1. What is the difference between a Namenode and a Secondary Namenode?

Answer 1:

NameNode:- Its is Also Known as Master node in Hadoop.

1. It manages all The file systems and Nodes.
2. It handles all the file system tasks like opening a file, closing a file, remaining a file.
3. It provides replication to the data nodes like how much copy of data should be made to keep the data safe and prevent it from loss and makes processes flawlessly. Normally Up to 3 replication copies are made is a database.
4. It keeps the track of process and that is also called as checkpointing.
5. It manages the load among all the data nodes.

Secondary Name Node:-It is pretty same a NameNode but it keeps synchronising with NameNode.

1. It Sync the checkpoint periodically that if Main namenode will be fail at some point the process should be lost and can be continued from the point where the last checkpoint is made.
2. It updates the FSIimages continuously to prevent the backup of processed.

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

Answer2:

If one Task from hundreds of task fails, Hadoop will reschedule the specific task and look for the replacement node on which the task can perform.

It will try up to generally four time (which is the default configuration) And if the task fails Even the 4th time then it will return failure.

3.What is JobTracker? What are some typical functions of Job Tracker?

Answer3:

Job Tracker:- It is the service in Hadoop associated with the “mapreduce”. It performs specific mapreduce tasks Like pointing out the data node and much more.

1. When client submits the tasks to job tracker, The job tracker will communicate with the Namenode to determine where the data node,s location is.
2. Then job tracker locates the task tracker that contains the data.
3. And job is submitted to task tracker, and wait for the reponse.
4. If the task tracker replies with some failure like the node is dead or not responding to the query.. then job tracker will search for replica of data node and mark the previous one as failed or sometimes even back list it.
5. After getting the job done task tracker informs the job tracker with the information.
6. After all the steps Done the jobtracker updates the status to client.
7. Now the client can pull the information.
8. IF the job tracker is point of failure in mapreduce then the whole system will goes down, and all the running processes are halted.

4. Consider case scenario: In M/R system,
- HDFS block size is 128 MB
- Input format is FileInputFormat
We have 3 files of size 128K, 129MBand 255MB. How many input splits will be made by Hadoop framework?

Answer4:

1. One split for 128k~0.125mb
2. 2 splits for 129mb i.e 128 mb for 1st split and 1 mb for remaining filesize.
3. 2 split for 255mb i.e 128mb for 1st part of file and 127 mb for the remaining file size.

5. What is the meaning of speculative execution in Hadoop? Why is it important?

Answer5:

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. Tehre 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 task 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 remaining tasks (which are slower) across the nodes that are free currently. When these tasks finish, 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.

6. How is node failure detected by the Namenode?

Answer6:

A DataNode is considered dead after a set period without any heartbeats (10.5 minutes by default).

When this happens, the NameNode performs the following actions to maintain the configured replication factor (3x replication by default):

1. The NameNode determines which blocks were on the failed DataNode.
2. The NameNode locates other DataNodes with copies of these blocks.
3. The DataNodes with block copies are instructed to copy those blocks to other DataNodes to maintain the configured replication factor.
4. If a DataNode fails to heartbeat for reasons other than disk failure, it needs to be recommissioned to be added back to the cluster.
5. If a DataNode rejoins the cluster, there is a possibility for surplus replicas of blocks that were on that DataNode. The NameNode will randomly remove excess replicas adhering to Rack-Awareness policies.

7.How are under-replicated blocks identified by the Namenode? How are these blocks replicated back?

Answer7:

Under-replicated blocks These are blocks that do not meet their target replication for the file they belong to. HDFS will automatically create new replicas of under-replicated blocks until they meet the target replication.

8.After the Map phase finishes, the Hadoop framework does “Partitioning, Shuffling and Sorting”. Explain what happens in these phases

Answer8:

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.

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.