1) **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

2) What are the four characteristics of Big Data?

**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.

**Veracity**: biases, noise and abnormality in data

3) What does ‘jps’ command do?

It gives the status of the daemon’s which run Hadoop cluster. It gives the output mentioning the status of name node, data node , secondary name node, Job tracker and Task tracker.

4) Which are the three modes in which Hadoop can be run?

The three modes in which Hadoop can be run are −

1. standalone (local) mode
2. Pseudo-distributed mode
3. Fully distributed mode

5) How is analysis of Big Data 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.

6)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

7)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.

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 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.

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?

No, 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.

8)What is a Namenode?

Namenode is the master node on which job tracker runs and consists of the metadata. It maintains and manages the blocks which are present on the datanodes. It is a high-availability machine and single point of failure in HDFS.

9)Is Namenode also a commodity hardware?

No. Namenode can never be 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.

10)What is a Datanode?

Datanodes are the slaves which are deployed on each machine and provide the actual storage. These are responsible for serving read and write requests for the clients.

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

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

11)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.

12)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.

13)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.

14)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. 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.

15)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?

16)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.

17)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.

What is the communication channel between client and namenode/datanode?

The mode of communication is SSH.

18)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.

19)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.

20)Explain how do ‘map’ and ‘reduce’ works.

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.

21)Why ‘Reading‘ is done in parallel and ‘Writing‘ is not in HDFS?

Through mapreduce program the file can be read by splitting its blocks when reading. But while writing as the incoming values are not yet known to the system mapreduce cannot be applied and no parallel writing is possible.

Copy a directory from one node in the cluster to another

Use ‘-distcp’ command to copy,

Default replication factor to a file is 3.

Use ‘-setrep’ command to change replication factor of a file to 2.

hadoop fs -setrep -w 2 apache\_hadoop/sample.txt

22)What is rack awareness?

Rack awareness is the way in which the namenode decides how to place blocks based on the rack definitions Hadoop will try to minimize the network traffic between datanodes within the same rack and will only contact remote racks if it has to. The namenode is able to control this due to rack awareness.

23)Which file does the Hadoop-core configuration?

core-default.xml

24)Is there a hdfs command to see available free space in hdfs

hadoop dfsadmin -report

The requirement is to add a new data node to a running Hadoop cluster; how do I start services on just one data node?

You do not need to shutdown and/or restart the entire cluster in this case.

First, add the new node's DNS name to the conf/slaves file on the master node.

Then log in to the new slave node and execute −

$ cd path/to/hadoop

$ bin/hadoop-daemon.sh start datanode

$ bin/hadoop-daemon.sh start tasktracker

then issuehadoop dfsadmin -refreshNodes and hadoop mradmin -refreshNodes so that the NameNode and JobTracker know of the additional node that has been added.

25)How do you gracefully stop a running job?

Hadoop job –kill jobid

26)Does the name-node stay in safe mode till all under-replicated files are fully replicated?

No. During safe mode replication of blocks is prohibited. The name-node awaits when all or majority of data-nodes report their blocks.

What happens if one Hadoop client renames a file or a directory containing this file while another client is still writing into it?

A file will appear in the name space as soon as it is created. If a writer is writing to a file and another client renames either the file itself or any of its path components, then the original writer will get an IOException either when it finishes writing to the current block or when it closes the file.

27)How to make a large cluster smaller by taking out some of the nodes?

Hadoop offers the decommission feature to retire a set of existing data-nodes. The nodes to be retired should be included into the *exclude file*, and the exclude file name should be specified as a configuration parameter **dfs.hosts.exclude**.

The decommission process can be terminated at any time by editing the configuration or the exclude files and repeating the **-refreshNodes** command

Can we search for files using wildcards?

Yes. For example, to list all the files which begin with the letter a, you could use the ls command with the \* wildcard &minu;

**hdfs dfs –ls a\***

28)What happens when two clients try to write into the same HDFS file?

HDFS supports exclusive writes only.

When the first client contacts the name-node to open the file for writing, the name-node grants a lease to the client to create this file. When the second client tries to open the same file for writing, the name-node will see that the lease for the file is already granted to another client, and will reject the open request for the second client

What does "file could only be replicated to 0 nodes, instead of 1" mean?

The namenode does not have any available DataNodes.

29)What is a Combiner?

The Combiner is a ‘mini-reduce’ process which operates only on data generated by a mapper. The Combiner will receive as input all data emitted by the Mapper instances on a given node. The output from the Combiner is then sent to the Reducers, instead of the output from the Mappers

Consider case scenario: In M/R system, - HDFS block size is 64 MB

- Input format is FileInputFormat

– We have 3 files of size 64K, 65Mb and 127Mb

30)How many input splits will be made by Hadoop framework?

Hadoop will make 5 splits as follows −

* - 1 split for 64K files
* - 2 splits for 65MB files
* - 2 splits for 127MB files

31)Suppose Hadoop spawned 100 tasks for a job and one of the task failed. What will Hadoop do?

It will restart the task again on some other TaskTracker and only if the task fails more than four ( the default setting and can be changed) times will it kill the job.

32)What are Problems with small files and HDFS?

HDFS is not good at handling large number of small files. Because every file, directory and block in HDFS is represented as an object in the namenode’s memory, each of which occupies approx 150 bytes So 10 million files, each using a block, would use about 3 gigabytes of memory. when we go for a billion files the memory requirement in namenode cannot be met.

33)What is speculative execution in Hadoop?

If a node appears to be running slow, the master node can redundantly execute another instance of the same task and first output will be taken .this process is called as Speculative execution.

34)Can Hadoop handle streaming data?

Yes, through Technologies like Apache Kafka, Apache Flume, and Apache Spark it is possible to do large-scale streaming.

35)Why is Checkpointing Important in Hadoop?

As more and more files are added the namenode creates large edit logs. Which can substantially delay NameNode startup as the NameNode reapplies all the edits. Checkpointing is a process that takes an fsimage and edit log and compacts them into a new fsimage. This way, instead of replaying a potentially unbounded edit log, the NameNode can load the final in-memory state directly from the fsimage. This is a far more efficient operation and reduces NameNode startup time.