Name: Biniam Arefaine

ID: 110972

---

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

Because Block location information's is re-constructed dynamically from the Data Nodes when they send block reports to the Name Node; when they join the cluster and periodically afterward to ensure the Name node's block mapping is up to date

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

Because without the Name node, in HDFS which consists of a single Name node in Hadoop 1, the HDFS cannot be used at all. This is because there is no way we can know which files are stored in which data nodes.

3. What details are there in the FsImage file?

FsImage consists of:

- Point in time snapshot, a complete persistent checkpoint of the file system metadata
- Information's such as file's replications, block size, a serialized form of a ll directories and etc., which are complete state of file system at a point in time.

4. What is the purpose of the Secondary NameNode?

Secondary Name node functions are it:

- Performs housekeeping functions for the NameNode.
- merges the FsImage and the edits log file periodically which by default is every hour and keeps edits log size within a limit.
- to a checkpoint in HDFS. It's just a helper node for Namenode.

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

No.
Because replication of data blocks does not occur in safemode. After the NameNode loads the file system state after start-up, it waits for DataNodes to report their blocks so that it does not prematurely start replicating the blocks. To replicate them, the name node must leave safemode.

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 difference are as follows:

	HADOOP1	HADOOP2
1	Has Single-Point-of-Failure (SPOF) – because of single Namenode- and in the case of Namenode failure, needs manual intervention to overcome.	Has feature to overcome SPOF with a standby Namenode and in the case of Namenode failure, it is configured for automatic recovery.
2	MR does both processing and cluster-resource management.	YARN (Yet Another Resource Negotiator) does cluster resource management and processing is done using different processing models.
3	Only in Java	Goes beyond java
4	Has limited scaling of nodes.	Has better scalability.
5	A single Namenode to manage the entire namespace.	Multiple Namenode servers manage multiple namespaces.

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

	MR1	MR2
1	Scales only up to 4000 nodes	Can scale up to 10000 nodes per cluster
2	Uses Job tracker	has a Resource Manager for each cluster, and each data node runs a Node Manager

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

HDFS Federation is the way of creating and maintaining more than one NameNode independent of each other in a Hadoop cluster. HDFS consists of two parts, NameSpace and Block Storage. NameSpace resides in NameNode and is responsible for file handling operations. It also stores metadata about the file system. Federation allows HDFS metadata to be shared across multiple NameNodes, which aides with HDFS scalability and also provides data isolation, allowing different applications or teams to run their own NameNodes without fear of impacting other NameNodes on the same cluster.

Advantages it provides are as follows:

- *Namespace Scalability*: Federation adds namespace horizontal scaling. Large deployments or deployments using lot of small files benefit from namespace scaling by allowing more NameNodes to be added to the cluster.

- *Performance:* File system throughput is not limited by a single NameNode. Adding more NameNodes to the cluster scales the file system read/write throughput.

- *Isolation*: A single NameNode does not offer namespace isolation in a multi-user environment. By using multiple NameNodes, different categories of applications and users can be isolated to different namespaces.

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

The High Availability (HA) feature in Hadoop 2 addresses the NameNode SPOF problem by providing the option of running two redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This eliminates the NameNode as a potential single point of failure (SPOF) in an HDFS cluster.

A. Availability if DataNode fails

- In HDFS, replicas of files are stored on different nodes.
- DataNodes in HDFS continuously sends heartbeat messages to NameNode every 3 seconds by default.
- If NameNode does not receive a heartbeat from DataNode within a specified time, the NameNode considers the DataNode to be dead.
- NameNode then checks for the data in DataNode and initiates data replication. NameNode instructs the DataNodes containing a copy of that data to replicate that data on other DataNodes.
- Whenever a user requests to access his data, NameNode provides the IP of the closest DataNode containing user data. Meanwhile, if DataNode fails, the NameNode redirects the

user to the other DataNode containing a copy of the same data. The user requesting for data read, access the data from other DataNodes containing a copy of data, without any downtime. Thus cluster is available to the user even if any of the DataNodes fails.

B. Availability if NameNode fails

NameNode is the only node that knows the list of files and directories in a Hadoop cluster. "The filesystem cannot be used without NameNode".

The addition of the High Availability feature in Hadoop 2 provides a fast failover to the Hadoop cluster. The Hadoop HA cluster consists of two NameNodes (or more after Hadoop 3) running in a cluster in an active/passive configuration with a hot standby. So, if an active node fails, then a passive node becomes the active NameNode, takes the responsibility, and serves the client request. This allows for the fast failover to the new machine even if the machine crashes. Thus, data is available and accessible to the user even if the NameNode itself goes down.

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

The Application Master is responsible for the execution of a single application. It asks for containers from the Resource Scheduler (Resource Manager) and executes specific programs on the obtained containers. The Application Master knows the application logic and thus it is framework specific. The MapReduce framework provides its own implementation of an Application Master.