# Session3-Assign5

**1. Explain what is High availability of Namenode**

HDFS in Hadoop 1.x were faced mainly by two issues-

1. Name Node memory constraint

2. Name node being single point of failure

The issue 2- Name node was turning out to be a single point of failure in Hadoop 1.x. If name node goes down the entire cluster would have come to a stand still. A Hadoop admin intervention is required, which would mean a cluster downtime of at least 45 minutes to 1 hour.

This issue ways addressed in Hadoop 2.x by HDFS as High Availability.

While processing the data it was released that the processing daemon was overloaded. It was doing too many things like job application, job scheduling and monitoring, resource allocation. So it becomes overloaded because of which it may face a downtime.

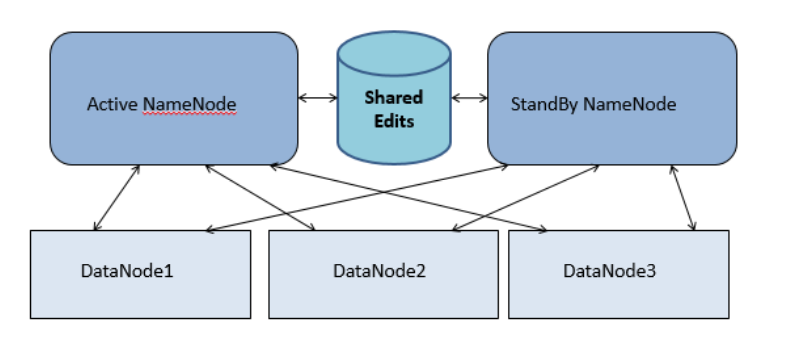
In High Availability systems there is a pair of Name nodes in an active-standby configuration.

In the event of failure of an active Name node, the standby takes over its duties without a significant interruption.

1. The name node must use highly-available shared storage to share the edit log. Edit logs are read by the Standby Name node when it takes the responsibility of the Active Name node.
2. Data Nodes must send block reports to both the Name nodes because of the block mappings.
3. The secondary name node’s role is subsumed by the standby name node. Standby name node takes periodic check points of the active name node.
4. Checkpoiniting is done by the standby name node.

Each DataNode reports block stored by it to both the NameNodes.

High Availability ca be represented as



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

Checkpointing is an essential part of maintaining file system meta data id HDFS. Filesystem meta data is stored in two different constructs- Edit Logs and FSImage.

Edit logs maintains transactional data, and keeps changing on live cluster.

FSImage maintains the merged data not changing on live cluster.

Need of checkpoints-

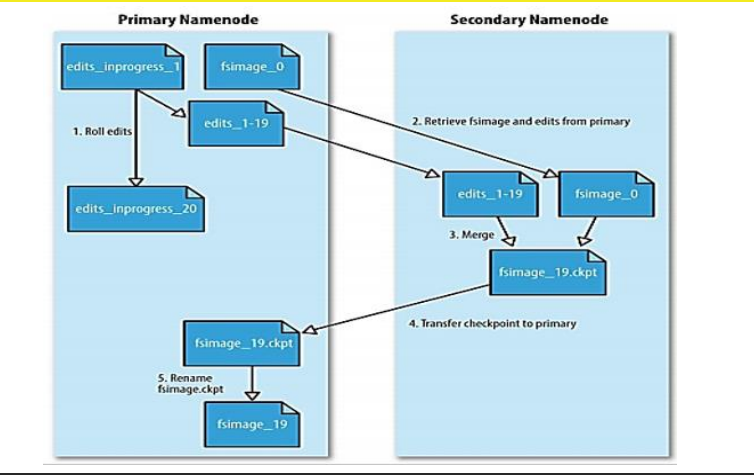
As the edit logs grows in size a couple of problems arises like it can fill up the disk space on a node and can also delay Name node startup as the name node reapplies all the edits. Here we need checkpointing.

Check pointing is the process of merging edit logs explicitly with FSImage. Check pointing takes edit logs and FSImage and merge them to on a new latest copy of FSImage. Name node can load this final in memory state directly from the FSImage.

This is a far more efficient operation and reduces Name Node startup time.

Check pointing can be done with High Availability name node also. The active and standby name nodes have a shared storage where edits are stored. The standby name node maintains a relatively up to date version of the name space by periodically replaying the new edits written to the share edits by name node.

Checkpointing is represented by-



**3. Explain what is HDFS federation.**

**Problem is Hadoop 1.x-**

HDFS in Hadoop comes with some limitations. One issue with HDFS is Name node memory constraint. The amount of meta data that name node can store is restricted to the memory available to name node daemon in Hadoop 1.x.

As of Hadoop 1.x the above limitations meant that the maximum amount of meta data the name node can store is limited to name node’s RAM.

Also number of nodes that a Hadoop 1.x cluster could manage stable was around 4000 node/slaves.

This was addressed in Hadoop 2.x with HDFS Federation.

**Solution-**

Problem to this problem is HDFS Federation, it is multiple name nodes with separate name spaces.

In the HDFS Federation, there are multiple NameNodes, each storing the metadata and block mapping of files and directories contained in particular sub-directories.

The list of sub-directories managed by a NameNode is called a namespace volume.

Blocks for files belonging to a namespace is called a block pool. For example, here we can have two name nodes, one for storing the metadata and block mapping for namespace volume /usr and one for /share. This way, if one Name Node fails, a namespace volume managed by another name node is still accessible. So the entire cluster doesn’t become inaccessible.

Let, Sub-directories and files inside /usr constitute NameSpace Volume 1

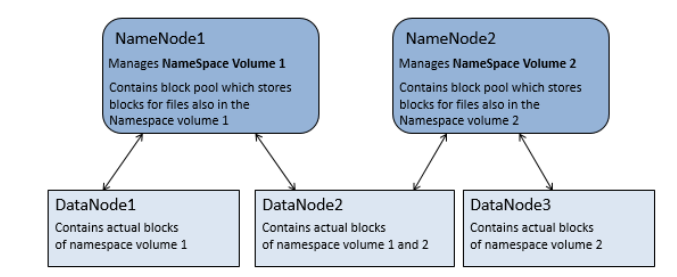
• Sub-directories and files inside /share constitute NameSpace Volume 2

• One DataNode can contain blocks of different namespace volumes

• So, namespace volumes are divided among NameNodes, but not among DataNodes.

**Key Benefits of HDFS Federation-**

1. Scalability and Isolation- Support for multiple namenodes horizontally scales the file system namespace. It separates namespace volumes for users and categories of applications and improves isolation.
2. Generic Storage Services- Block pool opens up the architecture for future innovation. New file systems can be built on top of block storage. New block pool categories are also possible, different from the default block pool.
3. Simple Design- Each name node is built to be robust. Federation is also backward compatible. It easily integrates with the existing single node deployments which work without any configuration changes.



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

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

1. Core-Site.xml
2. HDFS-site.xml
3. YARN-site.xml
4. xml
5. Core-Site.xml- Some important properties for setting Core-site.xml are-

* Configuring the name node address
* Configuring the rack awareness factor
* Selecting the type of security

1. HDFS-site.xml- Overriding the default parameters optimizes the cluster, improves performance and lets one know about the internal working of Hadoop ecosystem.

Settings to be done in configuring HDFS-site.xml-

* Configure port access.
* Manages ssl client authentication
* Controls Network interface
* Changes file permission

1. YARN-site.xml- YARN has Resource manager settings which effects resource allocation with node manager application manager. Its important properties are-

* WebAppProxy Configuration
* Map Reduce Configuration
* Node Manager Configuration
* Resource Manager Configuration
* IPC Configuration

1. Xml- We can specify the new value with tags like <property>,<name>,<description>,<final> etc inside predefined <configuration> tag. As Hadoop is an open source framework so the owners have provided option to override some features attribute inside various site.xml files.