Hadoop Components

Hadoop is addressed to solve the problem of handling large volumes and variety of data. It is based on the distributed file system architecture. The core concept is to divide the data into tiny chunks and distribute them across cluster nodes.

Hadoop Distributed file system

This file system inspired by Google File System (GFS), abstracts the complex details of storage. The data is not confined usually to one machine. The motivation behind HDFS is

* We are dealing with large volume of data, just cannot store everything on one disk
* Data does not change much or frequently. Streaming data access – written once, read multiple times
* Economically feasible and reliable

The base idea is

* Divide data into blocks of defined sizes (default size 128MB)
* Store replicas of blocks in different nodes
* Access in parallel may be 100s of machines
* Process chunks of data in parallel and then combine results

HDFS comprises of 2 main components – NameNode, DataNode, Secondary NameNode.

NameNode

* the master node for keeping a track of the storage
* keeps track of the metadata of files stored on the various datanodes.
* permission, block info, filesystem tree, file size, replication factor etc
* Also in-memory mapping of which blocks are stored on which datanode.
* The NameNode executes file system namespace operations like opening, closing, and renaming files and directories.
* It also determines the mapping of blocks to DataNodes.
* When namenode starts, fsimage is loaded into memory
* During start, block reports are retrieved from datanode
* Namenode also contains the file – edits
  + When writes to HDFS happens, the modifications are stored in edits file
  + Edits file is present on harddisk
  + Edits is merged into fsimage – process called checkpointing done by secondaryNameNode

HDFS federation

* In Hadoop 2.x; HDFS federation concept - there are multiple namenodes each having the information of the metadata of blocks
* This list of sub-directories manages by a namenode is called a namespace volume.
* Blocks of files belonging to a namespace is called block pool
* This helps in ensuring the entire cluster does not fail

DataNode1

DataNode2

DataNode3

Pool2

Pool3

Pool1

NameSpace2

NameSpace3

Namespace1

This is also called HDFS federation. This ensure that even if one of the namenode is down, the cluster is not down completely.

SecondaryNameNode

* This is a different machine from NameNode.
* We cannot replace NameNode with SecondaryNameNode
* Special node in HDFS cluster
* Main function is to take checkpoints of the file system metadata present on the NameNode
* It periodically reads 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.

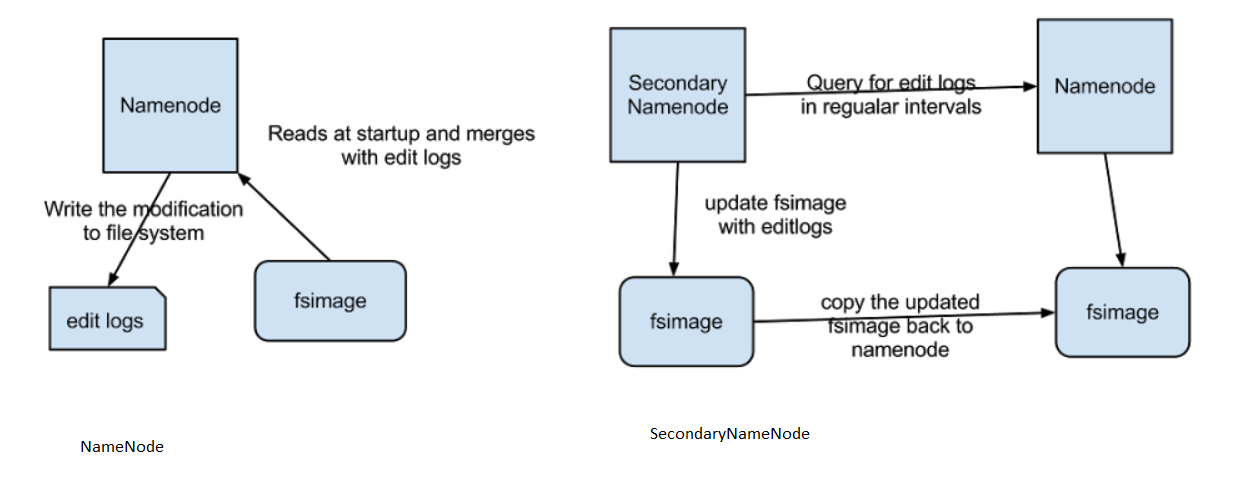
So basically,

Secondary namenode does something called checkpoint process.

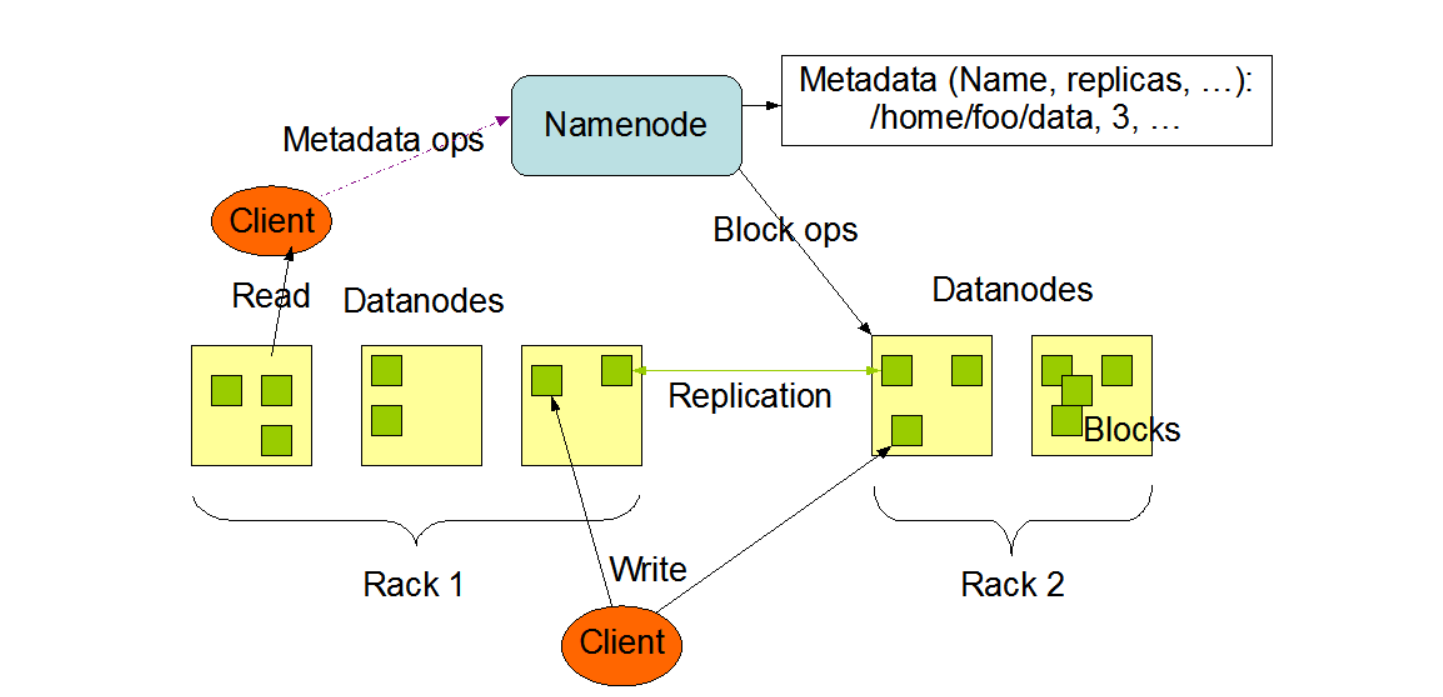
1. Secondary namenode gets editlogs and fsimage periodically from primary NN.

2.  Secondary loads both the fsimage and editlogs to main memory and applies each operation from edits to fsimage.

3. The secondary copies new fsimage to primary and also updates the modified time of fsimage to fstime file, so now fsimage is now updated.



DataNode



* Stores the actual data
* The file is divided into block of defined size (128MB default) and stored in chunks/blocks
* So a single file may be split into multiple blocks and stored
* The DataNodes are responsible for serving read and write requests from the file system’s clients.
* The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.
* The NameNode and DataNode are pieces of software designed to run on commodity machines.
* Replication factor determines how many copies are made.
* There would be multiple datanodes.