1. Explain what is High availability of Namenode

2. Explain what is check pointing and how it is useful

3. Explain what is HDFS federation

4. What are the configuration files that are to be edited for sure while installing a hadoop cluster.

Ans 1) Hadoop 2.0 High Availability feature that brings in an extra NameNode (Passive Standby NameNode) to the Hadoop Architecture which is configured for automatic failover.

The main motive of the Hadoop 2.0 High Availability project is to render availability to big data applications 24/7 by deploying 2  Hadoop NameNodes –One in active configuration and the other is the Standby Node in passive configuration.

Earlier there was one Hadoop NameNode for maintaining the tree hierarchy of the HDFS files and tracking the data storage in the cluster. Hadoop 2.0 High Availability allows users to configure Hadoop clusters with uncalled- for NameNodes so as to eliminate the probability of SPOF in a given Hadoop cluster. The Hadoop Configuration capability allows users to build clusters horizontally with several NameNodes which can operate autonomously through a common data storage pool, thereby, offering better computing scalability when compared to Hadoop 1.0

With Hadoop 2.0, Hadoop architecture is now configured in a manner that it supports automated failover with complete stack resiliency and a hot Standby NameNode.

Hadoop 2.2 has two Namenodes- Active Namenode and Passive Namenode.

1. **Hadoop High Availability feature tackles the namenode failure problem only for the MapReduce component in the hadoop stack.**

**ii) Hadoop High Availability feature supports only single Namenode within a Hadoop cluster.**

**iii)Hadoop High Availability feature tackles the namenode failure problem for all the components in the hadoop stack.**

From the above graph, it is evident that both the active and passive (Standby) NameNodes have state-of-the-art metadata that ensures flawless failover for large Hadoop clusters indicating that there would not be any downtime for your Hadoop cluster and it will be available all the time.

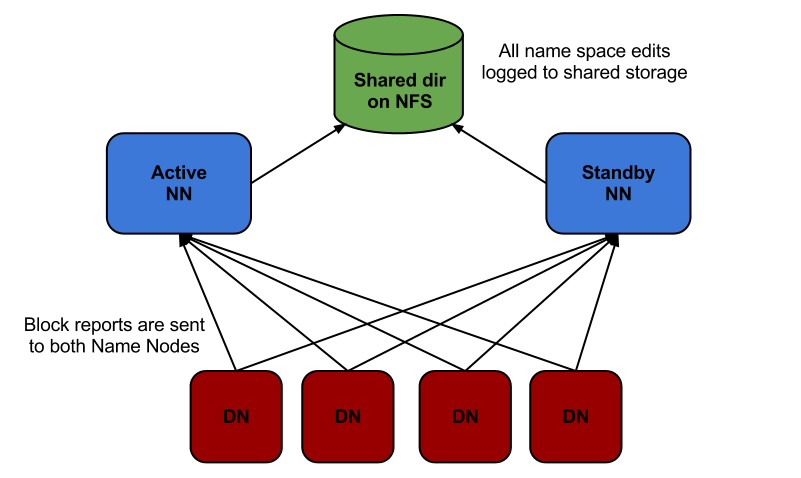
Hadoop 2.0 is keyed up to identify any failure~~s~~ in NameNode host and processes, so that it can automatically switch to the passive NameNode i.e. the Standby Node to ensure high availability of the HDFS services to the Big Data applications. With the advent of Hadoop 2.0 HA it’s time for Hadoop Administrators to take a breather, as this process does not require manual intervention.

With HDP 2.0 High Availability, the complete Hadoop Stack i.e. HBase, Pig, Hive, MapReduce, Oozie are equipped to tackle the NameNode failure problem- without having to lose the job progress or any related data. Thus, any critical long running jobs that are scheduled to be completed at a specific time will not be affected by the NameNode failure.

ANS 2)

Fsimage contains the HDFS metadata and edits contain the changes in the HDFS metadata. To get the exact view of the HDFS structure, both must be merged. At the time when a NameNode restarts, it merges both of them, as it needs to store this information in memory. Since edits can become very large with time, a NameNode’s startup time can become very lengthy. So, it is good to have them merged periodically.

• When any operation takes place in HDFS, the directory structure gets modified • These modifications are stored in the memory as well as in the edits files (edits files are stored on the hard disk) • If the existing fsimage file gets merged with edits, we’ll get an updated fsimage file • This process is called “checkpointing” and is carried out by the Secondary NameNode. It takes the fsimage and edits files from the NameNode and returns updated fsimage file after merging.

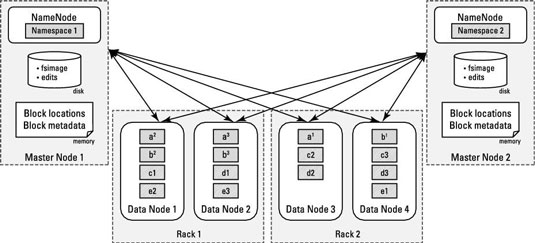
****

**ANS 3)**

The solution to expanding Hadoop clusters indefinitely is to *federate* the NameNode. Before Hadoop 2 entered the scene, Hadoop clusters had to live with the fact that NameNode placed limits on the degree to which they could scale. Few clusters were able to scale beyond 3,000 or 4,000 nodes.

NameNode’s need to maintain records for every block of data stored in the cluster turned out to be the most significant factor restricting greater cluster growth. When you have too many blocks, it becomes increasingly difficult for the NameNode to scale up as the Hadoop cluster scales out.

Specifically, you must set HDFS up so that you have multiple NameNode instances running on their own, dedicated master nodes and then making each NameNode responsible only for the file blocks in its own name space.



The figure shows replication patterns of data blocks in HDFS. You can see a Hadoop cluster with two NameNodes serving a single cluster. The slave nodes all contain blocks from both name spaces.

Ans 4)

The four files that need to be configured explicitly while setting up a single node hadoop cluster are:

* Core-site.xml
* HDFS-site.xml
* YARN-site.xml
* Xml