Hadoop Interview Questions for 2021

1. What is Big Data?

Any form of data that is difficult to capture, arrange or analyse can be termed ‘big data’. However, in the world of analytics, big data is usually referred to as a collection of large and complex sets of information. The utilization of these sets is not possible through traditional methods and tools. An example of such data could be- collection of information of each person who has existed in the world and who had a permanent address.

Tools like Apache Hadoop and its extended family of software can be used for analysis of such big data.

2. Describe Hadoop and its components

Hadoop is a family of software that can be used to store, analyse and process big data. It digs through big data and provides insights that a business can use to improve the development in its sector. In the above example, a country’s government can use that data to create a solid census report.

The two main components of Hadoop are:

- Storage Unit known as Hadoop Distributed File System (HDFS)
- Processing framework known as Yet Another Resource Negotiator (YARN)

These two components further have sub-components that carry out multiple tasks.

3. What are the various daemons of Hadoop?

Here are the various Hadoop daemons and their functions within the system:

- NameNode – master node; responsible for storing the metadata of all the files and directories
- DataNode – slave node; contains actual data

- Secondary NameNode – used in case of a failure of NameNode; it refreshes content periodically
- ResourceManager – central authority; manages resources and scheduling
- NodeManager – runs on slave machines and carries out most tasks like application execution and monitoring CPU usage; reports to ResourceManager

4. What are the various steps involved in the deploying of big-data solution?

The various steps which are involved in the big-data solution are:

- **Data Ingestion**

Data Ingestion is the fore-most procedure while deploying the big-data solution in order to extract the data from the diversified sources such as, ERP system (SAP), any CRM's like Siebel and Salesforce, Relational Database Management System such as Oracle and MySQL, or either could be flat-files, log-files, images, documents and the social-media feeds. This particular data is to be stored in the HDFS.

- **Data Storage**

After ingesting the data, the subsequent procedure is to store the data either in NoSQL database such as, HBase or HDFS. HDFS being optimized for the sequential access whereas, the HBase storage work for the access of random read or write.

- **Data Processing**

Data processing is the ultimate step for the processing of data using any of these processing frameworks such as Spark, Pig, MapReduce, Hive, etc.

5. What is Hadoop MapReduce?

Hadoop MapReduce is a framework that is used to process large amounts of data in a Hadoop cluster. It reduces time consumption as compared to the alternative method of data analysis. The uniqueness of MapReduce is that it runs tasks simultaneously across clusters to reduce processing time.

6. In Map Reduce Programming, why we use the sorting and shuffling phase. What is the purpose of using this?

In Map Reduce Programming, the mapper and the reducer are the two important phases, where the sorting and the shuffling are the two major operations in the map-reduce. As, the Hadoop framework basically takes the structured or unstructured data and then separate that data in key, value pair, where the mapper programs separates and arranges the data in the key and value to use it for further processing. The sorting and shuffling phase is done by the frame-work, where the data from each mapper being grouped by key and splits among the reducers and further sorted by key. Each reducer obtains all the values which are associated with the same key.

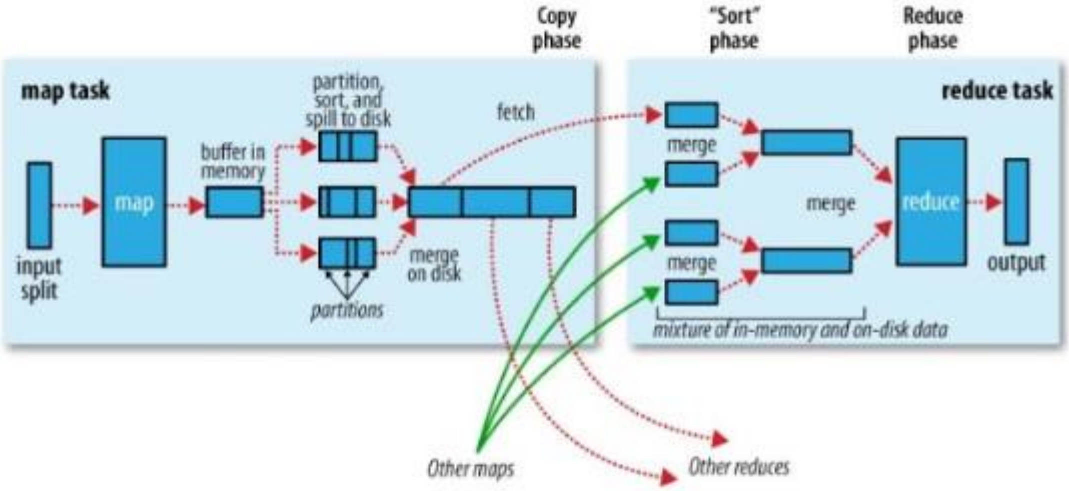


Fig: Graphical representation of sorting and shuffling phase (where shuffle is denoted by "copy")

Shuffling is the data-transfer process from mappers to reducers, thus it is being necessary for reducer. Shuffling process can gets start before the finish of map phase, in order to save some time. This is the reason of the reduce status to be greater than of 0% but less than that of 33%, while the map-status not achieved 100% at the same time.

Sorting, starts the newly reduce task when next key in sorted input-data is being different from the previous one. Each of the reduce task takes the key-value pairs list, in order to group the values by the keys, by calling the reduce() method whose input is the key-list(value).

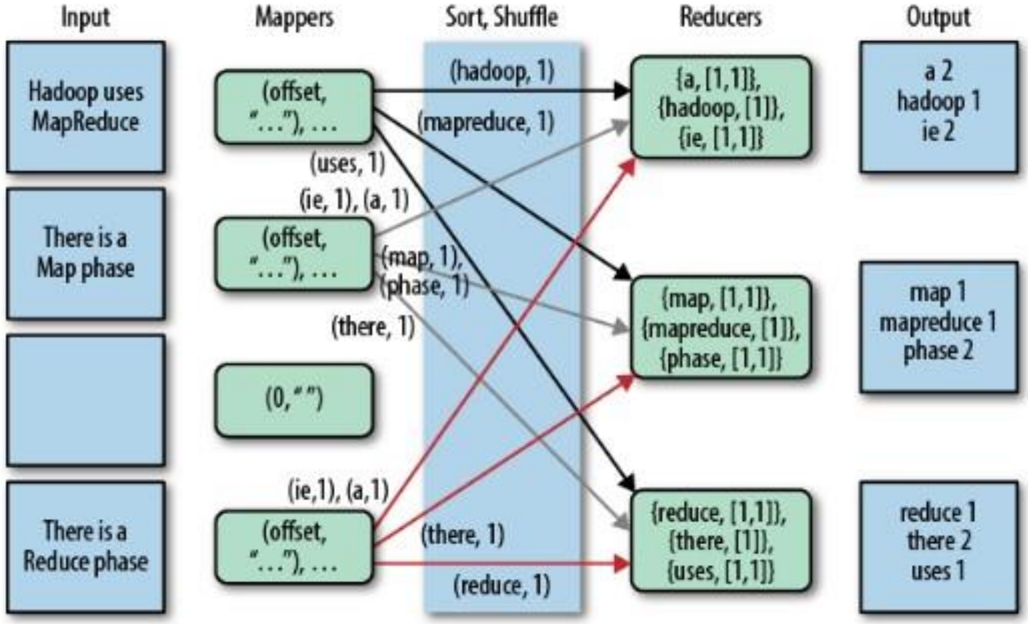


Fig: String based MapReduce program

In the Shuffling process, the intermediate data from the mappers gets transferred to 0, 1, or more reducers. Where each reducer gets one or more keys where its associated values depends on number of the reducers for the balance load. On the other hand, the values with each key are sorted locally. The sorting and shuffling will not be performed if we specify the zero reducers (i.e., setNumReduceTasks(0)). In this case, MapReduce stops at map phase and does not includes any kinds of sorting.

7. What are some Hadoop input formats?

There are three well-known input formats, namely:

- Text – lines of text
- Value – plaintext files
- Sequence – multiple files fed in a sequence

Text is the input format that is used as default.

8. What is a heartbeat in Hadoop parlance?

Heartbeat is a form of communication (a signal) shared between a data node and NameNode. If the NameNode or job tracker does not respond to this communication attempt, it means that there is an error in the system.

Also Read: Top Hadoop Jobs 2020

9. How do you fix NameNode when it is down?

The following steps can be followed to fix NameNode:

- FsImage, the file systems metadata replica, should be used to start a new NameNode
- Configuration of datanodes to acknowledge the creation of this new NameNode
- NameNode will begin its operation and the cluster will go back to normalcy after it has completely loaded the last FsImage checkpoint.

In some cases, NameNode revival can take a lot of time.

10. Differentiate between Hadoop 1 and Hadoop 2

The differentiation between Hadoop 1 and Hadoop 2 can be evinced through two parameters, as shown in the table below:

Parameters	Hadoop 1	Hadoop 2
Passive NameNode	Since it is the single point of failure, NameNode has to be revived to resume an application	It is not the single point of failure; passive NameNode can quickly take its place

Processing (YARN)	Limited to the MRV1 structure where other tools cannot take up the task of processing	MRV2 structure

11. What is FsImage? Explain its salient features

FsImage is a log of checkpoints of the namespace. The latest checkpoint allows the revival of NameNode and other elements whenever necessary.

12. What is a Checkpoint?

A checkpoint is the last load of saved data. It captures FsImage and edits the namespace log, then compacts both into a new FsImage. This is a continuous process. This task of creating a checkpoint is performed by Secondary NameNode.

13. Define the role of JobTracker in Hadoop?

Resource management, tracking resources as and when they are added or updated, and task life cycle management are some of the key responsibilities of JobTracker. In more detail:

- It identifies data location and communicates with NameNode
- Executes tasks by finding the best available nodes through TaskTracker
- Assigns overloads to slave nodes whenever necessary
- Monitoring of TaskTrackers

Essentially, a JobTracker works like a maintenance guy in the Hadoop ecosystem.

14. Name one major drawback of Hadoop?

One major drawback of Hadoop is the limit function security. Written on Java and crowdsourced, it is heavily vulnerable to hacks. This is a serious problem since critical data is stored and processed here.

15. What will happen if you try to pull a Hadoop job from an existing output directory?

The dialog will throw an error and say that an output file directory already exists. This is not ideal because, to run a MapReduce job one needs to ensure there is no directory present. In such a case, it has to be deleted. The shell can be used to delete the directory:

16. Explain how will you choose various file-formats in order to store and process the data using Apache Hadoop?

The decision of choosing the particular format of file is based on the following factors such as:

- Schema evolution in order to alter, add and rename the fields.
- Usage of patterns such as access of the 5 columns out of the 50 columns V/S access of most of the columns.
- Parallel processing of split-ability.
- Transfer/read/write performance to the block-compression of storage space saving.

There are various file-formats which are used with the Hadoop such as, JSON, CSV, Sequential files, Columnar, Parquet files and AVRO.

JSON Files

Each of the JSON files have their own record. The JSON store the record of both schema and data together. It also enables the schema evolution and the split-ability completely. However, the block-level compression is not supported in the JSON file format.

CSV Files

For the exchange of data between the Hadoop and the external system, the CSV files is the ideal fit for this. The header and the footer lines are not used while using the CSV files format.

Parquet Files

Parquet files are basically the columnar file-format which supports the block-level compression. It is also optimized for the query performance, as it allows the selection of ten or minimum number of columns from about 50+ records of column.

AVRO Files

For the long-term schema storage, AVRO file-format is best -suited. AVRO file store the meta-data with the data and also specify the independent schema in order to read the files.

Hadoop fs –rmr /path/to/your/output/

17. List the advantages of Apache Pig over MapReduce

Following are some of the major merits of Apache Pig:

- No need of Java implementations to carry out high-level data exchange. There are presets available
- Length of code is reduced by 20 times (compared to MapReduce)
- Addition of several built-in operations like joins, filters, and sorting without the need for additional implementation
- A Join operation can be executed singularly without the need for extra resources
- Supports nested data types.

All in all, Apache Pig works more efficiently as a high-level data flow language.

18. List the general steps to debug a code in Hadoop

Following are the steps involved in debugging a code:

- Check the list of MapReduce jobs currently running
- If orphaned jobs are running, check the ResourceManager by executing the following code
  - ps –ef | grep –I ResourceManager
- Check the log directory to detect any error messages that may be shown

- Basis the logs found in the above step, check the worker node involved in the action that may have the buggy code
- Log in to the node by executing the following code
    - ps –ef | grep –iNodeManager
- Examination of MapReduce log to find out the source of error.

This is the process for most error-detection tasks in the Hadoop cluster system.

19. If there are 8TB be the available disk space per node (i.e., 10 disks having 1TB, 2 disks is for Operating-System etc., were excluded). Then how will you estimate the number of data nodes(n)? (Assuming the initial size of data is 600 TB)

In the Hadoop environment, the estimation of hardware-requirements is challenging due to the increased of data at any-time in the organization. Thus, one must have the proper knowledge of the cluster based on the current scenario which depends on the following factor:

1. The actual data size to be store is around 600TB.
2. The rate of increase of data in future (daily/weekly/monthly/quarterly/yearly) depends on the prediction of the analysis of tending of data and the justified requirements of the business.
3. There is a default of 3x replica factor for the Hadoop.
4. For the overhead of the hardware machine (such as logs, Operating System etc.) the two disks were considered.
5. The output data on hard-disk is 1x for the intermediate reducer and mapper.
6. Space utilization is in between 60-70%.

Steps to find the number of the data-nodes which are required to store 600TB data:

- Given Replication factor: 3

Data size: 600TB

Intermediate data: 1

Requirements of total storage: 3+1*600=2400 TB

Available disk-size: 8TB

Total data-nodes required: 24008=300 machines.

- Actual Calculation = Disk-space utilization + Rough Calculation + Compression Ratio

Disk-space utilization: 65%

Compression ratio: 2.3

Total requirement of storage: 24002.3=1043.5TB

Available size of disk: 8*0.65=5.2 TB

Total data-nodes required: 1043.55.2=201 machines.

Actual usable size of cluster (100%): 201*8*2.34=925 TB

- Case: It has been predicted that there is 20% of the increase of data in quarter and we all need to predict is the new machines which is added in particular year

Increase of data: 20% quarterly

Additional data:

1st quarter: 1043.5*0.2=208.7 TB

2nd quarter: 1043.5*1.2*0.2=250.44 TB

3rd quarter: 1043.5*1.2*1.2*0.2=300.5 TB

4th quarter: 1043.5*1.2*1.2*1.2*0.2=360.6 TB

Additional data-nodes:

1st quarter: 208.75.2=41 machines

2nd quarter: 250.445.2=49 machines

3rd quarter: 300.55.2=58 machines

4th quarter: 360.65.2=70 machines

20. If you have a directory that contains these files- HadoopTraining.txt, _SpartTraining.txt, #DataScienceTraining.txt, .SalesforceTraining.txt. Then how many files are to be processed if you pass on this directory to Hadoop MapReduce jobs?

#DataScienceTraining.txt and HadoopTraining.txt will processed for the MapReduce jobs while processing the file (either individual or in directory) in the Hadoop using any of the FileInputFormat as, the KeyValueInputFormat, the TextInputFormat or the SequenceFileInputFormat, one have to confirm that none of that files contains the hidden file-prefix as, “_”, or “.” The reason is that the mapreduce FileInputFormat will be by default use the hiddenFileFilter class in order to ignore the files with any of these prefix names.

```
private static final PathFilter hiddenFileFilter = new
PathFilter() {

public boolean accept(Path p) {

String name = p.getName();

return !name.startsWith(“_”) && !name.startsWith(“.”);

}

};
```

However, hiddenFileFilter will always active even though if one uses the custom filter like FileInputFormat.setInputPathFilter in order to eliminate such criteria.

21. If one is uploading a file of 500MB into the HDFS. If 100MB of data has been successfully uploaded in the HDFS and the other client is about to read the uploaded data while the upload is still to be in progress. Then what will happen? Whether that 100MB of data which is uploaded will be displayed or not?

The default block-size of Hadoop1x is 64MB and of Hadoop2x is 128MB.

Let the block-size be 100MB, i.e., five blocks are to replicated three times (the default replication-factor)

Below procedure describes how the block is to be write in the HDFS:

If we have A, B, C, D and E be the five blocks for client, file, name-node and data-node. Then firstly, the client takes the Block A and approaches the name-node for the data-node location in order to store this current block and replicated copies of it. Once the data-node information is available to the client, he will reach directly to the data-node and starts the copying of Block A, which will at the same time gets replicated to second data-node. When the block gets copied and replicated to data-node, the confirmation of Block A storage will get to the client, then further, the client will re-start the same procedure for the next block i.e., Block B.

Thus, if one is uploading a file of 500MB into the HDFS where 100MB of data has been successfully uploaded in the HDFS and the other client is about to read the uploaded data while the upload is still to be in progress then only the present block which is being written will not be visible to the readers.

22. Why should we stop all tasks trackers while decommissioning nodes in the Hadoop cluster?

One should be very well aware of the complete procedure of decommissioning the data-node in the Hadoop cluster, but it is to be taken care of when the task trackers runs the MapReduce jobs on the data-node which is decommissioned. Unlike data-node, there is not any graceful way of decommissioning the task-tracker, where assumption is made as whenever the present task is to be moved to the another node then one should rely on task making process in order to stop from failure, and further it will be rescheduled on the cluster. Possibility is that when the final attempt of task runs on the task-tracker then the final failure will result on the entire failing of the job. So, the decommissioning stops the data-node, but in order to move the present task to the other node, one should manually stop the task-tracker which is running on the decommissioning node.

23. When does the NameNode enters a safe-mode?

The NameNode being responsible to manage the cluster's meta-storage, and if there is anything which is missing from cluster then the NameNode will held where all the crucial information is checked during the safe-mode before the cluster is available for writing to users.

There are several reasons when the NameNode enters the safe-mode during start-up as:

- NameNode load the file-system state from the fsimage and edits log-file, and waits for the data-nodes in order to report the blocks. Thus, the replication of the blocks could not start if it already exists in another cluster.
- Heartbeats from the data-nodes and the corrupt blocks exists in a cluster. Once all the relevant information gets verified by the NameNode, then it leaves the safe-mode and the cluster then gets accessible.

In order to manually enter or leave the safe-mode from the NameNode, the below command is used in the command line:

"hdfs dfsadmin -safemode enter/leave"

24. Did you ever run a lopsided job which resulted in and out of memory ever? If yes, then how will you handle it?

In the MapReduce jobs “OutOfMemoryError” is the common error which occur as the data grows with different sizes makes a challenge to the developer in order estimate the amount of memory required to allocate the job.

Thus, the following properties has to be set in an appropriate manner in order to consider the resources which are available in a cluster in order to avoid the out-of-memory error:

- mapreduce.map.memory.mb

Maximum memory used by the mapper in a container.

- mapreduce.map.java.opts

Maximum heap size used by the mapper. It must be less than the mapreduce.map.memory.mb size.

- mapreduce.reduce.memory.mb

Maximum memory which is used by the reducer in a container.

- mapreduce.reduce.java.opts

Maximum heap-size which is used by the reducer. It must be less than mapreduce.reduce.memory.mb size.

- yarn.schedule.maximum-allocation-mb

Allowed maximum allocation-size for the container, also requires the administrative privilege.

25. Explain the Erasure Coding in the Hadoop?

By default, HDFS replicate each of the block to three times in the Hadoop. HDFS replication is simple and have the robust form redundancy in order to shield the failure of the data-node. However, the replication is quite expensive. The 3x scheme of replication has 200% of overhead in the storage space.

The Hadoop2.x introduced the Erasure Coding in place of the Replication. The same level of the fault-tolerance with the less space-store and of 50% overhead storage is also provided in this. The Erasure coding uses the RAID (Redundant Array of Inexpensive Disk), which implements through striping in which the logical-sequential data is divided in the small units such as, byte, bit or blocks. Then, on the different disk this data is stored.

Encoding: Here, RAID calculate and then sort the parity cells for each strips of the data-cells, and recovers the error through parity. EC extends the message with the redundant data for fault-tolerant. The Erasure Coding codec operate on the data-cells which are uniformly sized. It takes the data-cells as input and produces the parity-cells as output. The data-cells and the parity-cells together called the EC group.

There exists two algorithm which are available for the EC:

- XOR-Algorithm
- Reed-Solomon-Algorithm

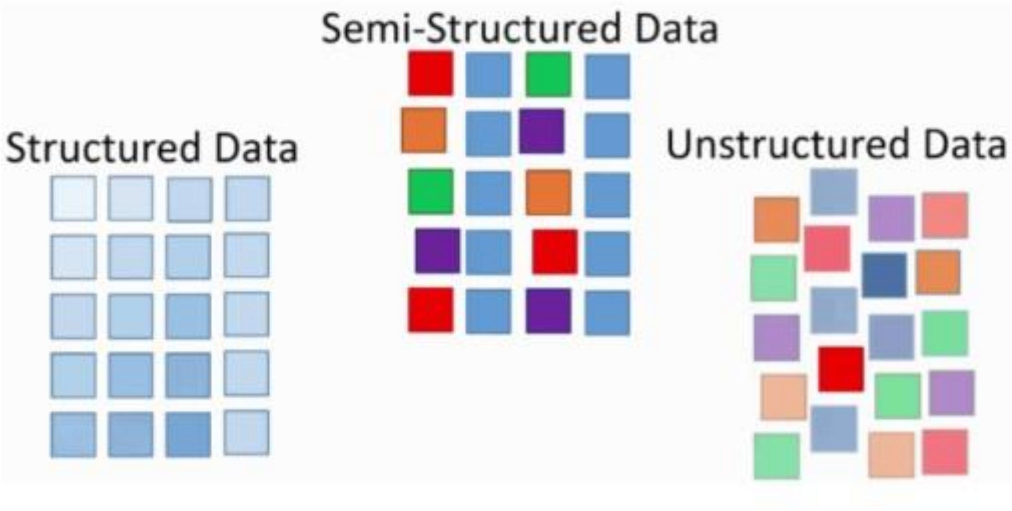
26. What will happen if the number of reducers be “0” in the Hadoop?

If the number of reducers is set to be “0”, then neither the reducer will be executed nor the aggregation will happen. Thus., in this case the “Map-only job” is preferred in Hadoop, where the map perform all the tasks with InputSplit and none of the job is done by the reducer. Here, Map output will be final output.

There is sort, key and shuffle phase between the map and reduce phases. Where the shuffle and the sort phases are responsible for the sorting of keys in an ascending order and then grouping the values of the same keys. However, we can avoid the reduce phase if it is not required here. The avoiding of reduce phase will eliminate the sorting and shuffling phases as well, which automatically saves the congestion in a network.

27. Differentiate structured, unstructured and semi-structured data.

Big-Data includes high velocity, huge volume and extensible data variety, which are of three types as: Structured Data, Unstructure Data, and Semi-Structured data.



Structured Data	Unstructured Data	Semi-Structured Data
The data is formatted in an organized way	Un-organized form of data	The data is being partially organized
Fixed data-scheme is used	There is an unknown schema	It lacks of the formal-structure of data-model
Based on the Relational data-base table	Based on the character and the binary data	Based on XML/RDF
The transaction of structured data is matured and various techniques of concurrency is also used.	There is neither transaction management nor concurrency.	The transaction here is basically adapted from the Database Management System which are not matured.
Version is over tuples and tables	Whole versioned	Version over the graph or tuple is possible here

As, Structured data is scheme dependent hence it is less flexible	Semi-Structured data is very flexible because of the absence of schema	More flexible than structured but less than that of unstructured data
It allows complex joining	Here only the textual queries are possible	It allows the queries over the same node
Difficulty in scaling of database schema	Very scalable	Scaling is simple in this as compared to the structured data
Example: Relational Data Base Management System data	Example: multimedia files	Example: XML & JSON files

28. If a file contains 100 billion URLs. Then how will you find the first unique URL?

This problem has the large set of data i.e., 100 billion URLs, so it has to be divided into the chunks which fits into the memory and then the chunks needs to be processed and then the results get combined in order to get a final answer. In this scenario, the file is divided in the smaller ones using uniformity in the hashing function which produces the N/M chunks, each is of M (i.e., size of main-memory). Here each URLs is read from an input file, and apply hash function to it in order to find the written chunk file and further append the file with the original line-numbers. Then each file is read from the memory and builds the hash-table for URLs which is used in order to count the occurrences of each of the URLs and then stores the line-number of each URL. After the hash-table built completely the lowest entry of the line-number having a count value of 1 is scanned, which is the first URL in the chunk file which is unique in itself. The above step is repeated for all the chunk files, and the line-number of each URL is compared after its processing. Hence, after the process of all the chunk-file, the 1st unique URL found out from all that processed input.

29. Differentiate the MapReduce InputSplit and the HDFS block?

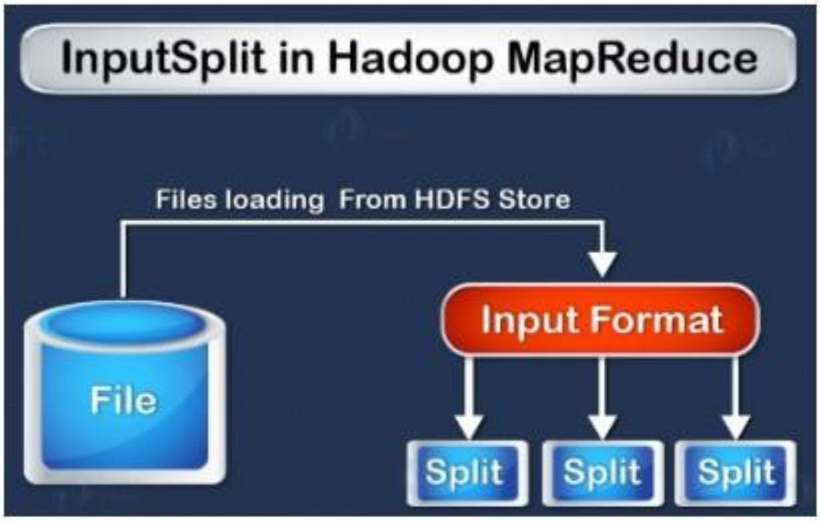


Fig: Hadoop MapReduce: InputSplit

Block:

- Block is contiguous location on hard-drive in which the HDFS data is stored. The FileSystem stores the data as the collection of blocks. Similarly, the HDFS store each of the file as a block and distribute it over Hadoop cluster.
- Block is data' physical representation.
- The default block-size of HDFS is of 128MB, which gets configured as per its requirement. Each block is of the same-size except the last one. The last-block can be either smaller or of same-size. In the Hadoop system, the file gets splits in 128MB of blocks and further store in the Hadoop FileSystem

InputSplit:

- InputSplit represent a data of individual Mapper to be processed. The splits are divided into records, where each of the record being processed by a map.
- InputSplits is the data' logical representation.
- By default, the size of split is approx. equal to the block size.

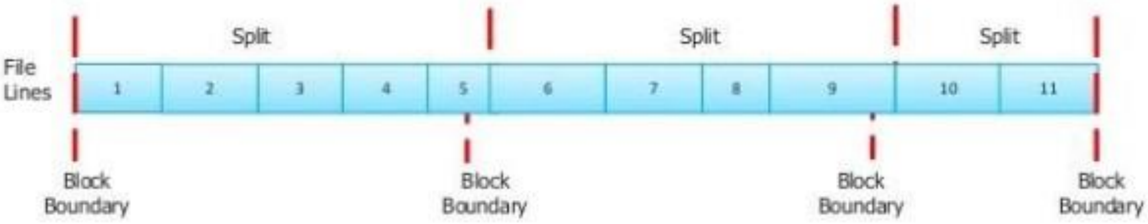


Fig: Relationship between HDFS blocks and InputSplits

- The logical record doesn’t fit neatly in HDFS blocks.
- The logical record is the line which crosses the blocks boundary.
- The first split contains five line although it gets spans over the blocks.

30. Can we use LIKE operator in Hive?

We can use LIKE operator, as the HIVE supports the LIKE operator. But the multivalued Like query is not supported in Hive like below:

*SELECT*FROM tablename WHERE firstname LIKE ANY ‘root~%’,’user~%’;*

Thus, one can easily use the LIKE operator whenever it is required in HIVE. In case if there is a need to use multivalued LIKE operator, we have break it, so as to work in HIKE.

WHERE tbl2.product LIKE concat(‘%’, tbl1.brand, ‘%’)

31. Differentiate the Static and the Dynamic Partition.

When the data is being inserted in the table, partitions gets created, which depends on how the data is loaded. Hence, it is the best performance-tuning technique.

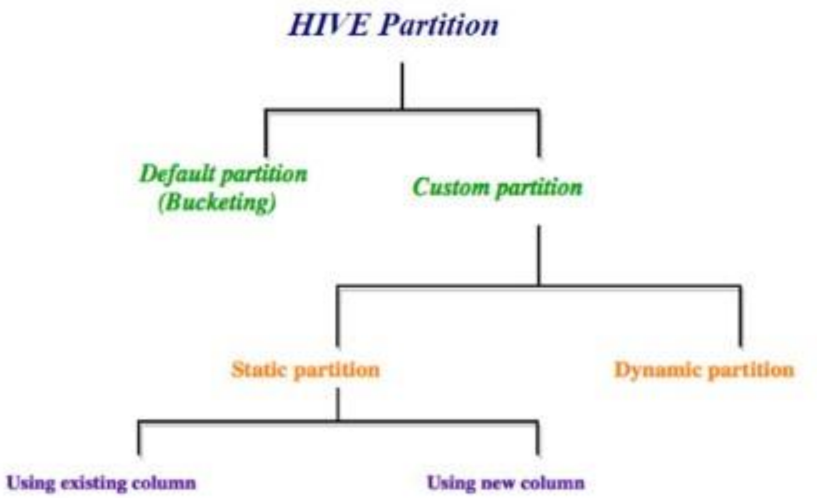


Fig: HIVE Partition

There are basically two types of partitions in Hive such as:

- Static Partition
- Dynamic Partition

Static partition:

When the big files are loaded into the HIVE tables, static partition is preferred, which saves our time of data loading as compared to the dynamic partition.

Here the partition columns are manually added and the files are moved in the partition-table manually.

One can get the name of partition column from the file-name without reading the file completely.

One has to specify the value of partition column for each load.

As, static partition is a default mode of the Hive, so one can find below property-set in the

`hive-site.xml`

```
set hive.mapred.mode=strict
```

Dynamic Partition:

Each of the data row which are available in a file are read and partitioned is done through the MapReduce job.

While doing the ETL jobs, the dynamic partition is done.

There is not any default mode in Hive, so one has to set following properties in the hive-

```
site.xml file.
```

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

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

In the dynamic partition, the values of the partition column have not been specified in every load statement. Thus, below are some steps which are used to create dynamic partition-table with the data.

- Create X, a non partition table and loads the data
- Create Y, a partition table for specifying the column partition.
- load the data from ‘X’ to ‘Y’ as:

hive>INSERT INTO TABLE Y PARTITIONstate SELECT*FROM X;

here partition column is last column of non-partitioned table.

32. List some industrial usage of Hadoop.

Hadoop is a way to handle structured and unstructured data. Here are few areas where hadoop and big data will be of good boost.

- Traffic flow management
- Processing inlet and outlet of streams
- CMS and emails

- Using hadoop computing cluster to analyze animal brain neurological signals
- Fraud detection and prevention
- Analyze click stream, transaction, video, social media data to project appropriate advertisement towards targeted audience
- Social media entities like content, posts, images, videos are handled well
- Improve business by analyzing customer data in real time
- Government agencies like intelligence, defense, cyber security, and scientific research
- Medical field data like medical devices, doctor's notes, imaging reports, lab results, clinical data and financial data

33. List the 5V's that are associated with the Big-Data.

While handling bulk data, we need to foresee situations related to processing the data. Following aspects helps us to describe the nature of big data.

- Volume – The size of the data may be in Petabytes or Exabytes. The exponential growth of the data justifies the voluminous data that gather over a period of time.
- Velocity – Rate of data growth. Data is accumulating from all kinds of source. For e.g., the data input from social media is huge in these days.
- Variety – The data is of different formats like video, audio, csv, word file, etc. This heterogeneity of data types brings in lots of challenge as well as benefits.
- Veracity – Incomplete or inconsistence data leads to uncertainty in the data. Accuracy, quality is difficult to manage as the data becomes big and of varied source. It becomes hard to trust.
- Value – It is difficult to acquire and handle big data. What is the benefit of going through this process? Is the big data adding any value to the business? Do we get good ROI, is the process profitable?

34. Explain how Hadoop is different from the Traditional Processing Systems using Relational Database Management?

Relational Database Management System	**Hadoop**

Relational Database Management System relies on structured data where the data scheme is known always.	In Hadoop, the data which is to be stored is of any kind i.e., either structured data, semi-structured data or unstructured data.
It provides no or limited processing capabilities	It allows the parallel distribution of data for processing.
Relational Database Management System based on the “schema-on-write” where the validation of schema has to be done before the loading of data.	Hadoop follows the read policy schema.
In Relational Database Management System, as the schema of data is known already thus there are faster reads.	In Hadoop none of the scheme validation exists during the HDFS write, hence writes are faster in this.
Online Transaction Processing (OLTP) is suitable.	Online Analytical Processing (OLAP) is suitable.
Computational speed is fast.	Computational speed is generally slower here.
It is a licensed software.	It is an open-source framework.

35. List the main configuration files of Hadoop.

Below are the main confirmation files of Hadoop:

- hadoop-env.sh
- core-site.xml
- hdfs-site.xml

- yarn-site.xml
- mapred-site.xml
- masters
- slaves

36. HDFS stores the data by using hardware commodity which has the higher chance of failure. Then explain how HDFS ensures the system’s capability in fault-tolerance?

Hadoop also creates a backup, which is termed as replication. Thus, in case of any failure also there should not be any mistake in Hadoop due to its replication.

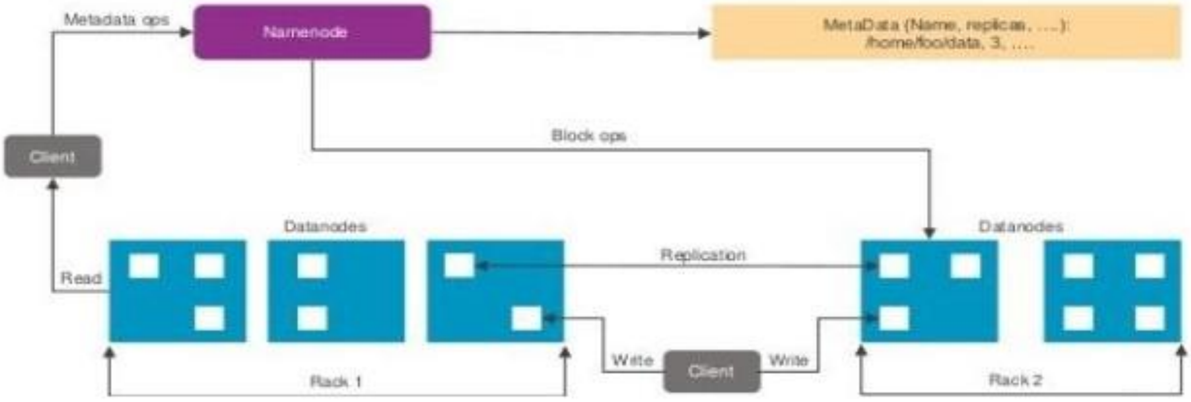


Fig: HDFS block replication

If HDFS stores the data by using hardware commodity which has the higher chance of failure. Then the HDFS ensures the system’s capability in fault-tolerance by the block replication.

- HDFS replicates blocks and then store it on different data-nodes.
- Here the default replication factor is 3.

37. What problem will arise if the HDFS has a lot of small files? Also provide the method in order to overcome this problem.

The problem with HDFS is that if we have lots of smaller files, then there are too many blocks for them. And for too many blocks, there exists too many metadata. Thus, in order to manage thus huge amount of metadata is very difficult.

However, we can overcome from this problem by using Hadoop Archive, where it clubs all the HDFS small files in a single archive file having .HAR extension

>hadoop archieve-archiveName myfilearchive.har /input/location /output/location

38. Suppose there is a file having size of 514MB is stored in the Hadoop (Hadoop 2.x) by using the default size-configuration of block and also by default replication-factor. Then, how many blocks will be created in total and what will be the size of each block?

The default replication factor is 3 and the default block-size is 128MB in Hadoop 2.x.

Thus, the 514MB of file can be split into:

a	b	c	d	e
128MB	128MB	128MB	128MB	2MB

- The default block size is 128MB
- Number of blocks: 514MB128MB=4.05 ≈5 blocks
- Replication factor: 3
- Total blocks: 5*3=15
- Total size: 514*3=1542MB

Hence, there are 15 blocks having size 1542MB.

39. How to copy a file into the HDFS having different block-size to that of the existing block-size configuration?

The copying of a file into the HDFS having different block-size to that of the existing block-size configuration can be done as:

- Block-size:

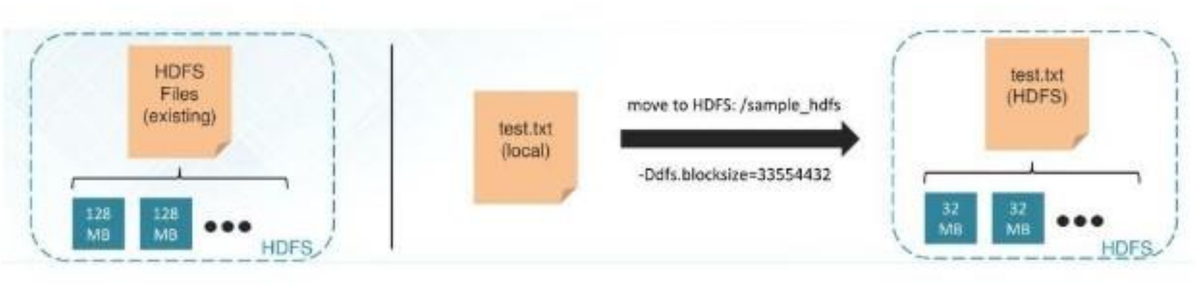
32MB=33554432 Bytes (Default block-size: 128MB)

- Command:

```
hadoop fs-Ddfs.blocksize=33554432-copyFromLocal /local/test.txt /sample_hdfs
```

- Checking of the block-size of test.txt:

```
hadoop fs-stat %o /sample_hdfs/test.txt
```



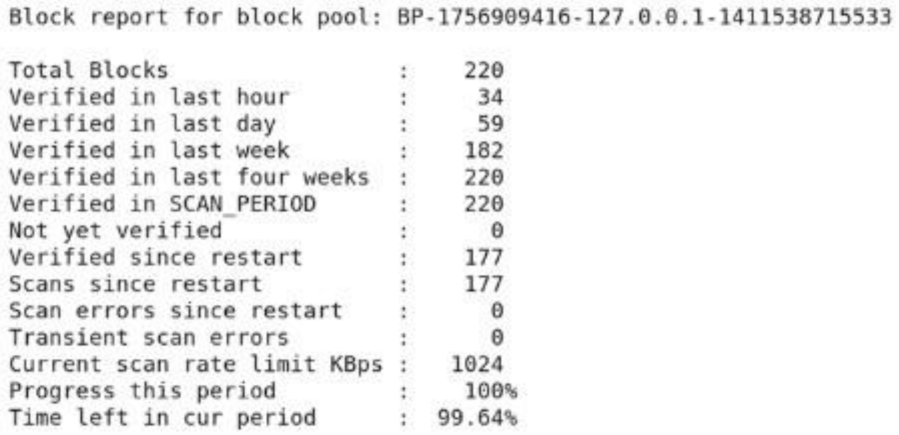
40. What do you mean by block scanner in the HDFS?

Block scanner in the HDFS:

- Block scanner basically maintains the integrity in data blocks.
- Periodically it runs over each data-node in order to verify that whether the data-blocks are correctly stored or not.

Below are the steps for this:

- The DataNode reports to the NameNode.
- The NameNode schedules the creation of new replica by using the good ones.
- Once the replication factor reaches the required level, the corrupted blocks will get deleted.



```
Block report for block pool: BP-1756909416-127.0.0.1-1411538715533

Total Blocks                  :    220
Verified in last hour         :     34
Verified in last day          :     59
Verified in last week         :    182
Verified in last four weeks   :    220
Verified in SCAN_PERIOD       :    220
Not yet verified              :      0
Verified since restart        :    177
Scans since restart           :    177
Scan errors since restart     :      0
Transient scan errors         :      0
Current scan rate limit KBps :   1024
Progress this period          :    100%
Time left in cur period       :  99.64%
```

Hadoop Interview Questions FAQS

Q: What is Hadoop used for?

A: Hadoop is an open-source software framework that is used for storing data and then running applications on groups of commodity hardware. Hadoop provides huge storage for any type of data, vast processing power, and the capability to handle virtually infinite concurrent tasks.

Q: How do I prepare for a big data interview?

A: Preparing for a big data interview may vary from one job to the other. However, some tips that can help you prepare well include having clear knowledge of the basics, knowing the audience, knowing your story, having the standard answers ready, asking good questions, taking a technical test, and practicing as much as you can.

Q: What is the Hadoop example?

A: The asset-intensive energy industry uses Hadoop-powered analytics for predictive maintenance. Input from the Internet of Things (IoT) devices serving data into big data programs is used. Financial services, retailers, telecommunication companies use Hadoop.

Q: Is Hadoop an ETL tool?

A: Hadoop is not an ETL tool. It rather is an ETL helper. This means it can help you to manage your ETL projects. ELT denotes Extract, Transform, and Load. ETL is suitable for dealing with smaller data sets that need difficult transformations.

Q: What are the key components of big data?

A: The core components of big data include ingestion, transformation, load, analysis, and consumption.

Q: What is SQL on Hadoop?

A: SQL-on-Hadoop is a class of methodical application tools that combine established SQL-style inquiring with the latest Hadoop data elements of the framework. While it supports familiar SQL queries, SQL-on-Hadoop enables a broader group of enterprise developers as well as business analysts that work with Hadoop on commodity computing groups.

Q: What type of system does Hadoop use?

A: Hadoop is an open-source, Java-based application of a clustered file system, which is called HDFS. This enables you to do cost-effective, dependable, and scalable distributed computing. The HDFS architecture is highly fault-tolerant and designed in a way that it can be used on low-cost hardware.

Q: How is Hadoop used in real life?

A: Hadoop in real life is used for Analyzing life-threatening risks, Identifying warning signs of security breaches, Preventing hardware failure, Understanding what people think about your company, Understanding when to sell certain products, Finding ideal prospects, and Gaining insight from your log files.