from RDBMS to Hadoop. The delta data can be facilitated through the incremental load command in Sqoop.

Incremental load can be performed by using Sqoop import command or by loading the data into hive without overwriting it. The different attributes that need to be specified during incremental load in Sqoop are-

1)Mode (incremental) –The mode defines how Sqoop will determine what the new rows are. The mode can have value as Append or Last Modified.

2)Col (Check-column) –This attribute specifies the column that should be examined to find out the rows to be imported.

3)Value (last-value) –This denotes the maximum value of the check column from the previous import operation.

55. Is it possible to do an incremental import using Sqoop?

Yes, Sqoop supports two types of incremental imports-

1)Append

2)Last Modified

To insert only rows Append should be used in import command and for inserting the rows and also updating Last-Modified should be used in the import command.

56. What is the standard location or path for Hadoop Sqoop scripts?

/usr/bin/Hadoop Sqoop

57. How can you check all the tables present in a single database using Sqoop?

The command to check the list of all tables present in a single database using Sqoop is as follows-

Sqoop list-tables --connect jdbc: mysql: //localhost/user;

58. How are large objects handled in Sqoop?

Sqoop provides the capability to store large sized data into a single field based on the type of data. Sqoop supports the ability to store-

1)CLOB 's – Character Large Objects

2)BLOB's –Binary Large Objects

Large objects in Sqoop are handled by importing the large objects into a file referred as “LobFile” i.e. Large Object File. The LobFile has the ability to store records of huge size, thus each record in the LobFile is a large object.

59. Can free form SQL queries be used with Sqoop import command? If yes, then how can they be used?

Sqoop allows us to use free form SQL queries with the import command. The import command should be used with the `-e` and `-query` options to execute free form SQL queries. When using the `-e` and `-query` options with the import command the `-target dir` value must be specified.

60. Differentiate between Sqoop and distCP.

DistCP utility can be used to transfer data between clusters whereas Sqoop can be used to transfer data only between Hadoop and RDBMS.

61. What are the limitations of importing RDBMS tables into Hcatalog directly?

There is an option to import RDBMS tables into Hcatalog directly by making use of `-hcatalog -database` option with the `-hcatalog -table` but the limitation to it is that there are several arguments like `-as-avrofile` , `-direct`, `-as-sequencefile`, `-target-dir` , `-export-dir` are not supported.

62. Is it suggested to place the data transfer utility sqoop on an edge node ?

It is not suggested to place sqoop on an edge node or gateway node because the high data transfer volumes could risk the ability of hadoop services on the same node to communicate. Messages are the lifeblood of any hadoop service and high latency could result in the whole node being cut off from the hadoop cluster.

Hadoop Flume Interview Questions and Answers

63. Explain about the core components of Flume.

The core components of Flume are –

Event- The single log entry or unit of data that is transported.

Source- This is the component through which data enters Flume workflows.

Sink-It is responsible for transporting data to the desired destination.

Channel- it is the duct between the Sink and Source.

Agent- Any JVM that runs Flume.

Client- The component that transmits event to the source that operates with the agent.

64. Does Flume provide 100% reliability to the data flow?

Yes, Apache Flume provides end to end reliability because of its transactional approach in data flow.

65. How can Flume be used with HBase?

Apache Flume can be used with HBase using one of the two HBase sinks –

- HBaseSink (org.apache.flume.sink.hbase.HBaseSink) supports secure HBase clusters and also the novel HBase IPC that was introduced in the version HBase 0.96.
- AsyncHBaseSink (org.apache.flume.sink.hbase.AsyncHBaseSink) has better performance than HBase sink as it can easily make non-blocking calls to HBase.

Working of the HBaseSink –

In HBaseSink, a Flume Event is converted into HBase Increments or Puts. Serializer implements the HBaseEventSerializer which is then instantiated when the sink starts. For every event, sink calls the initialize method in the serializer which then translates the Flume Event into HBase increments and puts to be sent to HBase cluster.

Working of the AsyncHBaseSink-

AsyncHBaseSink implements the AsyncHBaseEventSerializer. The initialize method is called only once by the sink when it starts. Sink invokes the setEvent method and then makes calls to the getIncrements and getActions methods just similar to HBase sink. When the sink stops, the cleanUp method is called by the serializer.

66. Explain about the different channel types in Flume. Which channel type is faster?

The 3 different built in channel types available in Flume are-

MEMORY Channel – Events are read from the source into memory and passed to the sink.

JDBC Channel – JDBC Channel stores the events in an embedded Derby database.

FILE Channel –File Channel writes the contents to a file on the file system after reading the event from a source. The file is deleted only after the contents are successfully delivered to the sink.

MEMORY Channel is the fastest channel among the three however has the risk of data loss. The channel that you choose completely depends on the nature of the big data application and the value of each event.

67. Which is the reliable channel in Flume to ensure that there is no data loss?

FILE Channel is the most reliable channel among the 3 channels JDBC, FILE and MEMORY.

68. Explain about the replication and multiplexing selectors in Flume.

Channel Selectors are used to handle multiple channels. Based on the Flume header value, an event can be written just to a single channel or to multiple channels. If a channel selector is not specified to the source then by default it is the Replicating selector. Using the replicating selector, the same event is written to all the channels in the source's channels list. Multiplexing channel selector is used when the application has to send different events to different channels.

69. How multi-hop agent can be setup in Flume?

Avro RPC Bridge mechanism is used to setup Multi-hop agent in Apache Flume.

70. Does Apache Flume provide support for third party plug-ins?

Most of the data analysts use Apache Flume has plug-in based architecture as it can load data from external sources and transfer it to external destinations.

71. Is it possible to leverage real time analysis on the big data collected by Flume directly? If yes, then explain how.

Data from Flume can be extracted, transformed and loaded in real-time into Apache Solr servers using *MorphlineSolrSink*

72. Differentiate between FileSink and FileRollSink

The major difference between HDFS FileSink and FileRollSink is that HDFS File Sink writes the events into the Hadoop Distributed File System (HDFS) whereas File Roll Sink stores the events into the local file system.

Hadoop Zookeeper Interview Questions and Answers

73. Can Apache Kafka be used without Zookeeper?

It is not possible to use Apache Kafka without Zookeeper because if the Zookeeper is down Kafka cannot serve client request.

74. Name a few companies that use Zookeeper.

Yahoo, Solr, Helprace, Neo4j, Rackspace

75. What is the role of Zookeeper in HBase architecture?

In HBase architecture, ZooKeeper is the monitoring server that provides different services like –tracking server failure and network partitions, maintaining the configuration information, establishing communication between the clients and region servers, usability of ephemeral nodes to identify the available servers in the cluster.

76. Explain about ZooKeeper in Kafka

Apache Kafka uses ZooKeeper to be a highly distributed and scalable system. Zookeeper is used by Kafka to store various configurations and use them across the hadoop cluster in a distributed manner. To achieve distributed-ness, configurations are distributed and replicated throughout the leader and follower nodes in the ZooKeeper ensemble. We cannot directly connect to Kafka by bye-passing ZooKeeper because if the ZooKeeper is down it will not be able to serve the client request.

77. Explain how Zookeeper works

ZooKeeper is referred to as the King of Coordination and distributed applications use ZooKeeper to store and facilitate important configuration information updates. ZooKeeper works by coordinating the processes of distributed applications. ZooKeeper is a robust replicated synchronization service with eventual consistency. A set of nodes is known as an ensemble and persisted data is distributed between multiple nodes.

3 or more independent servers collectively form a ZooKeeper cluster and elect a master. One client connects to any of the specific server and migrates if a particular node fails. The ensemble of ZooKeeper nodes is alive till the majority of nodes are working. The master node in ZooKeeper is dynamically selected by the consensus within the ensemble so if the master node fails then the role of master node will migrate to another node which is selected dynamically. Writes are linear and reads are concurrent in ZooKeeper.

78. List some examples of Zookeeper use cases.

- Found by Elastic uses Zookeeper comprehensively for resource allocation, leader election, high priority notifications and discovery. The entire service of Found built up of various systems that read and write to Zookeeper.
- Apache Kafka that depends on ZooKeeper is used by LinkedIn
- Storm that relies on ZooKeeper is used by popular companies like Groupon and Twitter.

79. How to use Apache Zookeeper command line interface?

ZooKeeper has a command line client support for interactive use. The command line interface of ZooKeeper is similar to the file and shell system of UNIX. Data in ZooKeeper is stored in a hierarchy of Znodes where each znode can contain data just similar to a file. Each znode can also have children just like directories in the UNIX file system.

Zookeeper-client command is used to launch the command line client. If the initial prompt is hidden by the log messages after entering the command, users can just hit ENTER to view the prompt.

80. What are the different types of Znodes?

There are 2 types of Znodes namely- Ephemeral and Sequential Znodes.

- The Znodes that get destroyed as soon as the client that created it disconnects are referred to as Ephemeral Znodes.

- Sequential Znode is the one in which sequential number is chosen by the ZooKeeper ensemble and is pre-fixed when the client assigns name to the znode.

81. What are watches?

Client disconnection might be troublesome problem especially when we need to keep a track on the state of Znodes at regular intervals. ZooKeeper has an event system referred to as watch which can be set on Znode to trigger an event whenever it is removed, altered or any new children are created below it.

82. What problems can be addressed by using Zookeeper?

In the development of distributed systems, creating own protocols for coordinating the hadoop cluster results in failure and frustration for the developers. The architecture of a distributed system can be prone to deadlocks, inconsistency and race conditions. This leads to various difficulties in making the hadoop cluster fast, reliable and scalable. To address all such problems, Apache ZooKeeper can be used as a coordination service to write correct distributed applications without having to reinvent the wheel from the beginning.

Hadoop Pig Interview Questions and Answers

83. What are different modes of execution in Apache Pig?

Apache Pig runs in 2 modes- one is the “Pig (Local Mode) Command Mode” and the other is the “Hadoop MapReduce (Java) Command Mode”. Local Mode requires access to only a single machine where all files are installed and executed on a local host whereas MapReduce requires accessing the Hadoop cluster.

84. Explain about co-group in Pig.

COGROUP operator in Pig is used to work with multiple tuples. COGROUP operator is applied on statements that contain or involve two or more relations. The COGROUP operator can be applied on up to 127 relations at a time. When using the COGROUP operator on two tables at once-Pig first groups both the tables and after that joins the two tables on the grouped columns.

Hadoop Hive Interview Questions and Answers

85. Explain about the SMB Join in Hive.

In SMB join in Hive, each mapper reads a bucket from the first table and the corresponding bucket from the second table and then a merge sort join is performed. Sort Merge Bucket (SMB) join in hive is mainly used as there is no limit on file or partition or table join. SMB join can best be used when the tables are large. In SMB join the columns are bucketed and sorted using the join columns. All tables should have the same number of buckets in SMB join.

86. How can you connect an application, if you run Hive as a server?

When running Hive as a server, the application can be connected in one of the 3 ways-

ODBC Driver-This supports the ODBC protocol

JDBC Driver- This supports the JDBC protocol

Thrift Client- This client can be used to make calls to all hive commands using different programming language like PHP, Python, Java, C++ and Ruby.

87. What does the overwrite keyword denote in Hive load statement?

Overwrite keyword in Hive load statement deletes the contents of the target table and replaces them with the files referred by the file path i.e. the files that are referred by the file path will be added to the table when using the overwrite keyword.

88. What is SerDe in Hive? How can you write your own custom SerDe?

SerDe is a Serializer DeSerializer. Hive uses SerDe to read and write data from tables. Generally, users prefer to write a Deserializer instead of a SerDe as they want to read their own data format rather than writing to it. If the SerDe supports DDL i.e. basically SerDe with parameterized columns and different column types, the users can implement a Protocol based DynamicSerDe rather than writing the SerDe from scratch.

89. What are the differences between internal and external tables in Hive?

Internal tables

External tables

Data tables created inside the database are considered internal tables of Hive by default.	External tables are found on shareable data locations in the HDFS outside the database
--	--

Hive has ownership of the internal tables and manages the metadata and the data within internal tables by itself.	Hive does not have ownership of external tables and only manages the metadata associated with the tables.
---	---

Both metadata and the table data will be removed when the table is dropped.	Only metadata is removed on dropping the table.
---	---

Internal tables support the TRUNCATE command.	External tables do not support the TRUNCATE command.
---	--

Internal tables support query result caching (the results of a particular query are temporarily stored if needed for use in a subsequent query).	External tables do not support query result caching.
--	--

90. Consider the following two queries. What is the difference in the query results?

- A. `SELECT x.value, y.value FROM x LEFT OUTER JOIN y ON (x.key=y.key)`
- B. `SELECT x.value, y.value FROM x RIGHT OUTER JOIN y ON (x.key=y.key)`
- C. In both Queries 1 and 2, the output rows will contain a list of x.value and y.value for all cases where the values of x.key and y.key are equal.

The difference here is that, in query 1, if there is no corresponding value 'value' in y for x.value it will be added to the output query as 'x.value,NULL', since x

is to the left of y (x LEFT OUTER JOIN y). All values for y.value which do not have a corresponding x.value will be dropped from the query result of query 1.

In the case of query 2, it is vice versa, since it is a right outer join and on the right, there is y. This means that in the case of Query 2, all y values which do not have a corresponding x value will be present in the result query as 'NULL, y.value' but all x values without a corresponding y value will be dropped.

91. How can one list down all partitions of a table in Hive?

There are two ways to list all the partitions of a table in Hive.

1. Using Hive prompt

```
$hive -e show partitions table_name;
```

2. Using Hive command

```
$ hive> show partitions table_name;
```

Hadoop YARN Interview Questions and Answers

92. What are the stable versions of Hadoop?

Release 2.7.1 (stable)

Release 2.4.1

Release 1.2.1 (stable)

93. What is Apache Hadoop YARN?

YARN is a powerful and efficient feature rolled out as a part of Hadoop 2.0. YARN is a large scale distributed system for running big data applications.

94. Is YARN a replacement of Hadoop MapReduce?

YARN is not a replacement of Hadoop but it is a more powerful and efficient technology that supports MapReduce and is also referred to as Hadoop 2.0 or MapReduce 2.

95. What are the additional benefits YARN brings in to Hadoop?

- Effective utilization of the resources as multiple applications can be run in YARN all sharing a common resource. In Hadoop MapReduce there are separate slots for Map and Reduce tasks whereas in YARN there is no fixed slot. The same container can be used for Map and Reduce tasks leading to better utilization.
- YARN is backward compatible so all the existing MapReduce jobs.
- Using YARN, one can even run applications that are not based on the MapReduce model

96. How can native libraries be included in YARN jobs?

There are two ways to include native libraries in YARN jobs-

- 1) By setting the `-Djava.library.path` on the command line but in this case there are chances that the native libraries might not be loaded correctly and there is possibility of errors.
- 2) The better option to include native libraries is to set the `LD_LIBRARY_PATH` in the `.bashrc` file.

97. Explain the differences between Hadoop 1.x and Hadoop 2.x

- In Hadoop 1.x, MapReduce is responsible for both processing and cluster management whereas in Hadoop 2.x processing is taken care of by other processing models and YARN is responsible for cluster management.
- Hadoop 2.x scales better when compared to Hadoop 1.x with close to 10000 nodes per cluster.
- Hadoop 1.x has single point of failure problem and whenever the NameNode fails it has to be recovered manually. However, in case of Hadoop 2.x StandBy NameNode overcomes the SPOF problem and whenever the NameNode fails it is configured for automatic recovery.
- Hadoop 1.x works on the concept of slots whereas Hadoop 2.x works on the concept of containers and can also run generic tasks.

98. What are the core changes in Hadoop 2.0?

Hadoop 2.x provides an upgrade to Hadoop 1.x in terms of resource management, scheduling and the manner in which execution occurs. In Hadoop 2.x the cluster resource management capabilities work in isolation from the MapReduce specific programming logic. This helps Hadoop to share resources dynamically between multiple parallel processing frameworks like Impala and the core MapReduce component. Hadoop 2.x Hadoop 2.x allows workable and fine grained resource configuration leading to efficient and better cluster utilization so that the application can scale to process larger number of jobs.

99. Differentiate between NFS, Hadoop NameNode and JournalNode.

HDFS is a write once file system so a user cannot update the files once they exist either they can read or write to it. However, under certain scenarios in the enterprise environment like file uploading, file downloading, file browsing

or data streaming –it is not possible to achieve all this using the standard HDFS. This is where a distributed file system protocol Network File System (NFS) is used. NFS allows access to files on remote machines just similar to how local file system is accessed by applications.

Namenode is the heart of the HDFS file system that maintains the metadata and tracks where the file data is kept across the Hadoop cluster.

StandBy Nodes and Active Nodes communicate with a group of light weight nodes to keep their state synchronized. These are known as Journal Nodes.

100. What are the modules that constitute the Apache Hadoop 2.0 framework?

Hadoop 2.0 contains four important modules of which 3 are inherited from Hadoop 1.0 and a new module YARN is added to it.

1. Hadoop Common – This module consists of all the basic utilities and libraries that required by other modules.
2. HDFS- Hadoop Distributed file system that stores huge volumes of data on commodity machines across the cluster.
3. MapReduce- Java based programming model for data processing.
4. YARN- This is a new module introduced in Hadoop 2.0 for cluster resource management and job scheduling.

101. How is the distance between two nodes defined in Hadoop?

Measuring bandwidth is difficult in Hadoop so network is denoted as a tree in Hadoop. The distance between two nodes in the tree plays a vital role in forming a Hadoop cluster and is defined by the network topology and java interface DNStoSwitchMapping. The distance is equal to the sum of the distance to the closest common ancestor of both the nodes. The method `getDistance(Node node1, Node node2)` is used to calculate the distance

between two nodes with the assumption that the distance from a node to its parent node is always 1.

We have further categorized Hadoop YARN Interview Questions for Freshers and Experienced-

- Hadoop Interview Questions and Answers for Freshers - Q.Nos- 2,3,4,6,7,9
- Hadoop Interview Questions and Answers for Experienced - Q.Nos- 1,5,8,10

102. What are the key components of YARN?

Hadoop YARN consists of the following components:

Resource Manager: The Resource manager handles all the allocation of resources in the cluster and management of applications that have to run. The Resource manager is the ultimate authority for resource allocation. When it receives a request for the processing of an application, it forwards this request to the node manager concerned.

Node Manager: The Node manager is responsible for handling an individual node in a Hadoop cluster. It manages tasks and applications on the node that it is responsible for handling. The Node manager is registered with the resource manager. It informs the resource manager about the health status of the node by sending heartbeat messages. The primary goal of the node manager is to manage application containers that are assigned to it by the resource manager. It also keeps track of the resource usage of the containers in the node.

Application Master: In Hadoop, an application refers to a single job that is submitted to the framework. Each application has a unique Application Master associated with it, that is specific to that particular framework. The Application Master coordinates the execution of an application in the cluster

and manages any faults which may occur. It negotiates resources and containers based on input from the resource manager. Periodic heartbeats are sent to the Resource Manager to keep it updated with respect to the health and resource demands associated with the resources. The Application Master along with the Node Manager monitor and execute the tasks carried out by various components.

Container: The container is a collection of physical resources including RAM, CPU cores and disks present in a single node of a cluster. Containers in YARN are managed by the CLC (container life-cycle). The CLC contains details about the environment variables, various dependencies stored in remotely accessible storage, security tokens and the payload associated with Node Manager services. The container grants an application the rights to use a specific amount of resources from a particular host.

103. What do you know about the resource manager available in YARN?

The Resource Manager is the ultimate authority and the master daemon of YARN. it is responsible for allocation of resources and management of all the applications. Whenever the resource manager receives a processing request, it forwards this request to the corresponding node manager and allocates resources for the execution of the request. There are two major components:

- Scheduler: it is responsible for performing scheduling of tasks based on the applications allotted and availability of resources. The Scheduler does not perform any other tasks beyond scheduling, such as monitoring or tracking of resources. It does not guarantee restarting of tasks in cases of failure due to either hardware or software failures.
- Application manager: the application manager is responsible for accepting applications and negotiating for a container to execute the ApplicationMaster which is specific to applications. It also provides the service for restarting the ApplicationMaster in case of any failures.

The Application manager maintains a collection of applications that have been submitted. It also maintains a cache of applications that have been completed.

104. What are the scheduling policies available in Apache YARN?

There are three types of scheduling policies that are provided by YARN. These are:

1. FIFO scheduler: In the First in first out (FIFO) scheduler, the tasks or applications that come first will be executed first. The tasks are placed into the queue when submitted and are serviced in the order in which they enter the queue. Once a job begins getting executed, no intervention is allowed from other jobs. This is the default scheduling policy that is used in Hadoop, and hence, does not require any special configuration. The downside to this scheduler is that even if a particular task is of high priority, it has no choice but to wait its turn to be executed.
2. Capacity Scheduler: in the case of a Capacity Scheduler, there are multiple queues for scheduling tasks. Corresponding to each job queue, there are some slots or cluster resources which will be used to perform job execution. This means that each job queue has its own resources to execute the tasks in that queue. If a situation arises where only one job queue has tasks to perform, then the tasks in that particular job queue can access the slots in other queues. The Cluster Scheduler also provides a layer of abstraction, which allows the user to see if a particular job is utilizing more cluster resources or slots. In this manner, it is possible to keep a check on jobs and ensure that they are not taking up excess time or excessive resources from the cluster. The Capacity Scheduler is an excellent way to maximise the throughput of a Hadoop Cluster.
3. Fair Scheduler: The Fair Scheduler works a little similar to the Capacity Scheduler, however, the priority of jobs is taken into

account. By means of the Fair Scheduler, YARN applications can distribute the resources in large Hadoop clusters in such a way that all applications within the cluster are handled in an equal amount of time. The Fair Scheduler takes decisions based on memory availability and maintains resources dynamically. Whenever a high priority task arrives in a queue, the task is executed in parallel by replacing some portion of the slots which are already dedicated. The Fair Scheduler is a good option when it comes to the management of high priority tasks.

Hadoop Testing Interview Questions

105. How will you test data quality?

The entire data that has been collected could be important but all data is not equal so it is necessary to first define from where the data came, how the data would be used and consumed. Data that will be consumed by vendors or customers within the business ecosystem should be checked for quality and needs to be cleaned. This can be done by applying stringent data quality rules and by inspecting different properties like conformity, perfection, repetition, reliability, validity, completeness of data, etc. The collected data might have issues like NULL values, outliers, data type issues, encoding issues, language issues, column shift issues, special characters, header issues etc. So it is important to have a data cleaning and validation framework in place to clean and validate the data issues to ensure data completeness. We can use python to read the incoming data into pandas data frames and perform various checks on the data and transform the data into the required format, clean the data, validate the data, and store it into the data lake or HDFS for further processing

106. What are the challenges that you encounter when testing large datasets?

- More data needs to be substantiated.
- Testing large datasets requires automation.
- Testing options across all platforms need to be defined.

107. What do you understand by MapReduce validation ?

This the subsequent and most important step of the big data testing process. Hadoop developer needs to verify the right implementation of the business logic on every hadoop cluster node and validate the data after executing it on all the nodes to determine -

- Appropriate functioning of the MapReduce function.
- Validate if rules for data segregation are implemented.
- Pairing and creation of key-value pairs.

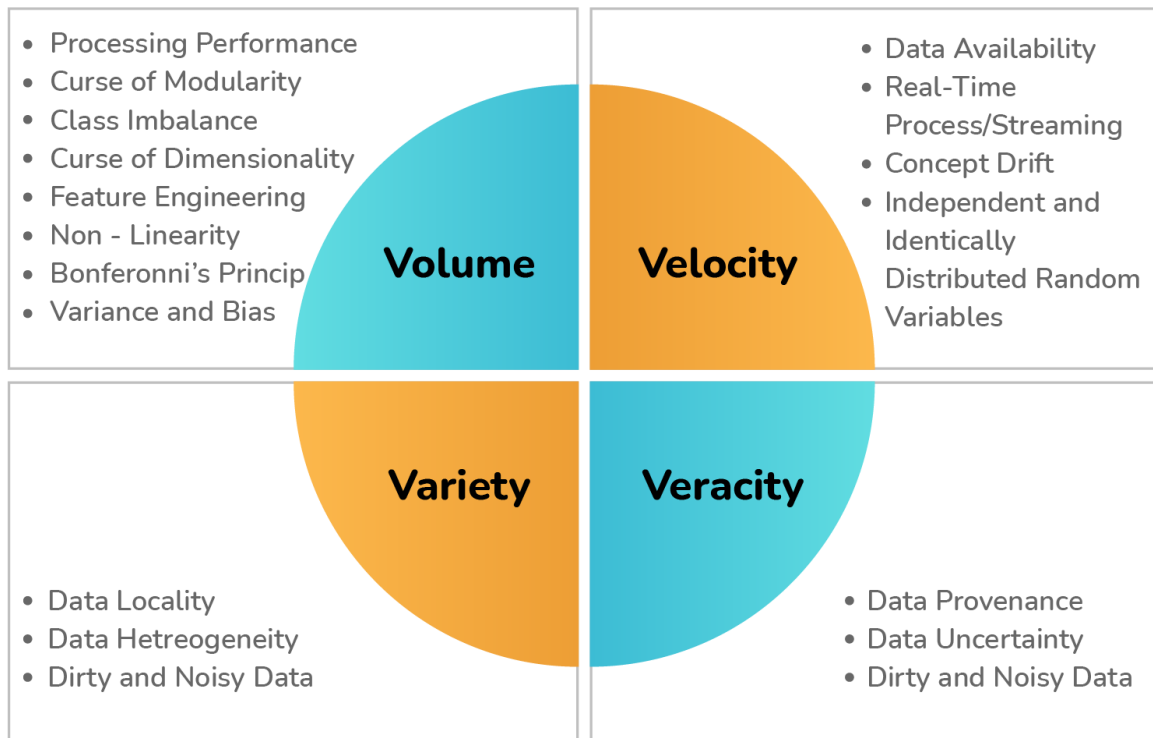
Hadoop Interview Questions

108. 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.



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.

109. 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

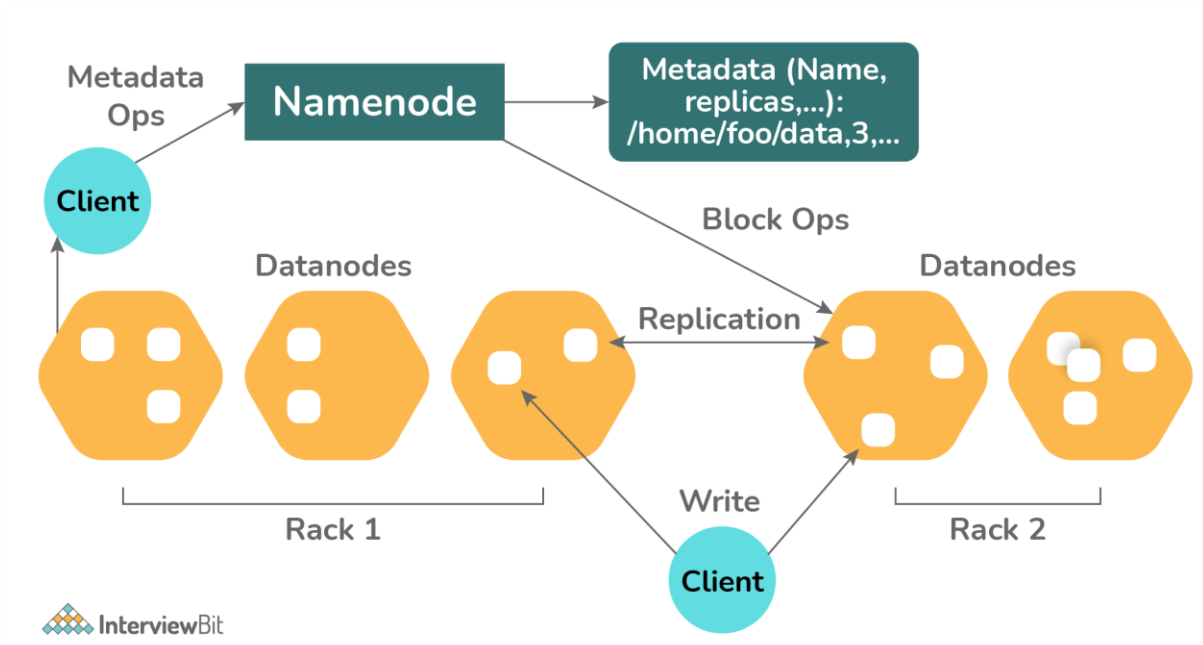
There are three components of Hadoop are:

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.



110. 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.

111. 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.

112. 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 Node and One Namespace per Cluster.
- It does not help the Horizontal Scalability of NameNode.
- It runs only Map/Reduce jobs.

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

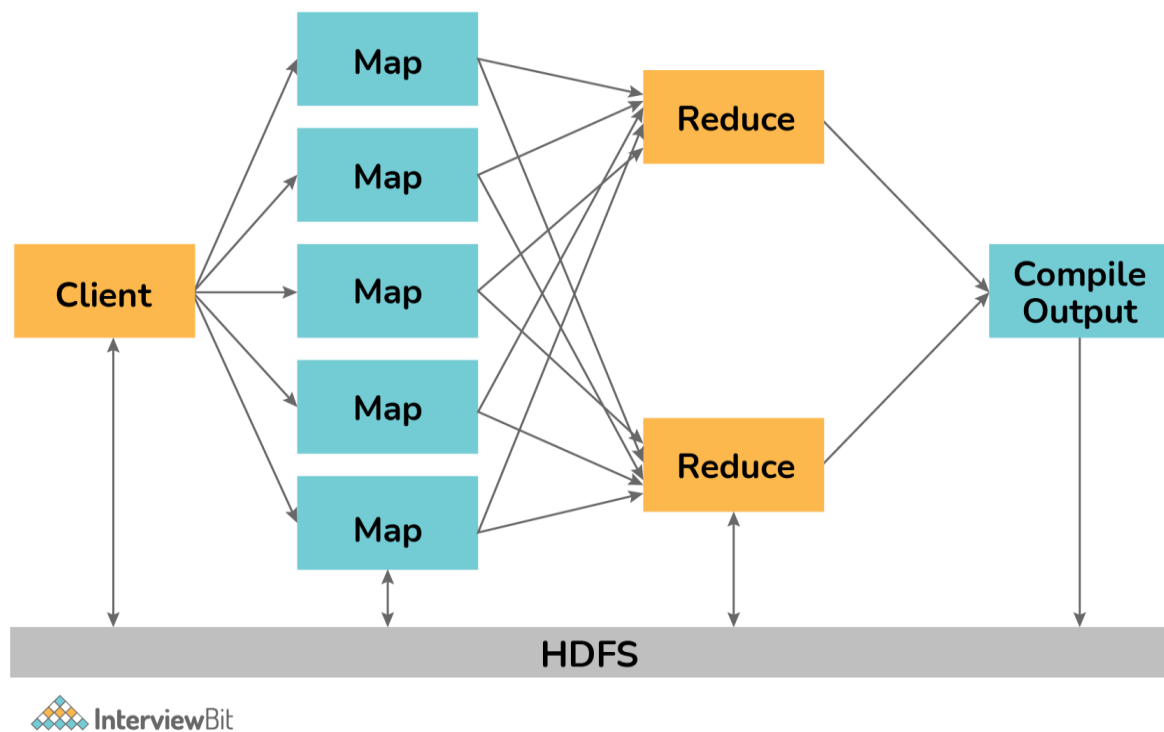
HDFS	NAS
HDFS is a Distributed File system that is mainly used to store data by commodity hardware.	NAS is a file-level computer data storage server connected to a computer network that provides network access to a heterogeneous group of clients.
HDFS is programmed to work with the MapReduce paradigm.	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.

114. 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.

115. 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.



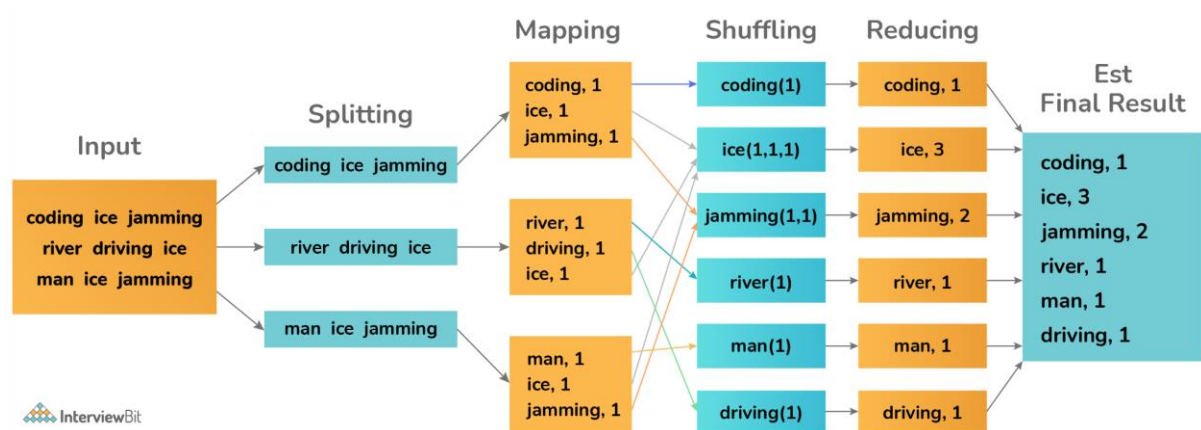
The first is the map operation, which takes a set of data and transforms it into a different collection of data, where individual elements are divided into tuples. The reduce operation consolidates those data tuples based on the key and subsequently modifies the value of the key.

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.

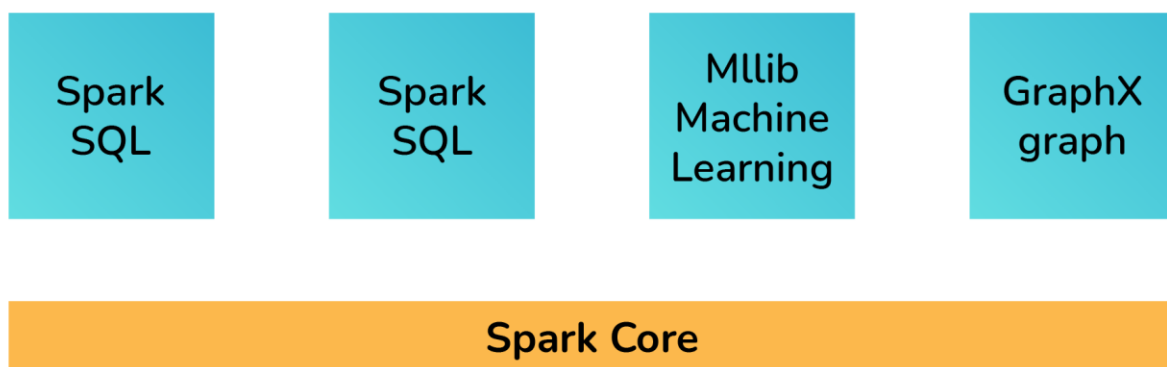
- Lastly, all the output key/value pairs are then assembled and written in the output file.

116. 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.

117. 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.

118. 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.

119. 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.

120. What is Apache Pig?

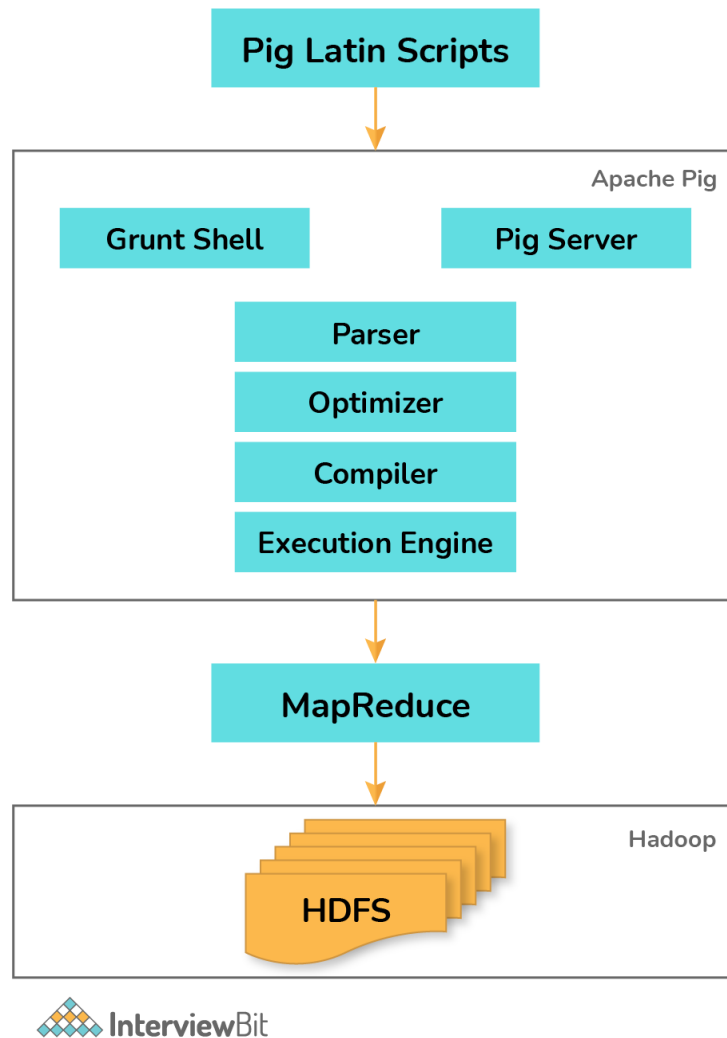
MapReduce needs programs to be translated into map and reduce stages. As not all data analysts are accustomed to MapReduce, Yahoo researchers

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.

121. 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.

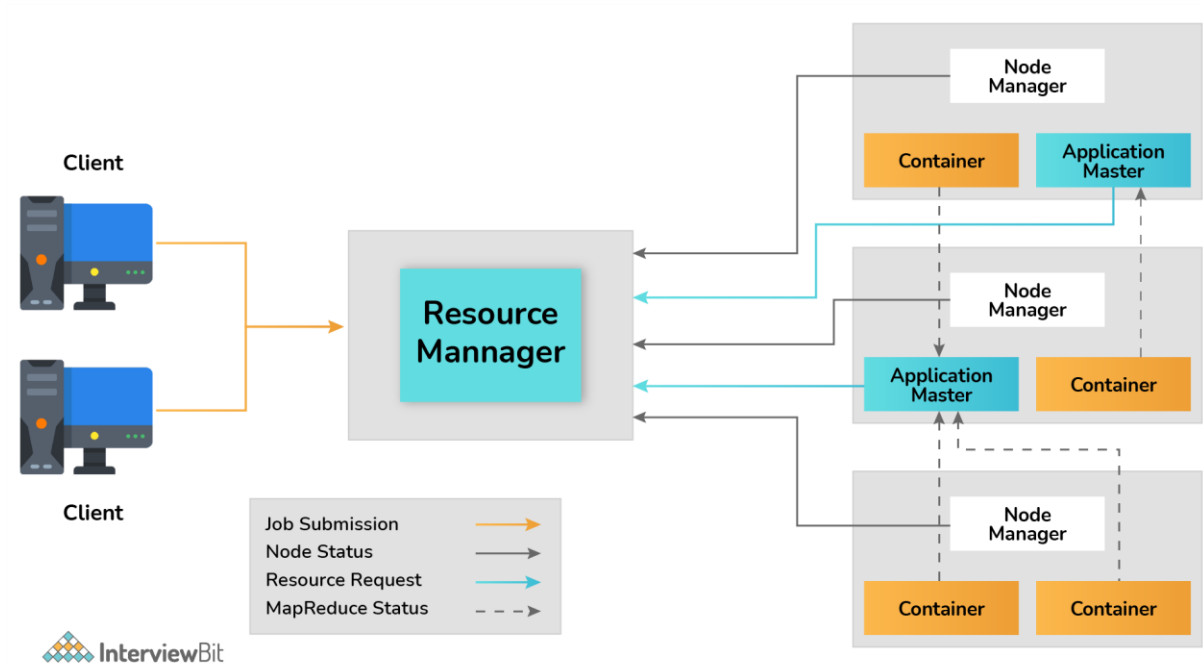
122. What is Yarn?

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.

123. 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.



124. 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.

125. 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.

126. 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 eznodel. Once client1 logs out of the ZooKeeper server, the eznodel 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 sznodel. In the ZooKeeper server, the sznodel 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.

127. 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 FS -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:~$
```

128. 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

129. 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.

130. 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.

131. 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.

132. 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.

133. 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.

134. 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`

135. 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]`

136. 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 leave get 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 details force	Requests the standard or detailed current status of the distributed upgrade, or forces the upgrade to proceed.



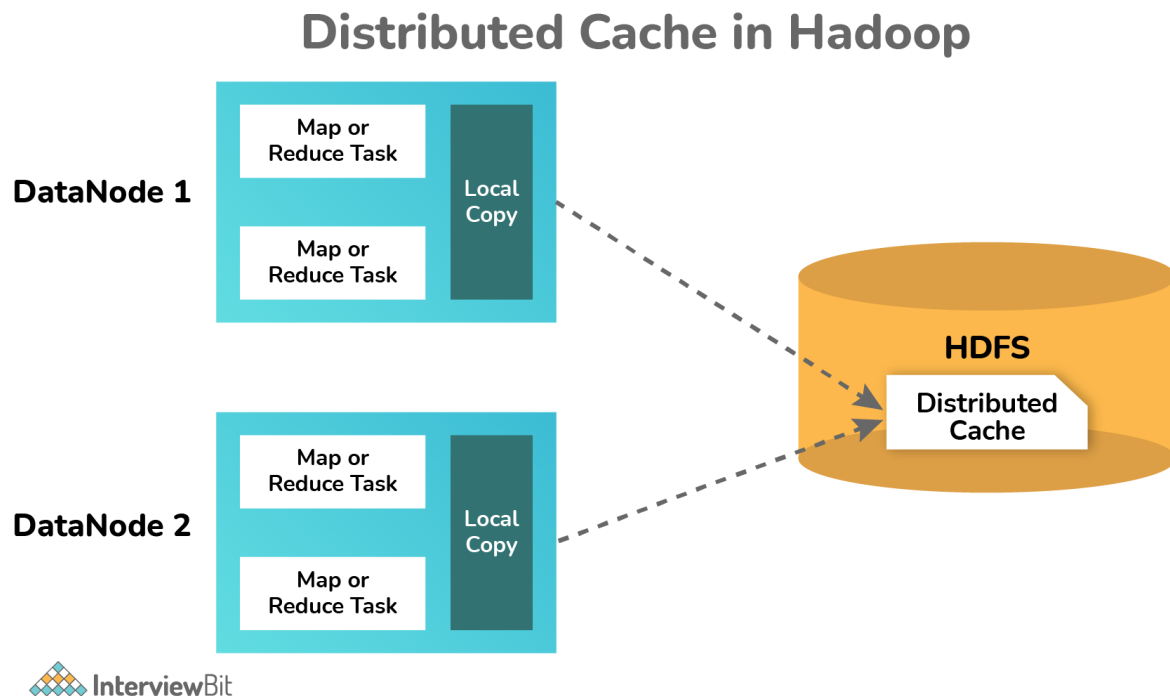
137. Explain the actions followed by a Jobtracker in Hadoop.

- 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.

138. 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.



139. 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.

140. 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.

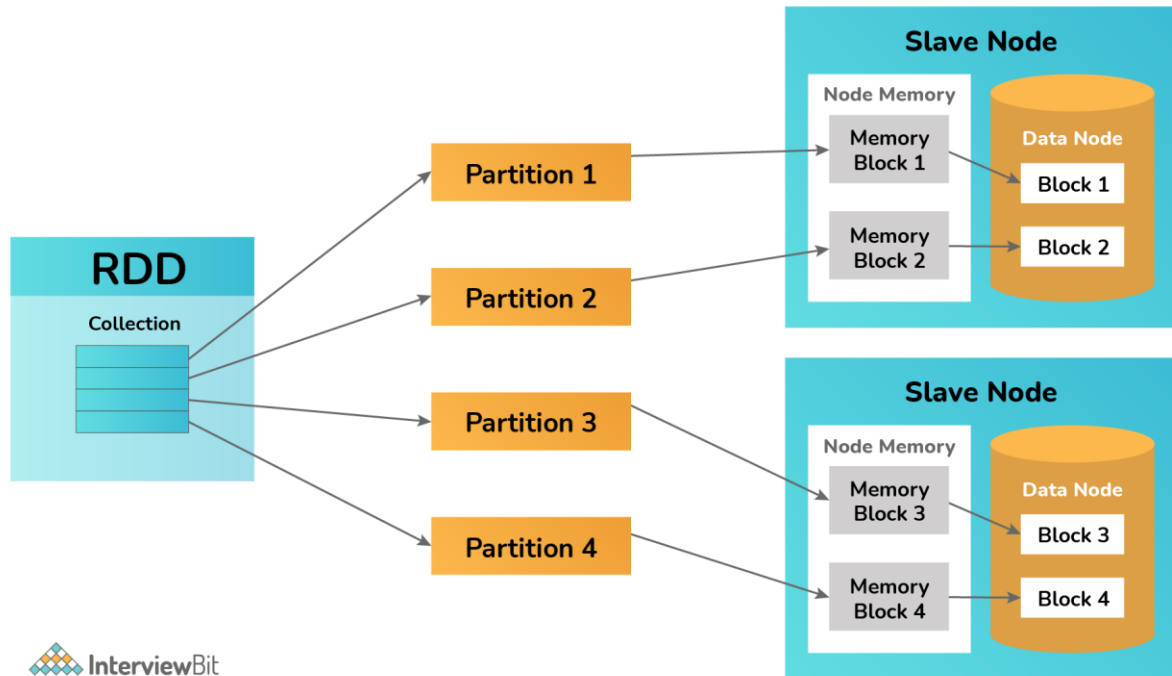
141. Mention the main Configuration parameters that has to be specified by the user to run MapReduce.

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.

142. 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.

143. 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.

144. What applications are supported by Apache Hive?

The applications that are supported by Apache Hive are,


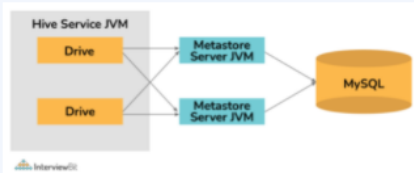
- Java
- PHP
- Python
- C++

- Ruby

145. 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.

146. Compare differences between Local Metastore and Remote Metastore

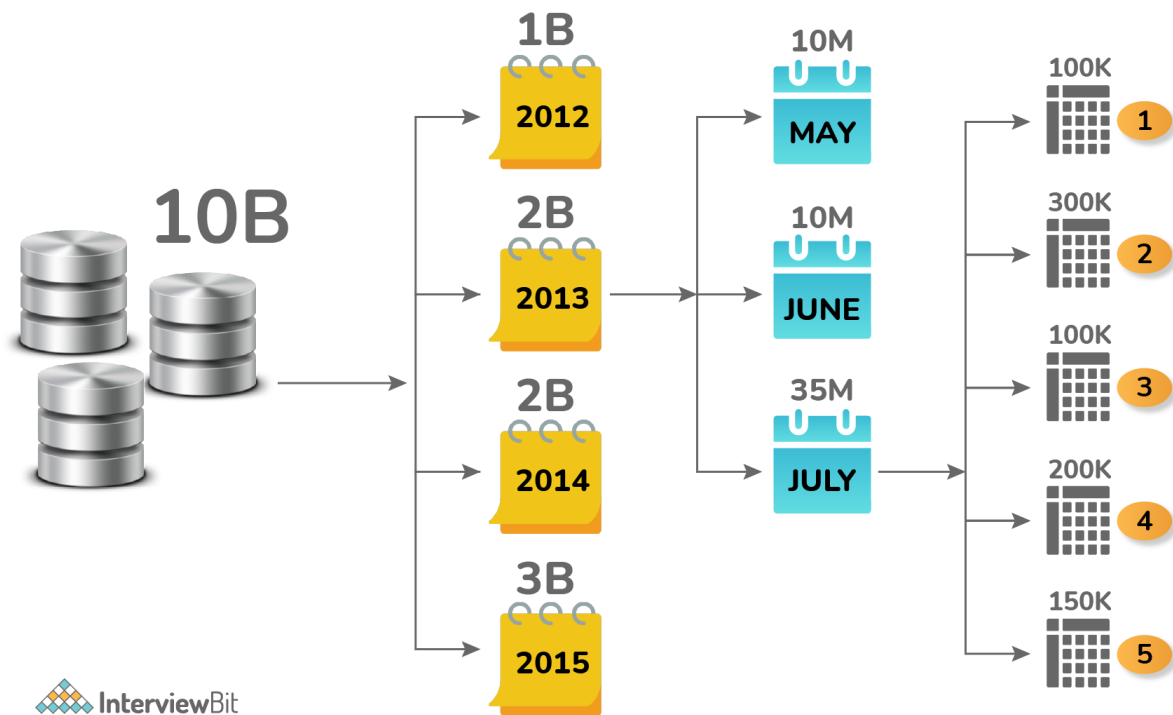
Local Metastore	Remote Metastore
Local metastore is a metastore service that runs in the same JVM in which the Hive service is running.	Remote Metastore has its own separate JVM which runs with its own JVM process.
It can also connect to a separate database running in a separate JVM in the same or separate machine.	The main advantage of remote mode over Local mode is that Remote mode does not acquire the administrator to share JDBC login information for the metastore database.
	

147. Are Multiline Comments supported in Hive? Why?

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

148. 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.

149. 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.

150. 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.

Inner vs, Outer Bag

```
grunt> chars = LOAD '/training/playArea/pig/b.txt' AS
(c:chararray);
grunt> charGroup = GROUP chars by c;
grunt> dump charGroup;
(a, {{(a), (a), (a)}})
(c, {{(c), (c)}})
(i, {{(i), (i), (i)}})
(k, {{(k), (k), (k), (k)}})
(l, {{(l), (l)}})
```

Inner Bag

Outer Bag

Pig Latin - FOREACH

• FOREACH<bag>GENERATE<data>

- Iterate over each element in the bag and produce a result
- Ex: grunt>result = FOREACH bag GENERATE f1;

```
grunt> records = LOAD 'data/a.txt' AS (c:chararray, i:int);
grunt> dump records;
(a, 1)
(b, 4)
(c, 9)
(k, 6)
grunt> counts = foreach records generate i;
grunt> dump counts;
(1)
(4)
(9)
(6)
```

For each row emit 'i' field

151. 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.

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

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

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

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

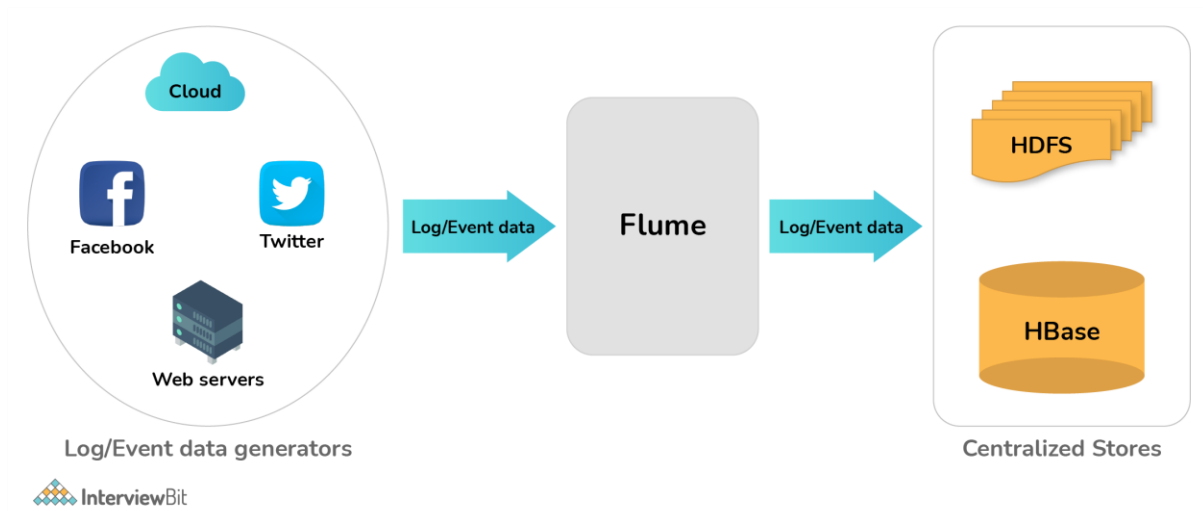
- Delimited Text File format
- Sequence File Format

154. 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.

155. 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, events from various references to a centralized data store.

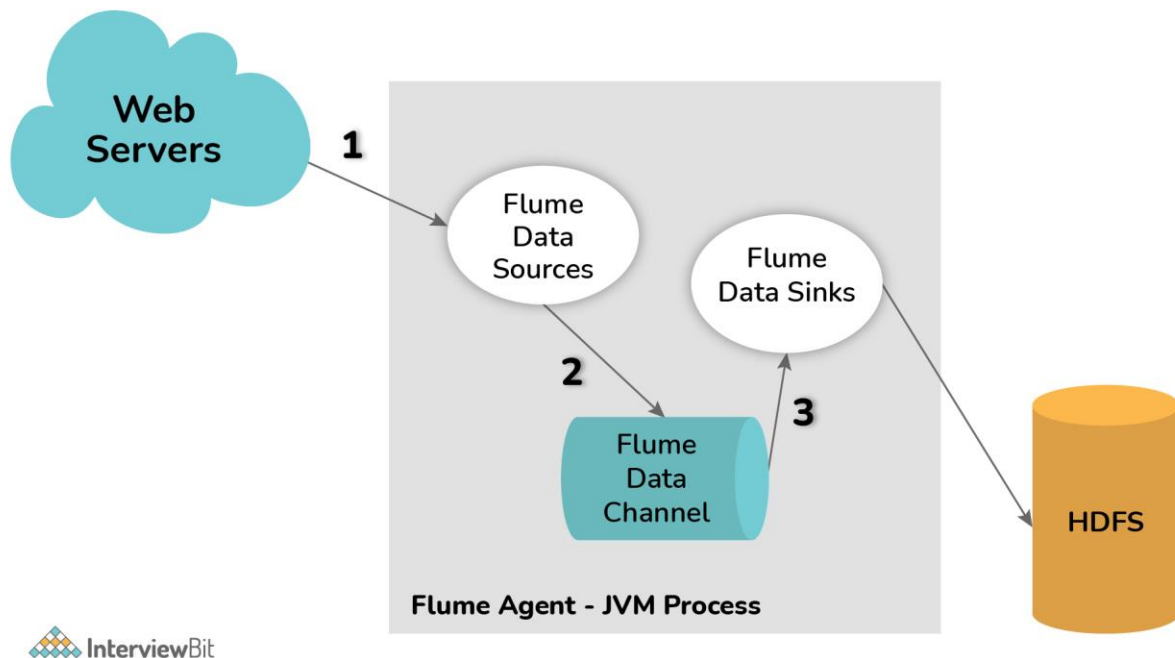


Flume is a very stable, distributed, and configurable tool. It is generally designed to copy streaming data (log data) from various web servers to HDFS.

156. 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.

157. 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.

Hadoop Interview Questions

158. What are the differences between Hadoop and Spark?

Criteria	Hadoop	Spark
Dedicated storage	HDFS	None
Speed of processing	Average	Excellent
Libraries	Separate tools available	Spark Core, SQL, Streaming, MLlib, and GraphX

159. What are the real-time industry applications of Hadoop?

Hadoop, well known as Apache Hadoop, is an open-source software platform for scalable and distributed computing of large volumes of data. It provides rapid, high performance, and cost-effective analysis of structured and unstructured data generated on digital platforms and within the enterprise. It is used in almost all departments and sectors today.

Here are some of the instances where Hadoop is used:

- Managing traffic on streets
- Streaming processing
- Content management and archiving e-mails
- Processing rat brain neuronal signals using a Hadoop computing cluster
- Fraud detection and prevention
- Advertisements targeting platforms are using Hadoop to capture and analyze click stream, transaction, video, and social media data
- Managing content, posts, images, and videos on social media platforms
- Analyzing customer data in real time for improving business performance
- Public sector fields such as intelligence, defense, cyber security, and scientific research
- Getting access to unstructured data such as output from medical devices, doctor's notes, lab results, imaging reports, medical correspondence, clinical data, and financial data

160. How is Hadoop different from other parallel computing systems?

Hadoop is a distributed file system that lets you store and handle massive amounts of data on a cloud of machines, handling data redundancy.

The primary benefit of this is that since data is stored in several nodes, it is better to process it in a distributed manner. Each node can process the data stored on it instead of spending time on moving the data over the network.

On the contrary, in the relational database computing system, we can query data in real time, but it is not efficient to store data in tables, records, and columns when the data is huge.

Hadoop also provides a scheme to build a column database with Hadoop HBase for runtime queries on rows.

161. What is Hadoop and what are its components?

Apache Hadoop is the solution for dealing with Big Data. It is an open-source framework that offers several tools and services to store, manage, process, and analyze Big Data. This allows organizations to make significant business decisions in an effective and efficient manner, which was not possible with traditional methods and systems.

Listed below are the main components of Hadoop:

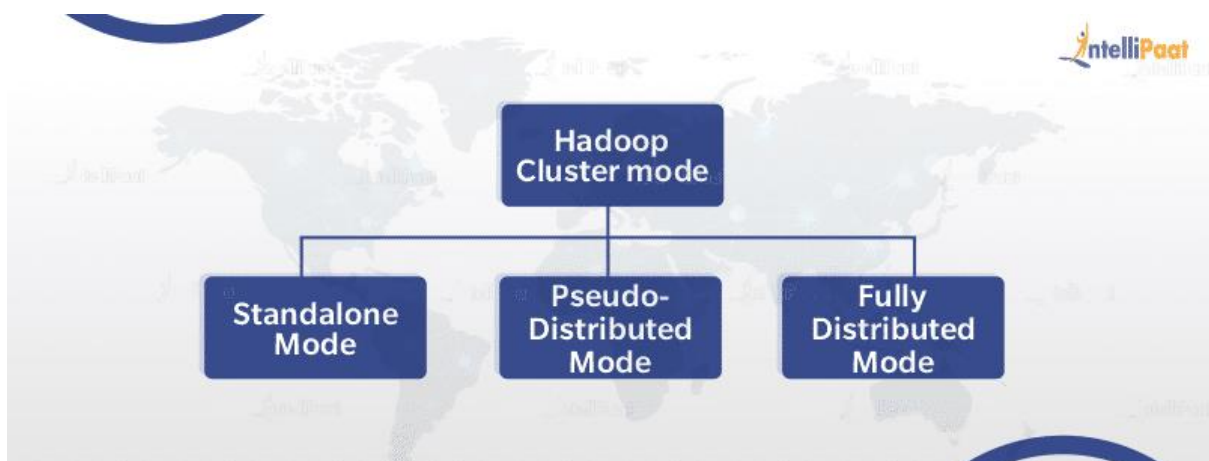
- HDFS: HDFS or Hadoop Distributed File System is Hadoop's storage unit.
- MapReduce: MapReduce the Hadoop's processing unit.
- YARN: YARN is the resource management unit of Apache Hadoop.

162. What is HBase?

Apache HBase is a distributed, open-source, scalable, and multidimensional database of NoSQL that is based on Java. It runs on HDFS and offers Google BigTable-like abilities and functionalities to Hadoop. Moreover, its fault-tolerant nature helps in storing large volumes of sparse data sets. It gets low latency and high throughput by offering faster access to large datasets for read/write functions.

163. In what all modes Hadoop can be run?

Hadoop can be run in three modes:



- **Standalone mode:** The default mode of Hadoop, it uses local file system for input and output operations. This mode is mainly used for the debugging purpose, and it does not support the use of HDFS. Further, in this mode, there is no custom configuration required for `mapred-site.xml`, `core-site.xml`, and `hdfs-site.xml` files. This mode works much faster when compared to other modes.
- **Pseudo-distributed mode (Single-node Cluster):** In this case, you need configuration for all the three files mentioned above. In this case, all daemons are running on one node, and thus both Master and Slave nodes are the same.

Fully distributed mode (Multi-node Cluster): This is the production phase of Hadoop (what Hadoop is known for) where data is used and distributed across several nodes on a Hadoop cluster. Separate nodes are allotted as Master and Slave.

164. What is the difference between RDBMS and Hadoop?

Following are some of the differences between RDBMS (Relational Database Management) and Hadoop based on various factors:

	RDBMS	Hadoop
Data Types	It relies on structured data and the data schema is always known.	Hadoop can store structured, unstructured, and semi-structured data.
Cost	Since it is licensed, it is a paid software.	It is a free open-source framework.
Processing	It offers little to no capabilities for processing.	It supports data processing for data distributed in a parallel manner across the cluster.
Read vs Write Schema	It follows 'schema on write', allowing the validation of schema to be done before data loading.	It supports the policy of schema on read.
Read/Write Speed	Reads are faster since the data schema is known.	Writes are faster since schema validation does not take place during HDFS write.
Best Use Case	It is used for Online Transactional Processing (OLTP) systems.	It is used for data analytics, data discovery, and OLAP systems.

165. Explain the major difference between HDFS block and InputSplit.

In simple terms, a block is the physical representation of data while split is the logical representation of data present in the block. Split acts as an intermediary between the block and the mapper.

Suppose we have two blocks:

Block 1: ii nntteell

Block 2: li ppaatt

Now considering the map, it will read Block 1 from ii to ll but does not know how to process Block 2 at the same time. Here comes Split into play, which will form a logical group of Block 1 and Block 2 as a single block.

It then forms a key-value pair using InputFormat and records reader and sends map for further processing with InputSplit. If you have limited resources, you can increase the split size to limit the number of maps. For instance, if there are 10 blocks of 640 MB (64 MB each) and there are limited resources, you can assign 'split size' as 128 MB. This will form a logical group of 128 MB, with only 5 maps executing at a time.

However, if the 'split size' property is set to false, the whole file will form one InputSplit and is processed by a single map, consuming more time when the file is bigger.

166. What is distributed cache? What are its benefits?

Distributed cache in Hadoop is a service by MapReduce framework to cache files when needed.

Once a file is cached for a specific job, Hadoop will make it available on each DataNode both in system and in memory, where map and reduce tasks are executing. Later, you can easily access and read the cache file and populate any collection (like array, hashmap) in your code.



Benefits of using distributed cache are as follows:

- It distributes simple, read-only text/data files and/or complex types such as jars, archives, and others. These archives are then unarchived at the slave node.
- Distributed cache tracks the modification timestamps of cache files, which notify that the files should not be modified until a job is executed.

167. What is a Combiner?

A Combiner is a mini version of a reducer that is used to perform local reduction processes. The mapper sends the input to a specific node of the Combiner which later sends the respective output to the reducer. It also reduces the quantum of data that needs to be sent to the reducers, improving the efficiency of MapReduce.

168. What are the various components of Apache HBase?

There are three main components of Apache HBase that are mentioned below:

- HMaster: It manages and coordinates the Region Server just like NameNode manages DataNodes in HDFS.
- Region Server: It is possible to divide a table into multiple regions and the Region Server makes it possible to serve a group of regions to the clients.
- ZooKeeper: Zookeeper is a coordinator in the distributed environment of HBase. It communicates through the sessions to maintain the state of the server in the cluster.

169. What are the components of Apache HBase's Region Server?

Following are the components of the Region Server of HBase:

- BlockCache: It resides on Region Server and stores data in the memory that is read frequently.

- WAL: WAL or Write Ahead Log is a file that is attached to each Region Server located in the distributed environment.
- MemStore: MemStore is the write cache that stores the input data before it is stored in the disk or permanent memory.
- HFile: HDFS stores the HFile that stores the cells on the disk.

170. What are the various schedulers in YARN?

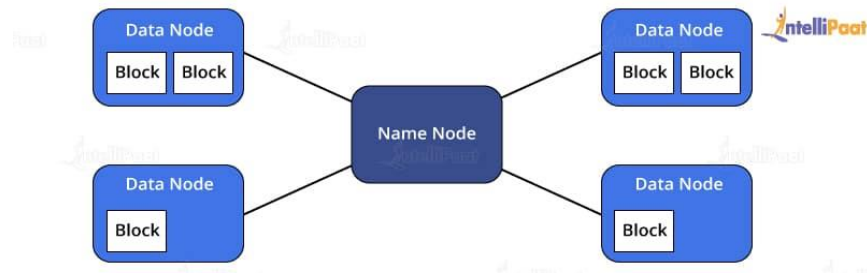
Mentioned below are the numerous schedulers that are available in YARN:

- FIFO Scheduler: FIFO or first in, first out scheduler places all the applications in a single queue and executes them in the same order as their submission. As it can block short applications due to long-running applications, it is less efficient and desirable by professionals.
- Capacity Scheduler: A different queue makes it possible to start executed short-term jobs as soon as they are submitted. Unlike in FIFO Scheduler, the long-term tasks are completed later in Capacity Scheduler.
- Fair Scheduler: Fair Scheduler, as the name suggests, works fairly. It balances the resources dynamically between all the running jobs and is not required to reserve a specific capacity for them.

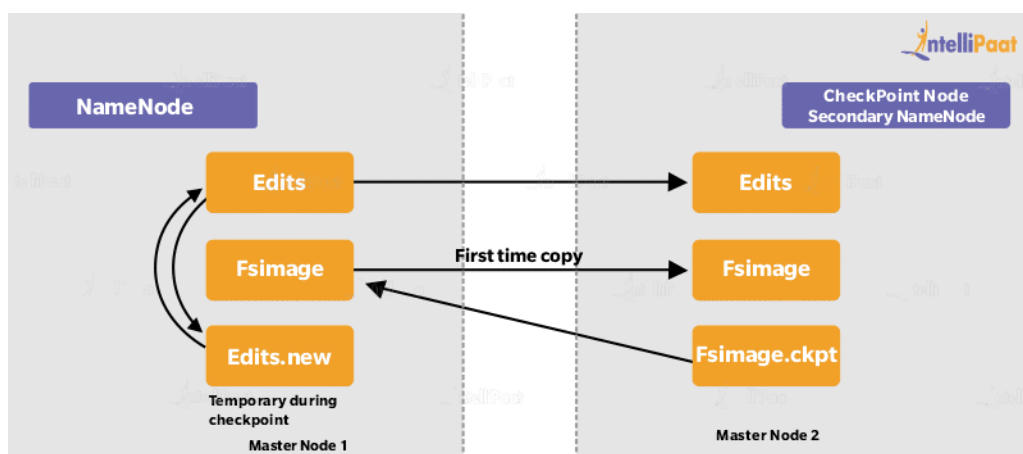
171. Explain the difference between NameNode, Checkpoint NameNode, and Backup Node.

- NameNode is the core of HDFS that manages the metadata—the information of which file maps to which block locations and which blocks are stored on which DataNode. In simple terms, it's the data about the data being stored. NameNode supports a directory tree-like structure consisting of all the files present in HDFS on a Hadoop cluster. It uses the following files for namespace:
 - fsimage file: It keeps track of the latest Checkpoint of the namespace.

- edits file: It is a log of changes that have been made to the namespace since Checkpoint.

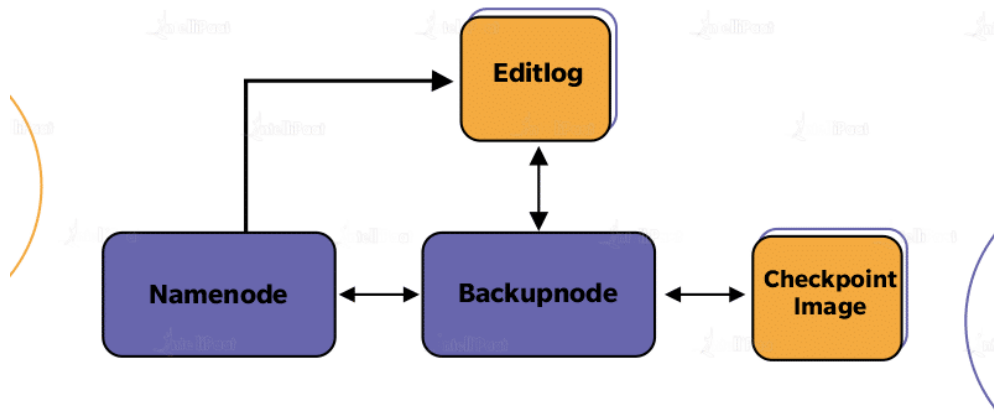


- Checkpoint NameNode has the same directory structure as NameNode and creates Checkpoints for namespace at regular intervals by downloading the fsimage, editing files, and margining them within the local directory. The new image after merging is then uploaded to NameNode. There is a similar node like Checkpoint, commonly known as the Secondary Node, but it does not support the 'upload to NameNode' functionality.



- Backup Node provides similar functionality as Checkpoint, enforcing synchronization with NameNode. It maintains an up-to-date in-memory copy of the file system namespace and doesn't require getting hold of changes after regular intervals. The Backup Node needs to save the current state in-memory to an image file to

create a new Checkpoint.



172. What are the most common input formats in Hadoop?

There are three most common input formats in Hadoop:

- Text Input Format: Default input format in Hadoop
- Key-Value Input Format: Used for plain text files where the files are broken into lines
- Sequence File Input Format: Used for reading files in sequence

173. How to execute a Pig script?

The three methods listed below enables users to execute a Pig script:

- Grunt shell
- Embedded script
- Script file

174. What is Apache Pig and why is it preferred over MapReduce?

Pig is a Hadoop-based platform that allows professionals to analyze large sets of data and represent them as data flows. Pig reduces the complexities that

are required while writing a program in MapReduce, giving it an edge over MapReduce.

Following are some of the reasons why Pig is more preferable as compared to MapReduce:

- While Pig is a language for high-level data flow, MapReduce paradigm for low-level data processing
- Without the need to write complex Java code in MapReduce, a similar result can easily be achieved in Pig
- Pig approximately reduces the code length by 20 times, reducing the time taken for development by about 16 times than MapReduce
- Pig offers built-in functionalities to perform numerous operations, including sorting, filters, joins, ordering, and more which are extremely difficult to perform in MapReduce
- Unlike MapReduce, Pig provides various nested data types, such as bags, maps, and tuples.

175. Mention some commands in YARN to check application status and to kill an application.

The YARN commands are mentioned below as per their functionalities:

1. yarn application - status ApplicationID

This command allows professionals to check the application status.

2. yarn application - kill ApplicationID

The command mentioned above enables users to kill or terminate a particular application.

176. What are the different components of Hive query processors?

There are numerous components that are used in Hive query processors and they are mentioned below:

- User-defined functions

- Semantic analyzer
- Optimizer
- Physical plan generation
- Logical plan generation
- Type checking
- Execution engine
- Parser
- Operators

177. Define DataNode. How does NameNode tackle DataNode failures?

DataNode stores data in HDFS; it is a node where actual data resides in the file system. Each DataNode sends a heartbeat message to notify that it is alive. If the NameNode does not receive a message from the DataNode for 10 minutes, the NameNode considers the DataNode to be dead or out of place and starts the replication of blocks that were hosted on that DataNode such that they are hosted on some other DataNode. A BlockReport contains a list of the all blocks on a DataNode. Now, the system starts to replicate what were stored in the dead DataNode.

The NameNode manages the replication of the data blocks from one DataNode to another. In this process, the replication data gets transferred directly between DataNodes such that the data never passes the NameNode.

178. What is the significance of Sqoop's eval tool?

The eval tool in Sqoop enables users to carry out user-defined queries on the corresponding database servers and check the outcome in the console.

179. What are the differences between Relational Databases and HBase?

The differences between Relational Databases and HBase are mentioned below:

Relational Database	HBase
It is schema-based.	It has no schema.
It is row-oriented.	It is column-oriented.
It stores normalized data.	It stores denormalized data.
It consists of thin tables.	It consists of sparsely populated tables.
There is no built-in support or provision for automatic partitioning.	It supports automated partitioning.

180. What are the core methods of a Reducer?

The three core methods of a Reducer are as follows:

1. `setup()`: This method is used for configuring various parameters such as input data size and distributed cache.
`public void setup (context)`
2. `reduce()`: Heart of the Reducer is always called once per key with the associated reduced task.
`public void reduce(Key, Value, context)`
3. `cleanup()`: This method is called to clean the temporary files, only once at the end of the task.
`public void cleanup (context)`

181. What are the differences between MapReduce and Pig?

The differences between MapReduce and Pig are mentioned below:

MapReduce	Pig
It has more lines of code as compared to Pig.	It has fewer lines of code.
It is a low-level language that makes it difficult to perform operations like join.	This high-level language makes it easy to perform join and other similar operations.
The compiling process is time-consuming.	During execution, all the Pig operators are internally converted into a MapReduce job.
A MapReduce program that is written in a particular version of Hadoop may not work in others.	It works in all Hadoop versions.

182. What is a SequenceFile in Hadoop?

Extensively used in MapReduce I/O formats, SequenceFile is a flat file containing binary key-value pairs. The map outputs are stored as SequenceFile internally. It provides Reader, Writer, and Sorter classes. The three SequenceFile formats are as follows:

1. Uncompressed key-value records
2. Record compressed key-value records—only 'values' are compressed here
3. Block compressed key-value records—both keys and values are collected in 'blocks' separately and compressed. The size of the 'block' is configurable

183. What do you mean by WAL in HBase?

WAL is otherwise referred to as Write Ahead Log. This file is attached to each Region Server present inside the distributed environment. It stores the new data which is yet to be kept in permanent storage. WAL is often used to recover data sets in case of any failure.

184. Explain the architecture of YARN and how it allocates various resources to applications.

There is an application, API, or client that communicates with the ResourceManager which then deals with allocating resources in the cluster. It has an awareness of the resources present with each node manager. There are two internal components of the ResourceManager: Application Manager and Scheduler. The scheduler is responsible for allocating resources to the numerous applications running in parallel based on their requirements. However, the scheduler does not track the application status.

The Application Manager accepts the submission of jobs and manages and reboots the application masters if there is a failure. It manages the applications' demand for resources and communicates with the scheduler to get the resources needed. It interacts with the NodeManager to manage and execute the tasks that monitor the jobs running. Moreover, it also monitors the resources utilized by each container.

A container consists of a set of resources, including CPU, RAM, and network bandwidth. It allows the applications to use a predefined number of resources.

The ResourceManager sends a request to the NodeManager to keep a few resources to process as soon as there is a job submission. Later, the NodeManager assigns an available container to carry out the processing. The ResourceManager then starts the application master to deal with the execution and it runs in one of the given containers. The rest of the containers available are used for the execution process. This is the overall process of how YARN allocates resources to applications via its architecture.

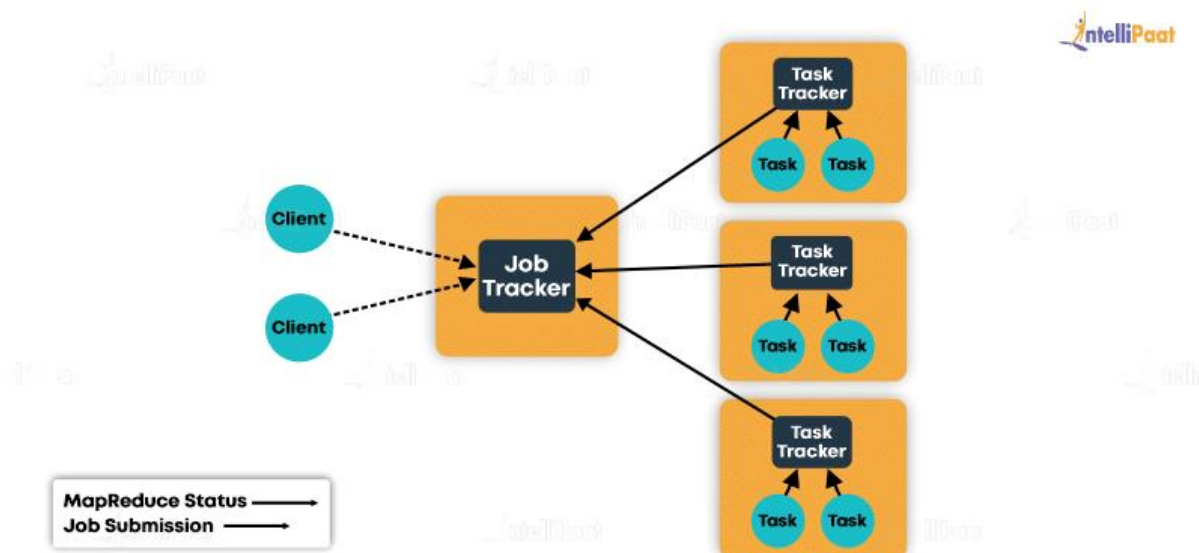
185. What is the difference between Sqoop and Flume?

Following are the various differences between Sqoop and Flume:

Sqoop	Flume
It works with NoSQL databases and RDBMS for importing and exporting data.	It works with streaming data which is regularly generated in the Hadoop environment.
In Sqoop, loading data is not event-driven.	In Flume, loading data is event-driven.
It deals with data sources that are structured and Sqoop connectors help in extracting data from them.	It extracts streaming data from application servers or web servers.
It takes data from RDBMS, imports it into HDFS, and exports it back to RDBMS.	Data from multiple sources flows into HDFS.

186. What is the role of a JobTracker in Hadoop?

A JobTracker's primary function is resource management (managing the TaskTrackers), tracking resource availability, and task life cycle management (tracking the tasks' progress and fault tolerance).



- It is a process that runs on a separate node, often not on a DataNode.
- The JobTracker communicates with the NameNode to identify data location.

- It finds the best TaskTracker nodes to execute the tasks on the given nodes.
- It monitors individual TaskTrackers and submits the overall job back to the client.
- It tracks the execution of MapReduce workloads local to the slave node.

187. What are the components of the architecture of Hive?

- User Interface: It requests for the execute interface for the driver and also builds a session for this query. Further, the query is sent to the compiler in order to create an execution plan for the same.
- Metastore: It stores the metadata and transfers it to the compiler to execute a query.
- Compiler: It creates the execution plan. The compiler consists of a DAG of stages wherein each stage can either be a map, a metadata operation, or reduces an operation or a job on HDFS.
- Execution Engine: This engine bridges the gap between Hadoop and Hive and helps in processing the query. It communicates with the metastore bidirectionally in order to perform various tasks.

188. Is it possible to import or export tables in HBase?

Yes, you can import and export tables between HBase clusters using the commands listed below:

For export:

```
hbase org.apache.hadoop.hbase.mapreduce.Export "table name" "target export location"
```

For import:

```
create 'emp_table_import', {NAME => 'myfam', VERSIONS => 10}
```

```
hbase org.apache.hadoop.hbase.mapreduce.Import "table name" "target  
import location"
```

189. Why does Hive not store metadata in HDFS?

Hive stores the data of HDFS and the metadata is stored in the RDBMS or it is locally stored. HDFS does not store this metadata because the read/write operations in HDFS take a lot of time. This is why Hive uses RDBMS to store this metadata in the megastore rather than HDFS. This makes the process faster and enables you to achieve low latency.

190. What are the significant components in the execution environment of Pig?

The main components of a Pig execution environment are as follows:

- Pig Scripts: They are written in Pig with the help of UDFs and built-in operators after which they are sent to the execution environment.
- Parser: It checks the script syntax and completes type checking. Parser's output is a Directed Acyclic Graph (DAG).
- Optimizer: It conducts optimization with operations like transform, merges, and more, to minimize the data in the pipeline.
- Compiler: The compiler automatically converts the code that is optimized into a MapReduce job.
- Execution Engine: The MapReduce jobs are sent to these engines in order to get the required output.

191. What are the components of HBase?

The major components of HBase are as follows:

- Region Server: It consists of HBase tables which are horizontally divided into Regions as per their key values. Moreover, it runs on every single node to estimate the region size. All region servers are

worker nodes that work on various client requests, like update, delete, read, and write.

- **HMaster:** HMaster assigns regions to the respective region servers for the purpose of load balancing. Further, it monitors the Hadoop cluster and can be used by clients if and when a schema or a metadata operation needs to be changed.
- **ZooKeeper:** ZooKeeper offers a distributed service to coordinate and manage the state of the server in a cluster. It also checks which servers are available and sends notifications of server failure. Region servers inform Zookeeper their status to indicate whether they are ready for the read and write operations.

192. What is the command used to open a connection in HBase?

The code mentioned below can be used to open a connection in HBase:

```
Configuration myConf = HBaseConfiguration.create();
```

```
HTableInterface usersTable = new HTable(myConf, "users");
```

193. What is the use of RecordReader in Hadoop?

Though InputSplit defines a slice of work, it does not describe how to access it. Here is where the RecordReader class comes into the picture, which takes the byte-oriented data from its source and converts it into record-oriented key-value pairs such that it is fit for the Mapper task to read it. Meanwhile, InputFormat defines this Hadoop RecordReader instance.

194. How does Sqoop import or export data between HDFS and RDBMS?

The steps followed by Sqoop to import and export data between HDFS and RDBMS using its architecture are listed below:

- Search the database to collect metadata.
- Sqoop splits the input dataset and makes use of respective map jobs to push these splits to HDFS.

- Search the database to collect metadata.
- Sqoop splits the input dataset and makes use of respective map jobs to push these splits to RDBMS. Sqoop exports back the Hadoop files to the RDBMS tables.

195. What is Speculative Execution in Hadoop?

One limitation of Hadoop is that by distributing the tasks on several nodes, there are chances that few slow nodes limit the rest of the program. There are various reasons for the tasks to be slow, which are sometimes not easy to detect. Instead of identifying and fixing the slow-running tasks, Hadoop tries to detect when the task runs slower than expected and then launches other equivalent tasks as backup. This backup mechanism in Hadoop is speculative execution.

It creates a duplicate task on another disk. The same input can be processed multiple times in parallel. When most tasks in a job comes to completion, the speculative execution mechanism schedules duplicate copies of the remaining tasks (which are slower) across the nodes that are free currently. When these tasks are finished, it is intimated to the JobTracker. If other copies are executing speculatively, Hadoop notifies the TaskTrackers to quit those tasks and reject their output.

Speculative execution is by default true in Hadoop. To disable it, we can set `mapred.map.tasks.speculative.execution` and `mapred.reduce.tasks.speculative.execution` JobConf options to false.

196. What is Apache Oozie?

Oozie is nothing but a scheduler that helps to schedule jobs in Hadoop and bundles them as a single logical work. Oozie jobs can largely be divided into the following two categories:

- Oozie Workflow: These jobs are a set of sequential actions that need to be executed.
- Oozie Coordinator: These jobs are triggered as and when there is data availability for them until which, it rests.

197. What happens if you try to run a Hadoop job with an output directory that is already present?

It will throw an exception saying that the output file directory already exists.

To run the MapReduce job, you need to ensure that the output directory does not exist in the HDFS.

To delete the directory before running the job, we can use shell:

```
Hadoop fs -rmr /path/to/your/output/
```

Or the Java API:

```
FileSystem.getLocal(conf).delete(outputDir, true);
```

198. How can you debug Hadoop code?

First, we should check the list of MapReduce jobs currently running. Next, we need to see that there are no orphaned jobs running; if yes, we need to determine the location of RM logs.

1. Run:

```
ps -ef | grep -I ResourceManager
```

Then, look for the log directory in the displayed result. We have to find out the job ID from the displayed list and check if there is any error message associated with that job.

2. On the basis of RM logs, we need to identify the worker node that was involved in the execution of the task.
3. Now, we will login to that node and run the below code:

```
ps -ef | grep -iNodeManager
```

4. Then, we will examine the Node Manager log. The majority of errors come from the user-level logs for each MapReduce job.

199. How to configure Replication Factor in HDFS?

The `hdfs-site.xml` file is used to configure HDFS. Changing the `dfs.replication` property in `hdfs-site.xml` will change the default replication for all the files placed in HDFS.

We can also modify the replication factor on a per-file basis using the below:

```
Hadoop FS Shell:[training@localhost ~]$ hadoopfs -setrep -w 3 /my/fileConversely,
```

We can also change the replication factor of all the files under a directory.

```
[training@localhost ~]$ hadoopfs -setrep -w 3 -R /my/dir
```

Learn more about Hadoop from this [Big Data Hadoop Training in New York](#) to get ahead in your career!

200. How to compress a Mapper output not touching Reducer output?

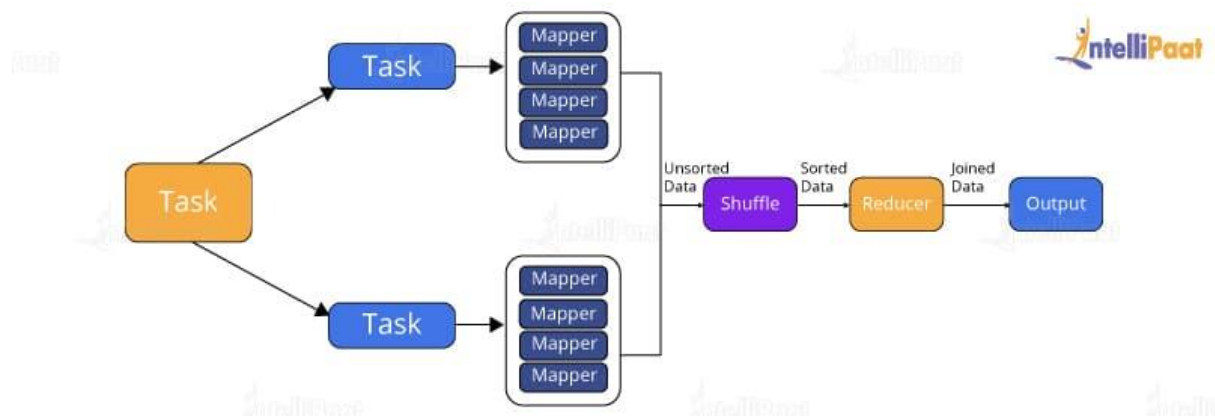
To achieve this compression, we should set:

```
conf.set("mapreduce.map.output.compress", true)
```

```
conf.set("mapreduce.output.fileoutputformat.compress", false)
```

201. What is the difference between Map-side Join and Reduce-side Join?

Map-side Join at Map side is performed when data reaches the Map. We need a strict structure for defining Map-side Join.



On the other hand, Reduce-side Join (Repartitioned Join) is simpler than Map-side Join since here the input datasets need not be structured. However, it is less efficient as it will have to go through sort and shuffle phases, coming with network overheads.

202. How can you transfer data from Hive to HDFS?

By writing the query:

```
hive> insert overwrite directory '/' select * from emp;
```

We can write our query for the data we want to import from Hive to HDFS. The output we receive will be stored in part files in the specified HDFS path.

203. Which companies use Hadoop?

Yahoo! (it is the biggest contributor to the creation of Hadoop; its search engine uses Hadoop); Facebook (developed Hive for analysis); Amazon; Netflix; Adobe; eBay; Spotify; Twitter; and Adobe.

204. What is Hadoop Map Reduce?

For processing large data sets in parallel across a Hadoop cluster, Hadoop MapReduce framework is used. Data analysis uses a two-step map and reduce process.

205. How Hadoop MapReduce works?

In MapReduce, during the map phase, it counts the words in each document, while in the reduce phase it aggregates the data as per the document spanning the entire collection. During the map phase, the input data is divided into splits for analysis by map tasks running in parallel across Hadoop framework.

206. Explain what is shuffling in MapReduce?

The process by which the system performs the sort and transfers the map outputs to the reducer as inputs is known as the shuffle.

207. Explain what is distributed Cache in MapReduce Framework?

Distributed Cache is an important feature provided by the MapReduce framework. When you want to share some files across all nodes in Hadoop Cluster, Distributed Cache is used. The files could be an executable jar files or simple properties file.

208. Explain what is NameNode in Hadoop?

NameNode in Hadoop is the node, where Hadoop stores all the file location information in HDFS (Hadoop Distributed File System). In other words, NameNode is the centerpiece of an HDFS file system. It keeps the record of all the files in the file system and tracks the file data across the cluster or multiple machines

209. Explain what is JobTracker in Hadoop? What are the actions followed by Hadoop?

In Hadoop for submitting and tracking MapReduce jobs, JobTracker is used. Job tracker run on its own JVM process

Job Tracker performs following actions in Hadoop

- Client application submit jobs to the job tracker
- JobTracker communicates to the Name node to determine data location
- Near the data or with available slots JobTracker locates TaskTracker nodes
- On chosen TaskTracker Nodes, it submits the work
- When a task fails, Job tracker notifies and decides what to do then.
- The TaskTracker nodes are monitored by JobTracker

210. Explain what is heartbeat in HDFS?

Heartbeat is referred to a signal used between a data node and Name node, and between task tracker and job tracker, if the Name node or job tracker does not respond to the signal, then it is considered there is some issues with data node or task tracker

211. Explain what combiners are and when you should use a combiner in a MapReduce Job?

To increase the efficiency of MapReduce Program, Combiners are used. The amount of data can be reduced with the help of combiner's that need to be transferred across to the reducers. If the operation performed is commutative and associative you can use your reducer code as a combiner. The execution of combiner is not guaranteed in Hadoop

212. What happens when a data node fails?

When a data node fails

- Jobtracker and namenode detect the failure
- On the failed node all tasks are re-scheduled
- Namenode replicates the user's data to another node

213. Explain what is Speculative Execution?

In Hadoop during Speculative Execution, a certain number of duplicate tasks are launched. On a different slave node, multiple copies of the same map or reduce task can be executed using Speculative Execution. In simple words, if a particular drive is taking a long time to complete a task, Hadoop will create a duplicate task on another disk. A disk that finishes the task first is retained and disks that do not finish first are killed.

214. Explain what are the basic parameters of a Mapper?

The basic parameters of a Mapper are

- LongWritable and Text
- Text and IntWritable

215. Explain what is the function of MapReduce partitioner?

The function of MapReduce partitioner is to make sure that all the value of a single key goes to the same reducer, eventually which helps even distribution of the map output over the reducers

216. Explain what is a difference between an Input Split and HDFS Block?

The logical division of data is known as Split while a physical division of data is known as HDFS Block

217. Explain what happens in text format?

In text input format, each line in the text file is a record. Value is the content of the line while Key is the byte offset of the line. For instance, Key: longWritable, Value: text

218. Mention what are the main configuration parameters that user need to specify to run MapReduce Job?

The user of the MapReduce framework needs to specify

- 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
- JAR file containing the mapper, reducer and driver classes

219. Explain what is WebDAV in Hadoop?

To support editing and updating files WebDAV is a set of extensions to HTTP. On most operating system WebDAV shares can be mounted as filesystems, so it is possible to access HDFS as a standard filesystem by exposing HDFS over WebDAV.

220. Explain what is Sqoop in Hadoop?

To transfer the data between Relational database management (RDBMS) and Hadoop HDFS a tool is used known as Sqoop. Using Sqoop data can be transferred from RDMS like MySQL or Oracle into HDFS as well as exporting data from HDFS file to RDBMS

221. Explain how JobTracker schedules a task?

The task tracker sends out heartbeat messages to Jobtracker usually every few minutes to make sure that JobTracker is active and functioning. The message also informs JobTracker about the number of available slots, so the JobTracker can stay up to date with wherein the cluster work can be delegated

222. Explain what is Sequencefileinputformat?

Sequencefileinputformat is used for reading files in sequence. It is a specific compressed binary file format which is optimized for passing data between the output of one MapReduce job to the input of some other MapReduce job.

223. Explain what does the conf.setMapper Class do?

Conf.setMapperclass sets the mapper class and all the stuff related to map job such as reading data and generating a key-value pair out of the mapper

224. Explain what is Hadoop?

It is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides enormous processing power and massive storage for any type of data.

225. Mention what is the difference between an RDBMS and Hadoop?

RDBMS	Hadoop
RDBMS is a relational database management system	Hadoop is a node based flat structure
It used for OLTP processing whereas Hadoop	It is currently used for analytical and for BIG DATA processing
In RDBMS, the database cluster uses the same data files stored in a shared storage	In Hadoop, the storage data can be stored independently in each processing node.
You need to preprocess data before storing it	you don't need to preprocess data before storing it

226. Mention Hadoop core components?

Hadoop core components include,

- HDFS
- MapReduce

227. What is NameNode in Hadoop?

NameNode in Hadoop is where Hadoop stores all the file location information in HDFS. It is the master node on which job tracker runs and consists of metadata.

228. Mention what are the data components used by Hadoop?

Data components used by Hadoop are

- Pig
- Hive

229. Mention what is the data storage component used by Hadoop?

The data storage component used by Hadoop is HBase.

230. Mention what are the most common input formats defined in Hadoop?

The most common input formats defined in Hadoop are;

- TextInputFormat
- KeyValueInputFormat
- SequenceFileInputFormat

231. In Hadoop what is InputSplit?

It splits input files into chunks and assigns each split to a mapper for processing.

232. For a Hadoop job, how will you write a custom partitioner?

You write a custom partitioner for a Hadoop job, you follow the following path

- Create a new class that extends Partitioner Class
- Override method getPartition
- In the wrapper that runs the MapReduce
- Add the custom partitioner to the job by using method set Partitioner Class or – add the custom partitioner to the job as a config file

233. For a job in Hadoop, is it possible to change the number of mappers to be created?

No, it is not possible to change the number of mappers to be created. The number of mappers is determined by the number of input splits.

234. Explain what is a sequence file in Hadoop?

To store binary key/value pairs, sequence file is used. Unlike regular compressed file, sequence file support splitting even when the data inside the file is compressed.

235. When Namenode is down what happens to job tracker?

Namenode is the single point of failure in HDFS so when Namenode is down your cluster will set off.

236. Explain how indexing in HDFS is done?

Hadoop has a unique way of indexing. Once the data is stored as per the block size, the HDFS will keep on storing the last part of the data which say where the next part of the data will be.

237. Explain is it possible to search for files using wildcards?

Yes, it is possible to search for files using wildcards.

238. List out Hadoop's three configuration files?

The three configuration files are

- core-site.xml
- mapred-site.xml
- hdfs-site.xml

239. Explain how can you check whether Namenode is working beside using the jps command?

Besides using the jps command, to check whether Namenode are working you can also use

/etc/init.d/hadoop-0.20-namenode status.

240. Explain what is “map” and what is "reducer" in Hadoop?

In Hadoop, a map is a phase in HDFS query solving. A map reads data from an input location, and outputs a key value pair according to the input type.

In Hadoop, a reducer collects the output generated by the mapper, processes it, and creates a final output of its own.

241. In Hadoop, which file controls reporting in Hadoop?

In Hadoop, the hadoop-metrics.properties file controls reporting.

242. For using Hadoop list the network requirements?

For using Hadoop the list of network requirements are:

- Password-less SSH connection
- Secure Shell (SSH) for launching server processes

243. Mention what is rack awareness?

Rack awareness is the way in which the namenode determines on how to place blocks based on the rack definitions.

244. Explain what is a Task Tracker in Hadoop?

A Task Tracker in Hadoop is a slave node daemon in the cluster that accepts tasks from a JobTracker. It also sends out the heartbeat messages to the JobTracker, every few minutes, to confirm that the JobTracker is still alive.

245. Mention what daemons run on a master node and slave nodes?

- Daemons run on Master node is "NameNode"
- Daemons run on each Slave nodes are "Task Tracker" and "Data"

246. Explain how can you debug Hadoop code?

The popular methods for debugging Hadoop code are:

- By using web interface provided by Hadoop framework
- By using Counters

247. Explain what is storage and compute nodes?

- The storage node is the machine or computer where your file system resides to store the processing data
- The compute node is the computer or machine where your actual business logic will be executed.

248. Mention what is the use of Context Object?

The Context Object enables the mapper to interact with the rest of the Hadoop system. It includes configuration data for the job, as well as interfaces which allow it to emit output.

249. Mention what is the next step after Mapper or MapTask?

The next step after Mapper or MapTask is that the output of the Mapper are sorted, and partitions will be created for the output.

250. Mention what is the number of default partitioner in Hadoop?

In Hadoop, the default partitioner is a “Hash” Partitioner.

251. Explain what is the purpose of RecordReader in Hadoop?

In Hadoop, the RecordReader loads the data from its source and converts it into (key, value) pairs suitable for reading by the Mapper.

252. Explain how is data partitioned before it is sent to the reducer if no custom partitioner is defined in Hadoop?

If no custom partitioner is defined in Hadoop, then a default partitioner computes a hash value for the key and assigns the partition based on the result.

253. Explain what happens when Hadoop spawned 50 tasks for a job and one of the task failed?

It will restart the task again on some other TaskTracker if the task fails more than the defined limit.

254. Mention what is the best way to copy files between HDFS clusters?

The best way to copy files between HDFS clusters is by using multiple nodes and the distcp command, so the workload is shared.

255. Mention what is the difference between HDFS and NAS?

HDFS data blocks are distributed across local drives of all machines in a cluster while NAS data is stored on dedicated hardware.

256. Mention how Hadoop is different from other data processing tools?

In Hadoop, you can increase or decrease the number of mappers without worrying about the volume of data to be processed.

257. Mention what job does the conf class do?

Job conf class separate different jobs running on the same cluster. It does the job level settings such as declaring a job in a real environment.

258. Mention what is the Hadoop MapReduce APIs contract for a key and value class?

For a key and value class, there are two Hadoop MapReduce APIs contract

- The value must be defining the `org.apache.hadoop.io.Writable` interface
- The key must be defining the `org.apache.hadoop.io.WritableComparable` interface

259. Mention what are the three modes in which Hadoop can be run?

The three modes in which Hadoop can be run are

- Pseudo distributed mode
- Standalone (local) mode
- Fully distributed mode

260. Mention what does the text input format do?

The text input format will create a line object that is an hexadecimal number. The value is considered as a whole line text while the key is considered as a line object. The mapper will receive the value as 'text' parameter while key as 'longwritable' parameter.

261. Mention how many InputSplits is made by a Hadoop Framework?

Hadoop will make 5 splits

- 1 split for 64K files
- 2 split for 65mb files
- 2 splits for 127mb files

262. Mention what is distributed cache in Hadoop?

Distributed cache in Hadoop is a facility provided by MapReduce framework. At the time of execution of the job, it is used to cache file. The Framework copies the necessary files to the slave node before the execution of any task at that node.

263. Explain how does Hadoop Classpath plays a vital role in stopping or starting in Hadoop daemons?

Classpath will consist of a list of directories containing jar files to stop or start daemons.

264. Explain Monad class?

Monad class is a class for wrapping of objects. E.g. identity with Unit & Bind with Map. It provides two operations as below:-

identity (return in Haskell, unit in Scala)

bind (>>= in Haskell, flatMap in Scala)

Scala doesn't have a built-in monad type, so we need to model the monad ourselves. However other subsidiaries of Scala like Scalaz have the monad built-in itself also it comes with theory family like applicatives , functors, monoids and so on.

The sample program to model monad with generic trait in Scala which provide method like unit() and flatMap() is below. Lets denote M in-short for monad.

```

trait M[A]

{

defflatMap[B](f: A => M[B]): M[B]

}

def unit[A](x: A): M[A]

```

265. Explain the reliability of Flume-NG data?

Apache Flume provides a reliable and distributed system for collecting, aggregating and moving large amounts of log data from many different sources to a centralized data store.

This work currently in progress and informally referred to as Flume NG. It has gone through two internal milestones – NG Alpha 1, and NG Alpha 2 and a formal incubator release of Flume NG is in the works.

The Core Concept of Flume-NG data are – Event, Flow, Client, Agent, Source, Channel and Sink. These core concept makes the architecture of Flume NG to achieve this objective.

266. What is Interceptor?

This is a Flume Plug-in that helps to listen any Incoming and alter event's content on the Fly.

e.g. Interceptor Implementation for JSON data.

267. What are the different Flume-NG Channel types?

The main channel types of Flume-NG are Memory Channel, JDBC Channel, Kafka Channel, File Channel, Spillable Memory Channel, Pseudo Transaction Channel.

In basic Flume, we have channel types like memory, JDBC, file and Kafka.

268. What is Base class in java?

A base class is also a class which facilitates the creation of other classes. In terms of object oriented programming, it is referred as derived class. This

helps to reuse the code implicitly from base class except constructors and destructors.

269. What is Base class in scala?

Base class concept is same for both java and scala. Only difference is in syntax. The Keywords in Scala are Base and Derived.

Ex:

```
abstract class Base( val x : String )  
  
final class Derived( x : String ) extends Base( "Base's " + x )  
  
{  
  
  override def toString = x  
  
}
```

270. What is Resilient Distributed Dataset(RDD)?

Resilient Distributed Dataset(RDD) is core of Apache Spark which provides primary data abstraction.

These are features of RDDs:

Resilient means fault-tolerant with the help of RDD lineage graph and so that it's easy to re-compute missing or damaged partitions due to failure of any node.

Distributed means this feature works with data residing on multiple nodes in a cluster.

Dataset means collection of partitioned data with primitive values or values of values, e.g. tuples or other objects.

271. Give a brief description of Fault tolerance in Hadoop?

Fault tolerance can be defined as, proper functioning of the system without any data loss even if some hardware components of the system fails. This feature of Hadoop is used for computing large data sets with parallel and distributed algorithms in the cluster without any failures. It uses the Heart of Hadoop i.e. MapReduce.

272. What is Immutable data with respect to Hadoop?

Immutability is the idea that data or objects cannot be modified once they are created. This concept provides the basic functionalities of the Hadoop in computing the large data without any data loss or failures. Programming languages, like Java and Python, treat strings as immutable objects which means we will not be able change it.

273. Which are the nodes that hadoop can b executed?

We have three modes in which Hadoop can run and which are:

Standalone (local) mode: Default mode of Hadoop, it uses the local file system for input and output operations. This mode is used for debugging purpose, and it does not support the use of HDFS.

Pseudo-distributed mode: In this case, you need configuration for all the three files mentioned above. In this case, all daemons are running on one node and thus, both Master and Slave nodes are on the same machine.

Fully distributed mode: This is the production phase of Hadoop where data is distributed across several nodes on a Hadoop cluster. Different nodes are allotted as Master and Slaves.

274. How is formatting done in HDFS?

Hadoop distributed file system(HDFS) is formatted using `bin/hadoop namenode -format` command. This command formats the HDFS via NameNode. This command is only used for the first time. Formatting the file system means starting the working of the directory specified by the `dfs.name.directory` variable. If you execute this command on existing filesystem, you will delete all your data stored on your NameNode. Formatting a Namenode will not format the DataNode.

275. What are the contents found in masterfile of hadoop?

The masters file contains information about Secondary NameNode server location.

276. Describe the main hdfs-site.xml properties?

The three important `hdfs-site.xml` properties are:

`checkpoint.dir` is the directory found on the filesystem where the Secondary NameNode collects the temporary images of edit logs, which is to be combined and the `FsImage` for backup.

277. Explain about spill factor with respect to the RAM?

The map output is stored in an in-memory buffer; when this buffer is almost full, then spilling phase begins in order to transport the data to a temp folder.

Map output is first written to buffer and buffer size is decided by `mapreduce.task.io.sort.mb`. By default, it will be 100 MB.

When the buffer outreaches certain threshold, it will start spilling buffer data to disk. This threshold is specified in `mapreduce.map.sort.spill.percent`.

278. Why do we require a password-less SSH in Fully Distributed environment?

We required a password-less SSH in a Fully-Distributed environment because when the cluster is live and working in Fully Distributed environment, the communication is too frequent. The DataNode and the NodeManager should be able to transport messages quickly to master server.

279. Does this requirement lead to security issues?

Hadoop cluster is an isolated cluster and generally, it has nothing to do with the internet. It has a different kind of a configuration. We doesn't worry about that kind of a security breach, like as, someone hacking through the internet, and so on. Hadoop also has a very secured way to connect to other devices to fetch and to process the built data.

280. What will happen to a NameNode, when ResourceManager is down?

When a ResourceManager is not working, it will not be functional (for submitting jobs) but NameNode will be available. So, the cluster is present if NameNode is working, even if the ResourceManager is not in a working state.

281. Tell about features of Fully Distributed mode?

This is one of the important questions as Fully Distributed mode is used in the production environment, in which we have 'n' number of machines resulting in the formation of a Hadoop cluster. Hadoop daemons work on a cluster of machines. There is one node on which Namenode is running and other nodes on which Datanodes are running. NodeManagers are placed on every DataNode and it is responsible for working of the task on every single DataNode. The work of ResourceManager is to manage all these NodeManagers. Another work of ResourceManager is to receive the processing requests and some parts of requests it passes to the corresponding NodeManagers and so on.

282. Explain about fsck?

The expansion of fsck is File System Check. Hadoop Distributed File System supports the file system check command to check for different inconsistencies. It is constructed or designed for reporting the problems with the files in HDFS, for example, missing blocks of a file or under-replicated blocks.

283. How to copy file from local hard disk to hdfs?

```
hadoop fs -copyFromLocal localfilepath hdfsfilepath
```

284. Is it possible to set the reducer to zero?

Yes, it is possible to set the number of reducers to zero in MapReduce (Hadoop).

When the number of reducers is set to zero, no reducers will be executed, and the output

of each mapper process will be stored to a separate file on HDFS.

285. map-side join / hive join?

To optimize the performance in Hive queries, we can use Map-side Join in Hive. We will use Map-Side Join when one of the tables in the join is small in size and can be loaded into primary memory.

So that join could be performed within a mapper process without using Reduce step.

286. Managed Table Vs External Table

Managed table stores the data in /user/hive/warehouse/tablename folder. And once you drop the table, along with the table schema, the data will be lost.

External table stores the data in the user specified location. And once you drop the table, only table schema will be lost. The data still available in HDFS for further use.

287. Difference between bucketing and partitioning

Bucketing – Bucketing concept is mainly used for data sampling. We can use Hive bucketing concept on Hive Managed tables / External tables. We can perform bucketing on a single column only not more than one column. The value of this single column will be distributed into number of buckets by using hash algorithm. Bucketing is an optimization technique and it improves the performance.

Partitioning – we can do partitioning with one or more columns and sub-partitioning (Partition within a Partition) is allowed. In static partitioning, we have to give the number of static partitions. But in dynamic partitioning, the number of partitions will be decided by number of unique values in the partitioned column.

288. Syntax to create hive table with partitioning

```
create table tablename
```

```
(
```

```
var1 datatype1,
```

```
var2 datatype2,
```

```
var3 datatype3
```

```
)
```

```
PARTITIONED BY (var4 datatype4,var5 datatype5)
```

ROW FORMAT DELIMITED

FIELDS TERMINATED BY 'delimiter'

LINES TERMINATED BY '\n'

TBLPROPERTIES ("SKIP.HEADER.LINE.COUNT"="1")

289. SQOOP split by:

For parallel importing / exporting the data to / from HDFS from / to RDBMS with multiple mappers. We can distribute the work load into multiple parts.

split-by used to specify the column of a table used to generate the splits for import. It means which column has to be used to create splits for imports will be declared by split-by.

Generally select min(split-by column) from table and select max(split-by column) from table will decide the out boundaries for the split (boundary-query). We need to define the column to create splits for parallel imports. Otherwise, sqoop will split the workload based on primary key of the table.

Syntax: bin/sqoop import --connect jdbc:mysql://localhost/database --table tablename --split-by column

290. file formats available in SQOOP Import

Delimited Text and sequenceFile

Delimited Text is default import file format. We can specify it as stored as-textfile

sequenceFile is binary format.

291. Default number of mappers in a sqoop command

the default number of mappers is 4 in a sqoop command.

292. Maximum number of mappers used a sqoop import command

The maximum number of mappers depends on many variables:

Database type.

Hardware that is used for your database server.

Impact to other requests that your database needs to process.

293. Flume Architecture

External data source ==> Source ==> Channel ==> Sink ==> HDFS

294. In Unix, command to show all processes

ps

295. partitions in hive

Partitions allows use to store the data in different sub-folders under main folder based on a Partitioned column.

Static Partitions: User has to load the data into static partitioned table manually.

Dynamic Partitions: We can load the data from a non-partitioned table to partitioned table using dynamic partitions.

```
set hive.exec.dynamic.partition = true
```

```
set hive.exec.dynamic.partition.mode = nonstrict
```

```
set hive.exec.dynamic.partitioned
```

```
set hive.exec.max.dynamic.partitions = 10000
```

```
set hive.exec.max.dynamic.partitions.pernode = 1000
```

296. File formats in hive

ORC File format – Optimized Row Columnar file format

RC File format – Row Columnar file format

TEXT File format – Defalut file format

Sequence file format – If the size of a file is smaller than the data block size in Hadoop, we can consider it as a small file. Due to this, metadata increases which will become an overhead to the NameNode. To solve this problem, sequence files are introduced. Sequence files act as containers to store multiple small files.

Avro file format

Custom INPUT FILE FORMAT and OUTPUT FILE FORMAT

297. Syntax to create bucketed table

```
create table tablename  
(  
  var1 datatype1,  
  var2 datatype2,  
  var3 datatype3  
)  
PARTITIONED BY (var4 datatype4,var5 datatype5)  
CLUSTERED BY (VAR1) INTO 5 BUCKETS  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY 'delimiter'  
LINES TERMINATED BY '\n'  
TBLPROPERTIES ("SKIP.HEADER.LINE.COUNT"="1")
```

298. Custom Partitioning

Custom Partitioner is a process that allows us to store the results in different reducers, based on the user condition. By setting a partitioner to partition by the key, we can confirm that, records for the same keys will go to the same reducers.

299. Difference between order by and sort by

Hive supports sortby – sort the data per reducer and orderby – sort the data for all reducers (mean sort the total data)

300. Purpose of Zoo Keeper

Zoo Keeper assists in cluster management.

Manage configuration across nodes: Hadoop cluster will have hundreds of systems. Zoo Keeper helps in synchronization of configurations across the cluster.

As many systems are involved, race condition and deadlocks are common problems when implementing distributed applications.

Race condition occurs when a system tries to perform two or more operations at the same time and this can be taken care by serialization property of ZooKeeper.

Deadlock is when two or more systems try to access same shared resource at the same time. Synchronization helps to solve the deadlock.

Partial failure of process, which can lead to uncertainty of data. Zookeeper handles this through atomicity, which means either whole of the process will finish or nothing will carry through after failure.

301. Sqoop Incremental last modified

```
bin/sqoop  import  -connect  jdbc:mysql://localhost/database  -table  
table_name -incremental-lastmodified -check-column column_name -last-  
value 'value' -m 1
```

302. Difference MR1 vs MR2

MR1 – It consists of Job Tracker and Task Tracker (For processing) and name node and data node (For storing). It supports only MR framework.

MR2 – Job Tracker has been splitted again into two parts application master (one per mr job) and resource manager (only one). It will support MR framework and other frameworks too (spark, storm)

303. select * from table – give what results for normal table and partitioned table?

give the same results in both the scenarios.

304. Explode and implode in hive

Explode – will explore the array of values into the individual values.

Syntax – select pageid, adid from page LATERAL VIEW explode (adid_list)
mytable as adid;

implode – collect aggregates records into either an array or map from multiple rows. It is the opposite of an explode().

syntax – `select userid, collect(actor_id) from actor group by userid;`

305. Interceptors in Flume:

Interceptors are designed to modify or drop an event of data. Flume is designed to pick the data from source and drop it into Sink.

Timestamp Interceptors: This will add the timestamp at which process is running to the header event.

Host Interceptors: this will write the hostname or ip address of the host system on which the agent or process is running to the event of data.

Static Interceptors: This will add the static string along with the static header to all events;

UUID Interceptors: Universal Unique Identifier, this setups a UUID on all events that are intercepted.

Search and Replace Interceptors: this will search and replace a string with a value in the event data.

Regex filtering Interceptors: This is used to include/exclude an event. This filters events selectively by interpreting a event body as text and against a matching text against a configured regular expression.

Regex Extractor Interceptors: this will extracts a match of regex interceptors against a regular expression.

306. Different types of distributed file systems:

HDFS – Hadoop Distributed File system

GFS – Google File System

MapR File system

Ceph File system

IBM General Parallel file system (GPFS)

307. Write a pig script to extract hive table?

First we need to enter the pig shell with option `useHCatalog` (`pig -useHCatalog`).

```
A = LOAD 'tablename' USING org.apache.hive.hcatalog.pig.HCatLoader();
```

```
A = LOAD 'airline.airdata' USING org.apache.hive.hcatalog.pig.HCatLoader();
```

308. Predefined value in sqoop to extract data from any database current date minus one?

```
sqoop import --connect jdbc:mysql://localhost/database --table table_name --  
where "time_stamp > day(now()-1)"
```

309. UNION, UNIONALL, MINUS and INTERSECT available in hive?

```
select_statement UNION [ALL | DISTINCT] select_statement
```

MINUS keyword is not available in Hive

INTERSECT keyword is not available in Hive

310. Difference between Distribute by, cluster by, order by, sort by

Distribute by – Distribute the data among n reducers (un-sorted manner).

Cluster by – Distribute the data among n reducers and sort the data (Distribute by and sort by).

order by – sort the data for all reducers.

sort by – sort the data per reducer.

311. Describe the main hdfs-site.xml properties? The three important hdfs-site.xml properties are:

dfs.name.dir which gives you the location in which NameNode stores the metadata (FsImage and edit logs) and where DFS is located – on the disk or onto the remote directory.

Location of the DataNodes is given by dfs.data.dir , and the data is stored in DataNodes.

fs.checkpoint.dir is the directory found on the filesystem where the Secondary NameNode collects the temporary images of edit logs, which is to be combined and the FsImage for backup.

312. What is Hadoop? Name the Main Parts of a Hadoop Application

Hadoop is what developed as the solution to the Big Data problem. Hadoop is described as the structure that offers a product of tools and services in order to collect and prepare Big Data. It also plays a relevant role in the analysis of big data and to make effective business choices when it is difficult to make the decision using the conventional method. Hadoop offers a vast toolset that makes it reasonable to store and prepare data very quickly. Here are all the main components of the Hadoop:

1. HDFS
2. Hadoop MapReduce
3. YARN
4. PIG and HIVE – The Data Access Components.
5. HBase – For Data Storage
6. Apache Flume, Sqoop, Chukwa – The Data Integration Components
7. Ambari, Oozie and ZooKeeper – Data Management and Monitoring Component
8. Thrift and Avro – Data Serialization components
9. Apache Mahout and Drill – Data Intelligence Components

313. How many Data Formats are there in Hadoop?

1. Text Input Format: The text input is the failure input format in Hadoop.
2. Sequence File Input Format: This input format is used to read files in order.
3. Key Value Input Format: This input format is used for clear text files.

314. What do you know about YARN?

YARN stands for Yet Another Resource Negotiator, it is the Hadoop processing structure. YARN is capable to manage the support and establish execution conditions for the processes.

315. Why do nodes are extracted and added regularly in Hadoop cluster?

1. The Hadoop framework uses materials hardware, and it is one of the great features of the Hadoop framework. It appears in a common DataNode crash in a Hadoop cluster.
2. The ease of scale is a yet different primary feature of the Hadoop framework that is implemented according to the rapid increase in data volume.

316. What do you understand by “Rack Awareness”?

In Hadoop, Rack Awareness is defined as the algorithm into which NameNode manages how the blocks and their models are stored in the Hadoop cluster. This is done via rack Sensitivity: Internal outlines that minimize the traffic inside DataNodes within the same rack. Let's take an example – we know that the failure value of replication factor is 3. According to the “Replica Placement Policy” two images of models for every block of data will be collected in a single rack whereas the three copy is stored in another rack.

317. What do you understand about the Speculative Execution?

In Hadoop, Speculative Execution is a means that takes place through the slower performance of a task at a node. In this manner, the master node starts performing another example of that same task on the different node. And the task which is completed first is taken and the execution of other is stopped by killing that.

318. State any of the main features of Hadoop

1. Hadoop framework is created on Google MapReduce that is based on Google's Big Data File Systems.
2. Hadoop framework can explain several questions efficiently for Big Data analysis.

319. Do you know some organizations that are using Hadoop?

Yahoo – working Hadoop Facebook – produced Hive for analysis Amazon, Adobe, Spotify, Netflix, eBay, and Twitter are any other well-known and lived companies that are doing Hadoop.

320. How can you distinguish RDBMS RDBMS and Hadoop?

1. RDBMS is arranged to store structured data, whereas Hadoop can put any kind of data i.e. unstructured, structured, or semi-structured.

2. RDBMS supports “Schema on write” method while Hadoop is based on “Schema on reading” policy.

3. The schema of data is previously known in RDBMS that performs Reads fast, whereas in HDFS, records no schema validation issues during HDFS write, so the Writes are fast.

4. RDBMS is licensed software, so one wants to pay for it, whereas Hadoop does open source software, so it is free of cost. 5. RDBMS is used for Online Transactional Processing (OLTP) method whereas Hadoop is used for data analytics, data discovery, and OLAP system as well.

321. What do you know about active and passive NameNodes

Active NameNode – The NameNode that moves in the Hadoop cluster, is the Dynamic NameNode. Passive NameNode – The standby NameNode that stores the corresponding data as that of the Active NameNode is the Passive NameNode. On the breakdown of active NameNode, the passive NameNode restores it and takes the charge. In this way, there is ever a running NameNode in the cluster and thus it nevermore fails

322. What are the Parts from Apache HBase?

1. Region Server: A Report can be split into several regions. A group of these areas gets completed to the clients by a Region Server.

2. H Master: This organizes and operates the Region server.

3. Zoo Keeper: This acts as an organizer inside HBase distributed context. It functions by having a server state inside of this cluster by communication in sessions.

323. How is the DataNode failure managed by NameNode?

Name Node continuously accepts a signal from all the DataNodes started in Hadoop cluster that defines the proper function of the DataNode. The record of all the pieces present on a DataNode is stored in a block report. If a DataNode is disappointed in sending the signal to the NameNode, it is marked Sensitivity: Internal decedent later a particular time period. Then the NameNode replicates/copies the blocks of the final node over different DataNode with the earlier built replicas.

324. Define the NameNode recovery process.

The process of NameNode restoration helps to prevent the Hadoop cluster running, and can be defined by the following steps –

Step 1: To start a new NameNode, appropriate the file system metadata model (FsImage).

Step 2: Configure the clients and Data Nodes to support the new NameNode.

Step 3: Once the new Name performs the loading of last checkpoint FsImage and takes a block store from the DataNodes, the new NameNode start assisting the client.

325. What are the various programs available in Hadoop?

The differently available schedulers in Hadoop are – COSHH – It lists resolutions by analyzing cluster, workload, and managing heterogeneity. FIFO Scheduler – It orders the jobs on the base of their approach time in a line without using heterogeneity. Fair Sharing – It defines a supply for each user that includes a representation of pictures and defeat slots on a resource. Each user is granted to use own pool for the performance of jobs.

326. Can DataNode and NameNode be specialty hardware?

DataNodes are the specialty hardware only as it can store data like laptops and individual computers, these are needed in high numbers. Instead, NameNode is the master node; it stores metadata about all the blocks saved in HDFS. It needs large memory space, thus works as a high-end the device with great memory space.

327. Whatever is that Hadoop daemon? Explain their roles.

NameNode – The master node, the subject for metadata warehouse for all lists and files is known as the NameNode. It also contains metadata data about each block of the data and their allocation in Hadoop cluster.

Secondary NameNode – This daemon is capable to merge and store the changed Filesystem Image into stable storage. It is used in case the NameNode fails.

DataNode – The slave node including actual data is the DataNode.

NodeManager – Working on the slave devices, the NodeManager controls the launch of the application container, controlling resource usage and reporting same to the ResourceManager.

ResourceManager – It is the main administration qualified to manage devices and to schedule forms running on the top of YARN.

JobHistoryServer – It is responsible to keep every information about the MapReduce jobs when the Application Master stops to work (terminates).

328. Define “Checkpointing”. What is its benefit?

Checkpointing is a system that compacts a FsImage and Edits record into a new FsImage. In this way, the NameNode manages the loading of the final in-memory time from the FsImage immediately, instead of replaying an edit log. The secondary NameNode is qualified to perform the checkpointing process. Benefit of Checkpointing Checkpointing is an extremely efficient process and reduces the startup time of the NameNode.

329. Name the methods in which Hadoop code can be run.

1. Fully-distributed mode
2. Pseudo-distributed mode
3. Standalone mode

330. What is the Hadoop's map reduction?

To implement a large data set parallel across a Hadoop cluster, the Hadoop Map Reduce architecture is used. Data analysis uses two step diagrams and reduces the processes.

331. How does the Hadoop map work?

In Map Reduce, during the map stage, it counts the words in each document, while combining data according to the document that contains the entire collection at the cutting point. During the map stage, the input data is divided into divisions to analyze graphic tasks in parallel to the hood structure.

332. Explain what happens in MapReduce

The system operates in any way and modifies graphic outputs for reuse, entries called shuffle

333. Explain the Cache distributed in MapReduce structure?

Distributed Cache is an important feature provided by MapReduce architecture. Whenever you want to share some files on the hooedo cluster, the distributed cache is used. Files can be an executable jar files or simple properties file.

334. What is JobTracker in Hupa? What does hatio continue to do?

Job Tracker is used in Hadoop to submit and monitor Map Reduce jobs. The work tracker runs on its own JVM process

Work performs the following activities in the tracker's hoard

Client application must submit work to work supervisor

Contacting Job Tracker Name Mode to determine the data location

The Job Tracker Task Tracker edges near the location or available locations are available

In the selected Task Tracker nodes, it submits to the job

If a work fails, the worker will announce and decide what to do.

Task Tracker edges are tracked by Job Tracker,

335. Explain what is heart rate in HDFS?

If the name tone or work supervisor is not answered with the signal, the heart rate is indicated for a signal used at a data endpoint and a name signal between the work monitor and the work monitor

336. Explain what the connectors do, and explain that you should use a partner in the mopredos work.

Connectors are used to increase the efficiency of Map Reduce. The amount of data can be reduced and can be reduced with the help of the connector. If surgery is transmitted and subtle, you can use your reproduction code as a collaborator. The execution of the coupator was not guaranteed.

337. What happens when a data node fails?

If a data node fails

Find Job tracker and name node failure

All the tasks in the failure end are reconstructed

Name node represents the user's data with another terminal

338. What is Special Execution?

During specific execution in hugo, some specific tasks start. In another slave terminal, you can reduce multiple copies or tasks of the same map. In simpler terms, if a particular driver takes a long time to complete a task, hoodo will create a duplicate task on another disk. A disc of the first work was retained and the first unreleased disks were killed.

339. What are the basic parameters of the modeler?

Basic parameters of the mapper type

Long and text

Text and Intwritable

340. What is the MapReduce partition function?

The function of the MapReduce partition is to ensure that all values of the same keys go to the same reader, which ultimately helps distribute the map output on the manufacturer.

341. What is the difference between input separation and HDFS block?

The logical segment of data is called split when the project section of the data is called HDFS Block

342. What happens in text form?

In the text input format, each line in the text file is a log. When the line is in the interior of the line, the content of the line is worth. For example, main: longwritable, value: text

343. Do you specify the main configuration parameters that the user should be prompted to work with?

The MapReduce Framework user must specify

Job's input spaces in the distributed file system

Work release location in the distributed file system

Input format

Output format

Class with map functionality

Class with squeeze function

JAP file mappers, driver and driver classes.

344. What would you explain to WebDAV at Hadoop?

WebDAV is a set of extensions for HTTP to support editing and updating the files. In most operating systems WebDAV shares can be loaded into file systems, so you can access HDFS as a standard file system by introducing HDFS to WebDAV.

345. Explain how JobTracker should schedule a task?

Sends Heartbeat messages as a drop-down driver that works every few minutes to ensure that JobTracker is active and active. The JobTracker also reports about the available availability of locations, so the JobTracker cluster is up to the date the task is given.

346. Explain what is Sequencefile?

Sequencefileinputformat is used to read files in queues. This is a specific compressed binary file format, which is optimized to obtain data between inputs for some MapReduce workspaces between a MapReduce task release.

347. What is Hadoop?

It is an open source software architecture for data and running applications as a hardware cluster of products. It offers massive processing and massive storage for any type of data.

348. What is the difference between RDBMS and Hadoop?

RDBMS vs Hadoop

RDBMS is a related database management system Hadoop is a node based flat system

It was used for OLTP processing, whereas Hadoop is currently used for analysis and BIG DATA processing.

In RDBMS, the database package uses the same data files stored in shared storage on the hi-tech, and storage data is stored personally on each processing point.

You need to prepare the data to prepare it before you save it.

349. Are you familiar with the Hadoop core elements?

Hadoop includes key elements,

HDFS

Map Reduce

350. What is the name of the master?

In Hadoop, name node is also stored in HDFS to store all file location information in HDFS. It is Master Node, which works with the Tracker and contains Metadata.

351. What are the data elements used by Hadoop?

There are data elements that Hadoop uses

Pig

Hive

352. How to Write a Custom Partition for a Hadoop Work?

You write a custom partition for a hobby work, and you follow the following path

Create a new class that extends the sharing class

Get the method

MapReduce running guitar

Add Custom Partition to Work by Using Partition Partitioner Clause – Add a Custom Share to Work as a Configuration File.

353. What will happen to the worker?

Name node is a single point for failure in HDFS, so your cluster will be turned off when Name node is down.

354. How is it being indexed in HDFS?

Hadoop has a unique way table. Once the volume level is collected, the next part of the HDFS data will store the last section where it says.

355. How is RDBMS different from HDFS?

RDBMS	HDFS
Based on structured data.	Any type of data can be used.
Limited processing capacity.	Processing in parallel manner.
Schema on write.	Schema on read.
Read is faster than write.	Write is faster than read.
Licensed	Open source

356. Explain a checkpoint?

Checkpointing is an approach that works by taking an FsImage with edit log then compact Given below, into a FsImage. Replaying of edit log is not required. NameNode loads the end in-memory state from FsImage. It proves out to be an efficient operation. It also reduces startup time of NameNode. Secondary NameNode is used for performing Checkpointing.

357. Explain Apache HBase?

It is an open source, distributed, multidimensional, scalable, NoSQL database. It is written in Java. It provides BigTable like capabilities, fault-tolerant ways, high throughput, etc.

358. Explain Apache Spark?

It is a framework used in real-time analytics of data in distributed computing. It runs in-memory computations and increases the data processing speed.

359. Explain “SerDe”?

The “SerDe” allows instruction of “Hive” about processing of a record. It is a combined form of “Serializer” with “Deserializer”.

360. Explain Apache ZooKeeper?

It coordinates with multiple services of distributed environment. A lot of time is saved by synchronization, grouping, naming, configuration maintenance etc.

361. What is RDD.

It stands for Resilient Distribution Datasets.

It is a set of operational elements running in parallel.

362. Explain Rack Awareness?

This is an algorithm where “NameNode” decides the placement of blocks with their replicas. This is based on definitions of racks so that network traffic can be minimized between “DataNodes” in the same rack.

363. What platform and Java version are required to run Hadoop?

Java 1.6.x or higher versions are good for Hadoop, preferably from Sun. Linux and Windows are the supported operating system for Hadoop, but BSD, Mac OS/X, and Solaris are more famous for working.

364. What kind of Hardware is best for Hadoop?

Hadoop can run on a dual processor/ dual core machines with 4-8 GB RAM using ECC memory. It depends on the workflow needs.

365. What are the most common input formats defined in Hadoop?

These are the most common input formats defined in Hadoop:

1. TextInputFormat
2. KeyValueInputFormat
3. SequenceFileInputFormat

TextInputFormat is a by default input format.

366. Explain the use of .media class?

For the floating of media objects from one side to another, we use this class.

367. Give the use of the bootstrap panel.

We use panels in bootstrap from the boxing of DOM components.

368. What is the purpose of button groups?

Button groups are used for the placement of more than one buttons in the same line.

369. Name the various types of lists supported by Bootstrap.

- Ordered list
- Unordered list
- Definition list

370. Which command is used for the retrieval of the status of daemons running the Hadoop cluster?

The 'jps' command is used for the retrieval of the status of daemons running the Hadoop cluster.

371. What is InputSplit in Hadoop? Explain.

When a Hadoop job runs, it splits input files into chunks and assigns each split to a mapper for processing. It is called the InputSplit.

372. What is TextInputFormat?

In TextInputFormat, each line in the text file is a record. Value is the content of the line while Key is the byte offset of the line. For instance, Key: longWritable, Value: text

373. What is the SequenceFileInputFormat in Hadoop?

In Hadoop, SequenceFileInputFormat is used to read files in sequence. It is a specific compressed binary file format which passes data between the output of one MapReduce job to the input of some other MapReduce job.

374. How many InputSplits is made by a Hadoop Framework?

Hadoop makes 5 splits as follows:

- One split for 64K files
- Two splits for 65MB files, and
- Two splits for 127MB files

375. What is the use of RecordReader in Hadoop?

InputSplit is assigned with a work but doesn't know how to access it. The record holder class is totally responsible for loading the data from its source and convert it into keys pair suitable for reading by the Mapper. The RecordReader's instance can be defined by the Input Format.

376. What is JobTracker in Hadoop?

JobTracker is a service within Hadoop which runs MapReduce jobs on the cluster.

377. What is WebDAV in Hadoop?

WebDAV is a set of extension to HTTP which is used to support editing and uploading files. On most operating system WebDAV shares can be mounted as filesystems, so it is possible to access HDFS as a standard filesystem by exposing HDFS over WebDAV.

378. What is Sqoop in Hadoop?

Sqoop is a tool used to transfer data between the Relational Database Management System (RDBMS) and Hadoop HDFS. By using Sqoop, you can transfer data from RDBMS like MySQL or Oracle into HDFS as well as exporting data from HDFS file to RDBMS.

379. What are the functionalities of JobTracker?

These are the main tasks of JobTracker:

- To accept jobs from the client.
- To communicate with the NameNode to determine the location of the data.
- To locate TaskTracker Nodes with available slots.

- To submit the work to the chosen TaskTracker node and monitors the progress of each task.

380. Define TaskTracker.

TaskTracker is a node in the cluster that accepts tasks like MapReduce and Shuffle operations from a JobTracker.

381. What is Map/Reduce job in Hadoop?

Map/Reduce job is a programming paradigm which is used to allow massive scalability across the thousands of server.

MapReduce refers to two different and distinct tasks that Hadoop performs. In the first step maps jobs which takes the set of data and converts it into another set of data and in the second step, Reduce job. It takes the output from the map as input and compresses those data tuples into the smaller set of tuples.

382. What is "map" and what is "reducer" in Hadoop?

Map: In Hadoop, a map is a phase in HDFS query solving. A map reads data from an input location and outputs a key-value pair according to the input type.

Reducer: In Hadoop, a reducer collects the output generated by the mapper, processes it, and creates a final output of its own.

383. What is shuffling in MapReduce?

Shuffling is a process which is used to perform the sorting and transfer the map outputs to the reducer as input.

384. What is NameNode in Hadoop?

NameNode is a node, where Hadoop stores all the file location information in HDFS (Hadoop Distributed File System). We can say that NameNode is the centerpiece of an HDFS file system which is responsible for keeping the record of all the files in the file system, and tracks the file data across the cluster or multiple machines.

385. What is heartbeat in HDFS?

Heartbeat is a signal which is used between a data node and name node, and between task tracker and job tracker. If the name node or job tracker doesn't respond to the signal then it is considered that there is some issue with data node or task tracker.

386. How is indexing done in HDFS?

There is a very unique way of indexing in Hadoop. Once the data is stored as per the block size, the HDFS will keep on storing the last part of the data which specifies the location of the next part of the data.

387. What happens when a data node fails?

If a data node fails the job tracker and name node will detect the failure. After that, all tasks are re-scheduled on the failed node and then name node will replicate the user data to another node.

388. What is Hadoop Streaming?

Hadoop streaming is a utility which allows you to create and run map/reduce job. It is a generic API that allows programs written in any languages to be used as Hadoop mapper.

389. What is a combiner in Hadoop?

A Combiner is a mini-reduce process which operates only on data generated by a Mapper. When Mapper emits the data, combiner receives it as input and sends the output to a reducer.

390. What are the Hadoop's three configuration files?

Following are the three configuration files in Hadoop:

- core-site.xml
- mapred-site.xml
- hdfs-site.xml

391. What are the network requirements for using Hadoop?

Following are the network requirement for using Hadoop:

- Password-less SSH connection.
- Secure Shell (SSH) for launching server processes.

392. What do you know by storage and compute node?

Storage node: Storage Node is the machine or computer where your file system resides to store the processing data.

Compute Node: Compute Node is a machine or computer where your actual business logic will be executed.

393. Is it necessary to know Java to learn Hadoop?

If you have a background in any programming language like C, C++, PHP, Python, Java, etc. It may be really helpful, but if you are nil in java, it is necessary to learn Java and also get the basic knowledge of SQL.

394. How to debug Hadoop code?

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

- By using Counters.
- By web interface provided by the Hadoop framework.

395. Is it possible to provide multiple inputs to Hadoop? If yes, explain.

Yes, It is possible. The input format class provides methods to insert multiple directories as input to a Hadoop job.

396. What is the relation between job and task in Hadoop?

In Hadoop, A job is divided into multiple small parts known as the task.

397. What is the difference between Input Split and HDFS Block?

The Logical division of data is called Input Split and physical division of data is called HDFS Block.

398. What is the difference between HDFS and NAS?

HDFS data blocks are distributed across local drives of all machines in a cluster whereas, NAS data is stored on dedicated hardware.

399. What is the difference between Hadoop and other data processing tools?

Hadoop facilitates you to increase or decrease the number of mappers without worrying about the volume of data to be processed.

400. What is distributed cache in Hadoop?

Distributed cache is a facility provided by MapReduce Framework. It is provided to cache files (text, archives etc.) at the time of execution of the job. The Framework copies the necessary files to the slave node before the execution of any task at that node.

401. What commands are used to see all jobs running in the Hadoop cluster and kill a job in LINUX?

Hadoop job - list

Hadoop job - kill jobID

402. What is the functionality of JobTracker in Hadoop? How many instances of a JobTracker run on Hadoop cluster?

JobTracker is a giant service which is used to submit and track MapReduce jobs in Hadoop. Only one JobTracker process runs on any Hadoop cluster. JobTracker runs it within its own JVM process.

Functionalities of JobTracker in Hadoop:

- When client application submits jobs to the JobTracker, the JobTracker talks to the NameNode to find the location of the data.
- It locates TaskTracker nodes with available slots for data.
- It assigns the work to the chosen TaskTracker nodes.
- The TaskTracker nodes are responsible to notify the JobTracker when a task fails and then JobTracker decides what to do then. It may resubmit the task on another node or it may mark that task to avoid.

403. How JobTracker assign tasks to the TaskTracker?

The TaskTracker periodically sends heartbeat messages to the JobTracker to assure that it is alive. This messages also inform the JobTracker about the number of available slots. This return message updates JobTracker to know about where to schedule task.

404. Is it necessary to write jobs for Hadoop in the Java language?

No, There are many ways to deal with non-java codes. HadoopStreaming allows any shell command to be used as a map or reduce function.

405. Which data storage components are used by Hadoop?

HBase data storage component is used by Hadoop.

406. What Is Compute And Storage Nodes?

Compute Node: This is the computer or machine where your actual business logic will be executed.

Storage Node: This is the computer or machine where your file system reside to store the processing data.

In most of the cases compute node and storage node would be the same machine.

407. How Does Master Slave Architecture In The Hadoop?

The MapReduce framework consists of a single master JobTracker and multiple slaves, each cluster-node will have one TaskTracker. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

408. How Does An Hadoop Application Look Like Or Their Basic Components?

Minimally an Hadoop application would have following components.

- Input location of data
- Output location of processed data.
- A map task.
- A reduced task.
- Job configuration
- The Hadoop job client then submits the job (jar/executable etc.) and configuration to the JobTracker which then assumes the responsibility of distributing the software / configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

409. Explain How Input And Output Data Format Of The Hadoop Framework?

The MapReduce framework operates exclusively on pairs, that is, the framework views the input to the job as a set of pairs and produces a set of pairs as the output of the job, conceivably of different types. See the flow mentioned below
(input) -> map -> -> combine/sorting -> -> reduce -> (output)

410. What Are The Restriction To The Key And Value Class ?

The key and value classes have to be serialized by the framework. To make them serializable Hadoop provides a Writable interface. As you know from the java itself that the key of the Map should be comparable, hence the key has to implement one more interface Writable Comparable.

411. Explain The Wordcount Implementation Via Hadoop Framework ?

We will count the words in all the input file flow as below

- input Assume there are two files each having a sentence
Hello World Hello World (In file 1) Hello World Hello World
(In file 2)
- Mapper : There would be each mapper for the a file For the given sample input the first map output:

- < Hello, 1>
- < World, 1>
- < Hello, 1>
- < World, 1>

The second map output:

- < Hello, 1>
- < World, 1>
- < Hello, 1>

< World, 1>Combiner/Sorting (This is done for each individual map) So output looks like this The output of the first map:

- < Hello, 2>
- < World, 2>

The output of the second map:

- < Hello, 2>

< World, 2>Reducer : It sums up the above output and generates the output as below

- < Hello, 4>
- < World, 4>

Output

Final output would look like
 Hello 4 times
 World 4 times

412. Which Interface Needs To Be Implemented To Create Mapper And Reducer For The Hadoop?

org.apache.hadoop.mapreduce.Mapper

org.apache.hadoop.mapreduce.Reducer

413. What Is The Inputsplit In Map Reduce Software?

An InputSplit is a logical representation of a unit (A chunk) of input work for a map task; e.g., a file name and a byte range within that file to process or a row set in a text file.

414. What Is The Inputformat ?

The InputFormat is responsible for enumerate (itemise) the InputSplits, and producing a RecordReader which will turn those logical work units into actual physical input records.

415. Where Do You Specify The Mapper Implementation?

Generally mapper implementation is specified in the Job itself.

416. How Mapper Is Instantiated In A Running Job?

The Mapper itself is instantiated in the running job, and will be passed a MapContext object which it can use to configure itself.

417. Which Are The Methods In The Mapper Interface?

The Mapper contains the run() method, which call its own setup() method only once, it also call a map() method for each input and finally calls it cleanup() method. All above methods you can override in your code.

418. What Happens If You Don't Override The Mapper Methods And Keep Them As It Is?

If you do not override any methods (leaving even map as-is), it will act as the identity function, emitting each input record as a separate output.

419. What Is The Use Of Context Object?

The Context object allows the mapper to interact with the rest of the Hadoop system. It Includes configuration data for the job, as well as interfaces which allow it to emit output.

420. How Can You Add The Arbitrary Key-value Pairs In Your Mapper?

You can set arbitrary (key, value) pairs of configuration data in your Job, e.g. with
Job.getConfiguration().set("myKey", "myVal"), and then retrieve this data in your mapper with

Context.getConfiguration().get("myKey"). This kind of functionality is typically done in the Mapper's setup() method.

421. How Does Mapper's Run() Method Works?

The Mapper.run() method then calls map(KeyInType, ValInType, Context) for each key/value pair in the InputSplit for that task

422. Which Object Can Be Used To Get The Progress Of A Particular Job ?

Context

423. What Is Next Step After Mapper Or Maptask?

The output of the Mapper are sorted and Partitions will be created for the output. Number of partition depends on the number of reducer.

424. How Can We Control Particular Key Should Go In A Specific Reducer?

Users can control which keys (and hence records) go to which Reducer by implementing a custom Partitioned.

425. What Is The Use Of Combiner?

It is an optional component or class, and can be specify via Job.setCombinerClass(ClassName), to perform local aggregation of the intermediate outputs, which helps to cut down the amount of data transferred from the Mapper to the Reducer.

426. How Many Maps Are There In A Particular Job?

The number of maps is usually driven by the total size of the inputs, that is, the total number of blocks of the input files. Generally it is around 10-100 maps per-node. Task setup takes awhile, so it is best if the maps take at least a minute to execute. Suppose, if you expect 10TB of input data and have a block size of 128MB, you'll end up with 82,000 maps, to control the number of block you can use

the `mapreduce.job.maps` parameter (which only provides a hint to the framework). Ultimately, the number of tasks is controlled by the number of splits returned by the `InputFormat.getSplits()` method (which you can override).

427. What Is The Reducer Used For?

Reducer reduces a set of intermediate values which share a key to a (usually smaller) set of values. The number of reduces for the job is set by the user via `Job.setNumReduceTasks(int)`.

428. Explain The Core Methods Of The Reducer?

The API of Reducer is very similar to that of Mapper, there's a `run()` method that receives a `Context` containing the job's configuration as well as interfacing methods that return data from the reducer itself back to the framework. The `run()` method calls `setup()` once, `reduce()` once for each key associated with the reduce task, and `cleanup()` once at the end. Each of these methods can access the job's configuration data by using `Context.getConfiguration()`.

As in Mapper, any or all of these methods can be overridden with custom implementations. If none of these methods are overridden, the default reducer operation is the identity function; values are passed through without further processing.

The heart of Reducer is its `reduce()` method. This is called once per key; the second argument is an `Iterable` which returns all the values associated with that key.

429. What Are The Primary Phases Of The Reducer?

Shuffle, Sort and Reduce.

430. Explain The Shuffle?

Input to the Reducer is the sorted output of the mappers. In this phase the

framework fetches the relevant partition of the output of all the mappers, via HTTP.

431.Explain The Reducer's Sort Phase?

The framework groups Reducer inputs by keys (since different mappers may have output the same key) in this stage. The shuffle and sort phases occur simultaneously; while map-outputs are being fetched they are merged (It is similar to merge-sort).

432. Explain The Reducer's Reduce Phase?

In this phase the `reduce(MapOutKeyType, Iterable, Context)` method is called for each pair in the grouped inputs. The output of the reduce task is typically written to the FileSystem via `Context.write (ReduceOutKeyType, ReduceOutValType)`. Applications can use the Context to report progress, set application-level status messages and update Counters, or just indicate that they are alive. The output of the Reducer is not sorted.

433. How Many Reducers Should Be Configured?

The right number of reduces seems to be 0.95 or 1.75 multiplied by (`<no.of nodes> * mapreduce.tasktracker.reduce.tasks.maximum`). With 0.95 all of the reduces can launch immediately and start transferring map outputs as the maps finish. With 1.75 the faster nodes will finish their first round of reduces and launch a second wave of reduces doing a much better job of load balancing. Increasing the number of reduces increases the framework overhead, but increases load balancing and lowers the cost of failures.

434. It Can Be Possible That A Job Has 0 Reducers?

It is legal to set the number of reduce-tasks to zero if no reduction is desired.

435. What Happens If Number Of Reducers Are 0?

In this case the outputs of the map-tasks go directly to the FileSystem, into

the output path set by `setOutputPath(Path)`. The framework does not sort the map-outputs before writing them out to the `FileSystem`.

436. How Many Instances Of Jobtracker Can Run On A Hadoop Cluster?

Only one

437. What Is The Jobtracker And What It Performs In A Hadoop Cluster?

JobTracker is a daemon service which submits and tracks the MapReduce tasks to the Hadoop cluster. It runs its own JVM process. And usually it runs on a separate machine, and each slave node is configured with job tracker node location. The JobTracker is single point of failure for the Hadoop MapReduce service. If it goes down, all running jobs are halted. JobTracker in Hadoop performs following actions

- Client applications submit jobs to the Job tracker.
- The JobTracker talks to the NameNode to determine the location of the data
- The JobTracker locates TaskTracker nodes with available slots at or near the data
- The JobTracker submits the work to the chosen TaskTracker nodes.
- A TaskTracker will notify the JobTracker when a task fails. The JobTracker decides what to do then: it may resubmit the job elsewhere, it may mark that specific record as something to avoid, and it may even blacklist the TaskTracker as unreliable.
- When the work is completed, the JobTracker updates its status.
- The TaskTracker nodes are monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different TaskTracker.
- A TaskTracker will notify the JobTracker when a task fails. The JobTracker decides what to do then: it may resubmit the job

elsewhere, it may mark that specific record as something to avoid, and it may even blacklist the TaskTracker as unreliable.

- When the work is completed, the JobTracker updates its status.
- Client applications can poll the JobTracker for information.

438. How A Task Is Scheduled By A Jobtracker?

The TaskTrackers send out heartbeat messages to the JobTracker, usually every few minutes, to reassure the JobTracker that it is still alive. These messages also inform the JobTracker of the number of available slots, so the JobTracker can stay up to date with where in the cluster work can be delegated. When the JobTracker tries to find somewhere to schedule a task within the MapReduce operations, it first looks for an empty slot on the same server that hosts the DataNode containing the data, and if not, it looks for an empty slot on a machine in the same rack.

439. How Many Instances Of Tasktracker Run On A Hadoop Cluster?

There is one Daemon Tasktracker process for each slave node in the Hadoop cluster.

440. What Are The Two Main Parts Of The Hadoop Framework?

Hadoop consists of two main parts.

- Hadoop distributed file system, a distributed file system with high throughput,
- Hadoop MapReduce, a software framework for processing large data sets.

441. Explain The Use Of Tasktracker In The Hadoop Cluster?

A Tasktracker is a slave node in the cluster which that accepts the tasks from JobTracker like Map, Reduce or shuffle operation. Tasktracker also runs in its own JVM Process.

Every TaskTracker is configured with a set of slots; these indicate the number of tasks that it can accept. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the task tracker. The Tasktracker monitors these task instances, capturing the output and exit codes. When the Task instances finish, successfully or not, the task tracker notifies the JobTracker. The TaskTrackers also send out heartbeat messages to the JobTracker, usually every few minutes, to reassure the JobTracker that it is still alive. These messages also inform the JobTracker of the number of available slots, so the JobTracker can stay up to date with where in the cluster work can be delegated.

442. What Do You Mean By Taskinstance?

Task instances are the actual MapReduce jobs which run on each slave node. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the entire task tracker. Each Task Instance runs on its own JVM process. There can be multiple processes of task instance running on a slave node. This is based on the number of slots configured on task tracker. By default a new task instance JVM process is spawned for a task.

443. How Many Daemon Processes Run On A Hadoop Cluster?

Hadoop is comprised of five separate daemons. Each of these daemons runs in its own JVM.

Following 3 Daemons run on Master nodes.

NameNode : This daemon stores and maintains the metadata for HDFS.

Secondary NameNode : Performs housekeeping functions for the NameNode.

JobTracker : Manages MapReduce jobs, distributes individual tasks to machines running the Task Tracker. Following 2 Daemons run on each Slave nodes

DataNode : Stores actual HDFS data blocks.

TaskTracker : It is Responsible for instantiating and monitoring individual Map and Reduce tasks.

444. How Many Maximum Jvm Can Run On A Slave Node?

One or Multiple instances of Task Instance can run on each slave node. Each task instance is run as a separate JVM process. The number of Task instances can be controlled by configuration. Typically a high end machine is configured to run more task instances.

445. What Is Nas?

It is one kind of file system where data can reside on one centralized machine and all the cluster member will read write data from that shared database, which would not be as efficient as HDFS.

446. How Hdfe Differs With Nfs?

Following are differences between HDFS and NAS

- In HDFS Data Blocks are distributed across local drives of all machines in a cluster. Whereas in NAS data is stored on dedicated hardware.
- HDFS is designed to work with MapReduce System, since computation is moved to data. NAS is not suitable for MapReduce since data is stored separately from the computations.
- HDFS runs on a cluster of machines and provides redundancy using replication protocol. Whereas NAS is provided by a single machine therefore does not provide data redundancy.

447. How Does A Namenode Handle The Failure Of The Data Nodes?

HDFS has master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the

cluster, which manage storage attached to the nodes that they run on. The NameNode and DataNode are pieces of software designed to run on commodity machines.

NameNode periodically receives a Heartbeat and a Block report from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. A Blockreport contains a list of all blocks on a DataNode. When NameNode notices that it has not received a heartbeat message from a data node after a certain amount of time, the data node is marked as dead. Since blocks will be under replicated the system begins replicating the blocks that were stored on the dead DataNode. The NameNode Orchestrates the replication of data blocks from one DataNode to another. The replication data transfer happens directly between DataNode and the data never passes through the NameNode.

448. Can Reducer Talk With Each Other?

No, Reducer runs in isolation.

449. Where The Mapper's Intermediate Data Will Be Stored?

The mapper output (intermediate data) is stored on the Local file system (NOT HDFS) of each individual mapper nodes. This is typically a temporary directory location which can be setup in config by the Hadoop administrator. The intermediate data is cleaned up after the Hadoop Job completes.

450. What Is The Use Of Combiners In The Hadoop Framework?

Combiners are used to increase the efficiency of a MapReduce program. They are used to aggregate intermediate map output locally on individual mapper outputs. Combiners can help you reduce the amount of data that needs to be transferred across to the reducers. You can use your reducer code as a combiner if the operation performed is commutative and associative. The execution of combiner is not guaranteed; Hadoop may or may not execute a combiner. Also, if required it may execute it more than 1 times.

Therefore your MapReduce jobs should not depend on the combiners' execution.

451. What Is The Hadoop Mapreduce Api Contract For A Key And Value Class?

- The Key must implement the `org.apache.hadoop.io.WritableComparable` interface.
- The value must implement the `org.apache.hadoop.io.Writable` interface.

452. What Is A Identitymapper And Identityreducer In Mapreduce?

- `org.apache.hadoop.mapred.lib.IdentityMapper`: Implements the identity function, mapping inputs directly to outputs. If MapReduce programmer does not set the Mapper Class using `JobConf.setMapperClass` then `IdentityMapper.class` is used as a default value.
- `org.apache.hadoop.mapred.lib.IdentityReducer` : Performs no reduction, writing all input values directly to the output. If MapReduce programmer does not set the Reducer Class using `JobConf.setReducerClass` then `IdentityReducer.class` is used as a default value.

453. What Is The Meaning Of Speculative Execution In Hadoop? Why Is It Important?

Speculative execution is a way of coping with individual Machine performance. In large clusters where hundreds or thousands of machines are involved there may be machines which are not performing as fast as others.

This may result in delays in a full job due to only one machine not performing well. To avoid this, speculative execution in hadoop can run

multiple copies of same map or reduce task on different slave nodes. The results from first node to finish are used.

454. When The Reducers Are Are Started In A Mapreduce Job?

In a MapReduce job reducers do not start executing the reduce method until the all Map jobs have completed. Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The programmer defined reduce method is called only after all the mappers have finished. If reducers do not start before all mappers finish then why does the progress on MapReduce job shows something like Map(50%) Reduce(10%)? Why reducers progress percentage is displayed when mapper is not finished yet? Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The progress calculation also takes in account the processing of data transfer which is done by reduce process, therefore the reduce progress starts showing up as soon as any intermediate key-value pair for a mapper is available to be transferred to reducer. Though the reducer progress is updated still the programmer defined reduce method is called only after all the mappers have finished.

455. What Is Hdfs ? How It Is Different From Traditional File Systems?

HDFS, the Hadoop Distributed File System, is responsible for storing huge data on the cluster. This is a distributed file system designed to run on commodity hardware.

It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant.

- HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware.
- HDFS provides high throughput access to application data and is suitable for applications that have large data sets.
- HDFS is designed to support very large files. Applications that are compatible with HDFS are those that deal with large data sets. These applications write their data only once but they

read it one or more times and require these reads to be satisfied at streaming speeds. HDFS supports write-once-read-many semantics on files.

456. What Is Hdfs Block Size? How Is It Different From Traditional File System Block Size?

In HDFS data is split into blocks and distributed across multiple nodes in the cluster. Each block is typically 64Mb or 128Mb in size. Each block is replicated multiple times. Default is to replicate each block three times. Replicas are stored on different nodes. HDFS utilizes the local file system to store each HDFS block as a separate file. HDFS Block size can not be compared with the traditional file system block size.

457. What Is A Namenode? How Many Instances Of Namenode Run On A Hadoop Cluster?

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself. There is only One NameNode process run on any hadoop cluster. NameNode runs on its own JVM process. In a typical production cluster its run on a separate machine.

The NameNode is a Single Point of Failure for the HDFS Cluster. When the NameNode goes down, the file system goes offline. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add /copy /move /delete a file. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives.

458. What Is A Datanode? How Many Instances Of Datanode Run On A Hadoop Cluster?

A DataNode stores data in the Hadoop File System HDFS. There is only One DataNode process run on any hadoop slave node. DataNode runs on its own

JVM process. On startup, a DataNode connects to the NameNode. DataNode instances can talk to each other, this is mostly during replicating data.

459. How The Client Communicates With Hdfs?

The Client communication to HDFS happens using Hadoop HDFS API. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file on HDFS. The NameNode responds the successful requests by returning a list of relevant DataNode servers where the data lives. Client applications can talk directly to a DataNode, once the NameNode has provided the location of the data.

460. How The Hdfs Blocks Are Replicated?

HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time. The NameNode makes all decisions regarding replication of blocks. HDFS uses rack-aware replica placement policy. In default configuration there are total 3 copies of a datablock on HDFS, 2 copies are stored on datanodes on same rack and 3rd copy on a different rack.

461. What Is Hadoop Framework?

Hadoop is a open source framework which is written in java by apache software foundation. This framework is used to wirite software application which requires to process vast amount of data (It could handle multi tera bytes of data). It works in-paralle on large clusters which could have 1000 of computers (Nodes) on the clusters. It also process data very reliably and fault-tolerant manner.

462. What Is Big Data?

Big Data is nothing but an assortment of such a huge and complex data that it becomes very tedious to capture, store, process, retrieve and analyze it with the help of on-hand database management tools or traditional data processing techniques.

463. Can You Give Some Examples Of Big Data?

There are many real life examples of Big Data! Facebook is generating 500+ terabytes of data per day, NYSE (New York Stock Exchange) generates about 1 terabyte of new trade data per day, a jet airline collects 10 terabytes of sensor data for every 30 minutes of flying time. All these are day to day examples of Big Data!

464. Can You Give A Detailed Overview About The Big Data Being Generated By Facebook?

As of December 31, 2012, there are 1.06 billion monthly active users on facebook and 680 million mobile users. On an average, 3.2 billion likes and comments are posted every day on Facebook. 72% of web audience is on Facebook. And why not! There are so many activities going on facebook from wall posts, sharing images, videos, writing comments and liking posts, etc. In fact, Facebook started using Hadoop in mid-2009 and was one of the initial users of Hadoop.

465. According To Ibm, What Are The Three Characteristics Of Big Data?

According to IBM, the three characteristics of Big Data are:

Volume: Facebook generating 500+ terabytes of data per day.

Velocity: Analyzing 2 million records each day to identify the reason for losses.

Variety: images, audio, video, sensor data, log files, etc.

466. How Analysis Of Big Data Is Useful For Organizations?

Effective analysis of Big Data provides a lot of business advantage as

organizations will learn which areas to focus on and which areas are less important. Big data analysis provides some early key indicators that can prevent the company from a huge loss or help in grasping a great opportunity with open hands! A precise analysis of Big Data helps in decision making! For instance, nowadays people rely so much on Facebook and Twitter before buying any product or service. All thanks to the Big Data explosion.

467. How Big Is 'big Data'?

With time, data volume is growing exponentially. Earlier we used to talk about Megabytes or Gigabytes. But time has arrived when we talk about data volume in terms of terabytes, petabytes and also zettabytes! Global data volume was around 1.8ZB in 2011 and is expected to be 7.9ZB in 2015. It is also known that the global information doubles in every two years!

468. Who Are 'data Scientists'?

Data scientists are soon replacing business analysts or data analysts. Data scientists are experts who find solutions to analyze data. Just as web analysis, we have data scientists who have good business insight as to how to handle a business challenge. Sharp data scientists are not only involved in dealing business problems, but also choosing the relevant issues that can bring value-addition to the organization.

469. Why The Name 'hadoop'?

Hadoop doesn't have any expanding version like 'oops'. The charming yellow elephant you see is basically named after Doug's son's toy elephant!

470. Why Do We Need Hadoop?

Everyday a large amount of unstructured data is getting dumped into our machines. The major challenge is not to store large data sets in our systems but to retrieve and analyze the big data in the organizations, that too data present in different machines at different locations. In this situation a necessity for Hadoop arises. Hadoop has the ability to analyze the data present in different machines at different locations very quickly and in a very

cost effective way. It uses the concept of MapReduce which enables it to divide the query into small parts and process them in parallel. This is also known as parallel computing.

471. What Are Some Of The Characteristics Of Hadoop Framework?

Hadoop framework is written in Java. It is designed to solve problems that involve analyzing large data (e.g. petabytes). The programming model is based on Google's MapReduce. The infrastructure is based on Google's Big Data and Distributed File System. Hadoop handles large files/data throughput and supports data intensive distributed applications. Hadoop is scalable as more nodes can be easily added to it.

472. Give A Brief Overview Of Hadoop History?

In 2002, Doug Cutting created an open source, web crawler project. In 2004, Google published MapReduce, GFS papers. In 2006, Doug Cutting developed the open source, Mapreduce and HDFS project.

In 2008, Yahoo ran 4,000 node Hadoop cluster and Hadoop won terabyte sort benchmark.

In 2009, Facebook launched SQL support for Hadoop.

473. Give Examples Of Some Companies That Are Using Hadoop Structure?

A lot of companies are using the Hadoop structure such as Cloudera, EMC, MapR, Hortonworks, Amazon, Facebook, eBay, Twitter, Google and so on.

474. 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 huge amount of data in the distributed file system and process it. 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.

475. What Is Structured And Unstructured Data?

Structured data is the data that is easily identifiable as it is organized in a structure. The most common form of structured data is a database where specific information is stored in tables, that is, rows and columns. Unstructured data refers to any data that cannot be identified easily. It could be in the form of images, videos, documents, email, logs and random text. It is not in the form of rows and columns.

476. What Are The Core Components Of Hadoop?

Core components of Hadoop are HDFS and MapReduce. HDFS is basically used to store large data sets and MapReduce is used to process such large data sets.

477. What Is Hdfs?

HDFS is a file system designed for storing very large files with streaming data access patterns, running clusters on commodity hardware.

478. What Are The Key Features Of Hdfs?

HDFS is highly fault-tolerant, with high throughput, suitable for applications with large data sets, streaming access to file system data and can be built out of commodity hardware.

479. What Is Fault Tolerance?

Suppose you have a file stored in a system, and due to some technical problem that file gets destroyed. Then there is no chance of getting the data back present in that file. To avoid such situations, Hadoop has introduced the feature of fault tolerance in HDFS. In Hadoop, when we store a file, it automatically gets replicated at two other locations also. So even if one or two of the systems collapse, the file is still available on the third system.

480. Replication Causes Data Redundancy Then Why Is It Pursued In Hdfs?

HDFS works with commodity hardware (systems with average

configurations) that has high chances of getting crashed any time. Thus, to make the entire system highly fault-tolerant, HDFS replicates and stores data in different places. Any data on HDFS gets stored at at least 3 different locations. So, even if one of them is corrupted and the other is unavailable for some time for any reason, then data can be accessed from the third one. Hence, there is no chance of losing the data. This replication factor helps us to attain the feature of Hadoop called Fault Tolerant.

481. Since The Data Is Replicated Thrice In Hdfs, Does It Mean That Any Calculation Done On One Node Will Also Be Replicated On The Other Two?

Since there are 3 nodes, when we send the MapReduce programs, calculations will be done only on the original data. The master node will know which node exactly has that particular data. In case, if one of the nodes is not responding, it is assumed to be failed. Only then, the required calculation will be done on the second replica.

482. What Is Throughput? How Does Hdfs Get A Good Throughput?

Throughput is the amount of work done in a unit time. It describes how fast the data is getting accessed from the system and it is usually used to measure performance of the system. In HDFS, when we want to perform a task or an action, then the work is divided and shared among different systems. So all the systems will be executing the tasks assigned to them independently and in parallel. So the work will be completed in a very short period of time. In this way, the HDFS gives good throughput. By reading data in parallel, we decrease the actual time to read data tremendously.

483. What Is Streaming Access?

As HDFS works on the principle of 'Write Once, Read Many', the feature of streaming access is extremely important in HDFS. HDFS focuses not so much on storing the data but how to retrieve it at the fastest possible speed, especially while analyzing logs. In HDFS, reading the complete data is more important than the time taken to fetch a single record from the data.

484. What Is A Commodity Hardware? Does Commodity Hardware Include Ram?

Commodity hardware is a non-expensive system which is not of 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 there will be some services which will be running on RAM.

485. Is Namenode Also A Commodity?

No. Namenode can never be a commodity hardware because the entire HDFS rely on it. It is the single point of failure in HDFS. Namenode has to be a high-availability machine.

486. What Is A Metadata?

Metadata is the information about the data stored in data nodes such as location of the file, size of the file and so on.

487. What Is A Daemon?

Daemon is a process or service that runs in background. In general, we use this word in UNIX environment. The equivalent of Daemon in Windows is "services" and in Dos is "TSR".

488. What Is A Job Tracker?

Job tracker is a daemon that runs on a namenode for submitting and tracking MapReduce jobs in Hadoop. It assigns the tasks to the different task tracker. In a Hadoop cluster, there will be only one job tracker but many task trackers. It is the single point of failure for Hadoop and MapReduce Service. If the job tracker goes down all the running jobs are halted. It receives heartbeat from task tracker based on which Job tracker decides whether the assigned task is completed or not.

489. What Is A Task Tracker?

Task tracker is also a daemon that runs on datanodes. Task Trackers

manage the execution of individual tasks on slave node. When a client submits a job, the job tracker will initialize the job and divide the work and assign them to different task trackers to perform MapReduce tasks. While performing this action, the task tracker will be simultaneously communicating with job tracker by sending heartbeat. If the job tracker does not receive heartbeat from task tracker within specified time, then it will assume that task tracker has crashed and assign that task to another task tracker in the cluster.

490. Is Namenode Machine Same As Datanode Machine As In Terms Of Hardware?

It depends upon the cluster you are trying to create. The Hadoop VM can be there on the same machine or on another machine. For instance, in a single node cluster, there is only one machine, whereas in the development or in a testing environment, Namenode and data nodes are on different machines.

491. What Is A Heartbeat In Hdfs?

A heartbeat is a signal indicating that it is alive. A datanode sends heartbeat to Namenode and task tracker will send its heart beat to job tracker. If the Namenode or job tracker does not receive heart beat then they will decide that there is some problem in datanode or task tracker is unable to perform the assigned task.

492. Are Namenode And Job Tracker On The Same Host?

No, in practical environment, Namenode is on a separate host and job tracker is on a separate host.

493. What Is A 'block' In Hdfs?

A 'block' is the minimum amount of data that can be read or written. In HDFS, the default block size is 64 MB as contrast to the block size of 8192 bytes in Unix/Linux. Files in HDFS are broken down into block-sized chunks, which are stored as independent units. HDFS blocks are large as compared to disk blocks, particularly to minimize the cost of seeks.

494. If A Particular File Is 50 Mb, Will The Hdfs Block Still Consume 64 Mb As The Default Size?

No, not at all! 64 mb is just a unit where the data will be stored. In this particular situation, only 50 mb will be consumed by an HDFS block and 14 mb will be free to store something else. It is the MasterNode that does data allocation in an efficient manner.

495. What Are The Benefits Of Block Transfer?

A file can be larger than any single disk in the network. There's nothing that requires the blocks from a file to be stored on the same disk, so they can take advantage of any of the disks in the cluster. Making the unit of abstraction a block rather than a file simplifies the storage subsystem. Blocks provide fault tolerance and availability. To insure against corrupted blocks and disk and machine failure, each block is replicated to a small number of physically separate machines (typically three). If a block becomes unavailable, a copy can be read from another location in a way that is transparent to the client.

496. If We Want To Copy 10 Blocks From One Machine To Another, But Another Machine Can Copy Only 8.5 Blocks, Can The Blocks Be Broken At The Time Of Replication?

In HDFS, blocks cannot be broken down. Before copying the blocks from one machine to another, the Master node will figure out what is the actual amount of space required, how many block are being used, how much space is available, and it will allocate the blocks accordingly.

497. How Indexing Is Done In Hdfs?

Hadoop has its own way of indexing. Depending upon the block size, once the data is stored, HDFS will keep on storing the last part of the data which will say where the next part of the data will be. In fact, this is the base of HDFS.

498. If A Data Node Is Full How It's Identified?

When data is stored in datanode, then the metadata of that data will be stored in the Namenode. So Namenode will identify if the data node is full.

499. If Datanodes Increase, Then Do We Need To Upgrade Namenode?

While installing the Hadoop system, Namenode is determined based on the size of the clusters. Most of the time, we do not need to upgrade the Namenode because it does not store the actual data, but just the metadata, so such a requirement rarely arise.

500. Are Job Tracker And Task Trackers Present In Separate Machines?

Yes, job tracker and task tracker are present in different machines. The reason is job tracker is a single point of failure for the Hadoop MapReduce service. If it goes down, all running jobs are halted.

501. When We Send A Data To A Node, Do We Allow Settling In Time, Before Sending Another Data To That Node?

Yes, we do.

502. Does Hadoop Always Require Digital Data To Process?

Yes. Hadoop always require digital data to be processed.

503. On What Basis Namenode Will Decide Which Datanode To Write On?

As the Namenode has the metadata (information) related to all the data nodes, it knows which datanode is free.

504. Doesn't Google Have Its Very Own Version Of Dfs?

Yes, Google owns a DFS known as “Google File System (GFS)” developed by Google Inc. for its own use.

505. Who Is A 'user' In Hdfs?

A user is like you or me, who has some query or who needs some kind of data.

506. Is Client The End User In Hdfs?

No, Client is an application which runs on your machine, which is used to interact with the Namenode (job tracker) or datanode (task tracker).

507. What Is The Communication Channel Between Client And Namenode/datanode?

The mode of communication is SSH.

508. What Is A Rack?

Rack is a storage area with all the datanodes put together. These datanodes can be physically located at different places. Rack is a physical collection of datanodes which are stored at a single location. There can be multiple racks in a single location.

509. On What Basis Data Will Be Stored On A Rack?

When the client is ready to load a file into the cluster, the content of the file will be divided into blocks. Now the client consults the Namenode and gets 3 datanodes for every block of the file which indicates where the block should be stored. While placing the datanodes, the key rule followed is “for every block of data, two copies will exist in one rack, third copy in a different rack“. This rule is known as “Replica Placement Policy“.

510. Do We Need To Place 2nd And 3rd Data In Rack 2 Only?

Yes, this is to avoid datanode failure.

511. What If Rack 2 And Datanode Fails?

If both rack2 and datanode present in rack 1 fails then there is no chance of getting data from it. In order to avoid such situations, we need to replicate that data more number of times instead of replicating only thrice. This can

be done by changing the value in replication factor which is set to 3 by default.

512. What Is A Secondary Namenode? Is It A Substitute To The Namenode?

The secondary Namenode constantly reads the data from the RAM of the Namenode and writes it into the hard disk or the file system. It is not a substitute to the Namenode, so if the Namenode fails, the entire Hadoop system goes down.

513. What Is The Difference Between Gen1 And Gen2 Hadoop With Regards To The Namenode?

In Gen 1 Hadoop, Namenode is the single point of failure. In Gen 2 Hadoop, we have what is known as Active and Passive Namenodes kind of a structure. If the active Namenode fails, passive Namenode takes over the charge.

514. Can You Explain How Do 'map' And 'reduce' Work?

Namenode takes the input and divide it into parts and assign them to data nodes. These datanodes process the tasks assigned to them and make a key-value pair and returns the intermediate output to the Reducer. The reducer collects this key value pairs of all the datanodes and combines them and generates the final output.

515. What Is 'key Value Pair' In Hdfs?

Key value pair is the intermediate data generated by maps and sent to reduces for generating the final output.

516. What Is The Difference Between Mapreduce Engine And Hdfs Cluster?

HDFS cluster is the name given to the whole configuration of master and slaves where data is stored. Map Reduce Engine is the programming module which is used to retrieve and analyze data.

517. Is Map Like A Pointer?

No, Map is not like a pointer.

518. Do We Require Two Servers For The Namenode And The Datanodes?

Yes, we need two different servers for the Namenode and the datanodes. This is because Namenode requires highly configurable system as it stores information about the location details of all the files stored in different datanodes and on the other hand, datanodes require low configuration system.

519. Why Are The Number Of Splits Equal To The Number Of Maps?

The number of maps is equal to the number of input splits because we want the key and value pairs of all the input splits.

520. Is A Job Split Into Maps?

No, a job is not split into maps. Split is created for the file. The file is placed on datanodes in blocks. For each split, a map is needed.

521. Which Are The Two Types Of 'writes' In Hdfs?

There are two types of writes in HDFS: posted and non-posted write. Posted Write is when we write it and forget about it, without worrying about the acknowledgement. It is similar to our traditional Indian post. In a Non-posted Write, we wait for the acknowledgement. It is similar to the today's courier services. Naturally, non-posted write is more expensive than the posted write. It is much more expensive, though both writes are asynchronous.

522. Why 'reading' Is Done In Parallel And 'writing' Is Not In Hdfs?

Reading is done in parallel because by doing so we can access the data fast. But we do not perform the write operation in parallel. The reason is that if we perform the write operation in parallel, then it might result in data inconsistency. For example, you have a file and two nodes are trying to write

data into the file in parallel, then the first node does not know what the second node has written and vice-versa. So, this makes it confusing which data to be stored and accessed.

523. Can Hadoop Be Compared To Nosql Database Like Cassandra?

Though NOSQL is the closet technology that can be compared to Hadoop, it has its own pros and cons. There is no DFS in NOSQL. Hadoop is not a database. It's a file system (HDFS) and distributed programming framework (MapReduce).

524. How Can I Install Cludera Vm In My System?

When you enrol for the hadoop course at Edureka, you can download the Hadoop Installation steps.pdf file from our dropbox.

525. How Jobtracker Schedules A Task?

The TaskTrackers send out heartbeat messages to the JobTracker, usually every few minutes, to reassure the JobTracker that it is still alive. These message also inform the JobTracker of the number of available slots, so the JobTracker can stay up to date with where in the cluster work can be delegated. When the JobTracker tries to find somewhere to schedule a task within the MapReduce operations, it first looks for an empty slot on the same server that hosts the DataNode containing the data, and if not, it looks for an empty slot on a machine in the same rack.

526. What Is A Task Tracker In Hadoop? How Many Instances Of Tasktracker Run On A Hadoop Cluster?

A TaskTracker is a slave node daemon in the cluster that accepts tasks (Map, Reduce and Shuffle operations) from a JobTracker. There is only One Task Tracker process run on any hadoop slave node. Task Tracker runs on its own JVM process. Every TaskTracker is configured with a set of slots, these indicate the number of tasks that it can accept. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the task tracker. The

TaskTracker monitors these task instances, capturing the output and exit codes. When the Task instances finish, successfully or not, the task tracker notifies the JobTracker. The TaskTrackers also send out heartbeat messages to the JobTracker, usually every few minutes, to reassure the JobTracker that it is still alive. These message also inform the JobTracker of the number of available slots, so the JobTracker can stay up to date with where in the cluster work can be delegated.

527. What Is A Task Instance In Hadoop? Where Does It Run?

Task instances are the actual MapReduce jobs which are run on each slave node. The TaskTracker starts a separate JVM processes to do the actual work (called as Task Instance) this is to ensure that process failure does not take down the task tracker. Each Task Instance runs on its own JVM process. There can be multiple processes of task instance running on a slave node. This is based on the number of slots configured on task tracker. By default a new task instance JVM process is spawned for a task.

528. How Many Daemon Processes Run On A Hadoop System?

Hadoop is comprised of five separate daemons. Each of these daemon run in its own JVM. Following 3 Daemons run on Master nodes
NameNode : This daemon stores and maintains the metadata for HDFS.
Secondary NameNode : Performs housekeeping functions for the NameNode.

JobTracker : Manages MapReduce jobs, distributes individual tasks to machines running the Task Tracker.

Following 2 Daemons run on each Slave nodes

DataNode : Stores actual HDFS data blocks.

TaskTracker : Responsible for instantiating and monitoring individual Map and Reduce tasks.

529. What Is Configuration Of A Typical Slave Node On Hadoop Cluster?

How Many Jvms Run On A Slave Node?

Single instance of a Task Tracker is run on each Slave node. Task tracker is run as a separate JVM process.

- Single instance of a DataNode daemon is run on each Slave node. DataNode daemon is run as a separate JVM process.
- One or Multiple instances of Task Instance is run on each slave node. Each task instance is run as a separate JVM process. The number of Task instances can be controlled by configuration. Typically a high end machine is configured to run more task instances.

530. What Is The Difference Between Hdfs And Nas ?

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. Following are differences between HDFS and NAS

- In HDFS Data Blocks are distributed across local drives of all machines in a cluster. Whereas in NAS data is stored on dedicated hardware.
- HDFS is designed to work with MapReduce System, since computation are moved to data. NAS is not suitable for MapReduce since data is stored separately from the computations.
- HDFS runs on a cluster of machines and provides redundancy using a replication protocol. Whereas NAS is provided by a single machine therefore does not provide data redundancy.

531. How Namenode Handles Data Node Failures?

NameNode periodically receives a Heartbeat and a Blockreport from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. A Blockreport contains a list of all blocks

on a DataNode. When NameNode notices that it has not received a heartbeat message from a data node after a certain amount of time, the data node is marked as dead. Since blocks will be under replicated the system begins replicating the blocks that were stored on the dead datanode. The NameNode Orchestrates the replication of data blocks from one datanode to another. The replication data transfer happens directly between datanodes and the data never passes through the namenode.

532. Does Mapreduce Programming Model Provide A Way For Reducers To Communicate With Each Other? In A Mapreduce Job Can A Reducer Communicate With Another Reducer?

Nope, MapReduce programming model does not allow reducers to communicate with each other. Reducers run in isolation.

533. Can I Set The Number Of Reducers To Zero?

Yes, Setting the number of reducers to zero is a valid configuration in Hadoop. When you set the reducers to zero no reducers will be executed, and the output of each mapper will be stored to a separate file on HDFS. [This is different from the condition when reducers are set to a number greater than zero and the Mappers output (intermediate data) is written to the Local file system(NOT HDFS) of each mapper slave node.]

534. Where Is The Mapper Output (intermediate Key-value Data) Stored ?

The mapper output (intermediate data) is stored on the Local file system (NOT HDFS) of each individual mapper nodes. This is typically a temporary directory location which can be setup in config by the hadoop administrator. The intermediate data is cleaned up after the Hadoop Job completes.

535. What Are Combiners? When Should I Use A Combiner In My Mapreduce Job?

Combiners are used to increase the efficiency of a MapReduce program. They

are used to aggregate intermediate map output locally on individual mapper outputs. Combiners can help you reduce the amount of data that needs to be transferred across to the reducers. You can use your reducer code as a combiner if the operation performed is commutative and associative. The execution of combiner is not guaranteed, Hadoop may or may not execute a combiner. Also, if required it may execute it more than 1 times. Therefore your MapReduce jobs should not depend on the combiners execution.

536. What Is Writable & WritableComparable Interface?

- `org.apache.hadoop.io.Writable` is a Java interface. Any key or value type in the Hadoop Map-Reduce framework implements this interface. Implementations typically implement a static `read(DataInput)` method which constructs a new instance, calls `readFields(DataInput)` and returns the instance.
- `org.apache.hadoop.io.WritableComparable` is a Java interface. Any type which is to be used as a key in the Hadoop Map-Reduce framework should implement this interface. `WritableComparable` objects can be compared to each other using Comparators.

537. What Is A Identitymapper And Identityreducer In Mapreduce ?

`org.apache.hadoop.mapred.lib.IdentityMapper` Implements the identity function, mapping inputs directly to outputs. If MapReduce programmer do not set the Mapper Class using `JobConf.setMapperClass` then `IdentityMapper.class` is used as a default value.

- `org.apache.hadoop.mapred.lib.IdentityReducer` Performs no reduction, writing all input values directly to the output. If MapReduce programmer do not set the Reducer Class using `JobConf.setReducerClass` then `IdentityReducer.class` is used as a default value.

538. When Is The Reducers Are Started In A Mapreduce Job?

In a MapReduce job reducers do not start executing the reduce method until the all Map jobs have completed. Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The programmer defined reduce method is called only after all the mappers have finished.

539. If Reducers Do Not Start Before All Mappers Finish Then Why Does The Progress On Mapreduce Job Shows Something Like Map(50%) Reduce(10%)? Why Reducers Progress Percentage Is Displayed When Mapper Is Not Finished Yet?

Reducers start copying intermediate key-value pairs from the mappers as soon as they are available. The progress calculation also takes in account the processing of data transfer which is done by reduce process, therefore the reduce progress starts showing up as soon as any intermediate key-value pair for a mapper is available to be transferred to reducer. Though the reducer progress is updated still the programmer defined reduce method is called only after all the mappers have finished.

