Components of Hadoop:

\*NameNode:

->Contains details about which blocks are stored in which Data node.

->Contains the Hadoop FileSystem Tree and other metadata information about files and directories.

->It contains two important file in its hard disk such as

I)FsImage(File System Image)

II)Edits

1. fsimage

(file system image)

\*It contains the complete state of the file system at a point in time.

\*It represents state of the file system after all modifications are made up to specific transaction ID.

It contains:

•All directory structure of HDFS.

•Replication level of file.

•Modification and access times of files.

•Access permissions of files and directories.

•Block size of files.

•The blocks constituting a file.

2.Edits:

\*An edits file is a log that lists each file system change such as file creation, deletion or modification that was made after the most recent fsimage.

\*When any write operation takes place in HDFS, the directory structure gets modified.

\*This modified structure is stored in the edit .

\*By merging edits with the existing fsimage we will get updated fsimage.

\*We merge it because 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.

\*Secondary Namenode:

->It helps name node by managing activities like periodic merging of namespace and edits.

->But this is not a back up for name node.

->Periodically read the filesystem changes log and apply them into the fsimage file, thus bringing it up to date. This allows the namenode to start up faster next time.

->Secondary NameNode merges fsimage and edit files.

->This process is called Checkpointing and is carried out by the Secondary NameNode. It takes fsimage and edits files from NameNode and returns updated fsimagefile after merging.

\*DataNode:

->It actually stores data blocks of file in hdfs.

->Sends signals to NameNode periodically to verify if it is active.

->If datanode doesn’t send any signal namenode will consider it as inactive and it will stop any further communication with that particular node.

->They perform all the block operations, including periodic the Checksum.

->They receive instructions from the NameNode of where to put the blocks and how to put the blocks.

\*JobTracker:

->Controls overall execution of map reduce jobs.

->It process runs on a separate node and not usually on a DataNode.

->It finds the best TaskTracker nodes to execute tasks based on the data locality.

->It talks to the NameNode to determine the location of the data.

\*TaskTracker:

->Runs individual map reduce jobs on DataNodes.

->Periodically communicates with the JobTracker to give updates and receive instructions.