

1. For NameNode, why it's not necessary to store block locations persistently?

For NameNode it is not necessary to store block locations persistently because the information on block locations can be dynamically computed from the block report periodically sent by the DataNodes to the NameNode.

2. Why is it important to make the NameNode resilient to failures?

It is important to make the NameNode resilient to failures because unlike DataNodes NameNode is one of the single point of failure (SPOF) in Hadoop (1.x). There is too much dependency on NameNode so when the NameNode is down the Hadoop cluster is unusable.

3. What details are there in the FsImage file?

FsImage file is the point in time snapshot of the Hadoop metadata which includes all the information including list of files and path, size, owner, permissions, list of blocks of files, list of DataNodes, block size, replication factor.

4. What is the purpose of the secondary name-node?

The function of Secondary NameNode is to merge the FsImage and the edit log file in NameNode and create and copy back new FsImage to the NameNode periodically. Secondary NameNode is not a standby for NameNode. This is necessary because if there are more edits in the cluster then the size of edit log grows much bigger.

5. Does the NameNode stay in the safe mode until all under-replicated files are fully replicated? Why or why not?

No, the NameNode does not stay in safe mode until all under-replicated files are fully replicated because safe mode for the NameNode is read-only mode for the Hadoop Cluster and the modification of file system or blocks are not allowed in this mode rather the NameNode checks in the safely replicated blocks in all data nodes and exits safe mode, then the name node makes list of the blocks that need to be replicated and starts replicating blocks to data nodes.

6. What are the core changes in Hadoop 2.x compared to Hadoop 1.x? In other words, state the major differences between Hadoop 1 and Hadoop 2.

The major differences between Hadoop1 and Hadoop2 are illustrated in a table below:

BDT- Assignment-2 (Sujiv Shrestha 610145)

Hadoop 1.x	Hadoop 2.x
i. It has single NameNode (master) in the master-slave architecture making the NameNode single point of failure (SPOF).	i. It has Active and Hot Standby NameNode (master) in the master-slave architecture achieving High Availability.
ii. It also has single job tracker which may get overloaded at high processing load and can be single point of failure (SPOF) as it has to manage resources as well as application processing.	ii. The resource management is done by the YARN which only performs resource management task and the application processing is managed/monitored by Application Master which is performed by one of the DataNode for each application.
iii. Only MapReduce jobs can be deployed in Hadoop 1.x cluster. No multitenancy.	iii. Uses MapReduce v2 as a user library and YARN also supports non-MapReduce type of processing (multitenancy).
iv. It uses static Map and Reduce slots for allocating Resources separately for map and reduce tasks which prevails the efficient use of resources.	iv. It uses variable-sized containers that can carry out map, reduce or any other tasks resulting high cluster utilization.
v. It has low scalability supporting around 4000 nodes in a cluster.	v. It has relatively high scalability supporting more than 10,000 nodes in a cluster.
vi. Depends too much on NameNode and block storage are tightly coupled to Namespace. This also results in lack of isolation between application environment.	vi. HDFS Federation partitions the filesystem namespace over multiple NameNodes removing tight coupling of block storage and Namespace.
vii. It does not support cross platform applications.	vii. It supports applications written beyond java.

7. What is the difference between MR1 in Hadoop 1.0 and MR2 in Hadoop2.0?

In MR1 the job tracker has to do both resource management as well as job scheduling and monitoring and the Hadoop Cluster was able to perform only map-reduce jobs, while in MR2 in Hadoop 2.0 the YARN does the resource management job and the ApplicationMaster schedules and monitors life cycle of map-reduce as well as other applications.

8. What is HDFS Federation? What advantage does it provide?

To eliminate the limited NameSpace scalability and lack of isolation Hadoop 2.0 has additional feature for adding multiple NameNodes in the cluster and the filesystem NameSpace is partitioned over different NameNodes which do not communicate between themselves and failure of one NameNode does not effect the other. This architecture is referred as HDFS Federation.

The advantage of HDFS Federation are:

- i. The NameNode which is overloaded in Hadoop 1.0 has efficient load sharing and NameNode in Hadoop2.0 is no more SPOF and has High Availability.
- ii. There is better isolation in application environment as the NameNodes do not communicate within themselves.
- iii. The tight coupling of block storage and NameSpace is removed by use of block pools in HDFS Federation.
- iv. It has better scalability.

9. What is NameNode High Availability and how is it achieved in Hadoop 2?

NameNode High Availability is one of the feature of Hadoop2 that increases the reliability of Hadoop cluster eliminating the dependency on single NameNode in the cluster (SPOF) by adding Hot Standby NameNode which provide a fast recovery of the service by replacing the Active NameNode at time of failure.

This is achieved in Hadoop 2 by having Active and Standby NameNodes in same cluster. The standby NameNode constantly reads the shared storage directory with the Active NameNode and maintains any edits in its namespace written by Active NameNodes so as to ensure it has read all of the edits from the shared storage before it can replace the Active NameNode when required.

10. What is the role of Application Master in YARN application execution?

The role of Application Master in YARN application execution is to coordinate the sub-task of applications like map-reduce and other by scheduling and monitoring the execution of the tasks within its application. Each application has one Application Master which runs in the container in DataNode and its function is to maintain the life cycle of the tasks of application and ask resource containers for it.