International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 6 Issue 10, October-2017, Impact Factor: 3.578

Hadoop Version 2.0

Gaurav Malik

ABSTRACT

In this paper I have explained Hadoop Version 2.0 and explained the shortcomings of Hadoop version1.0 because of which we updated Hadoop and then I explained some other added features of Hadoop 2.0

Keywords:– Namenode High Availability , Yarn

Why We Upgrade Hadoop version 1.0 to Hadoop version 2.0

1. In Hadoop 1.0 NameNode is Single point of Failure

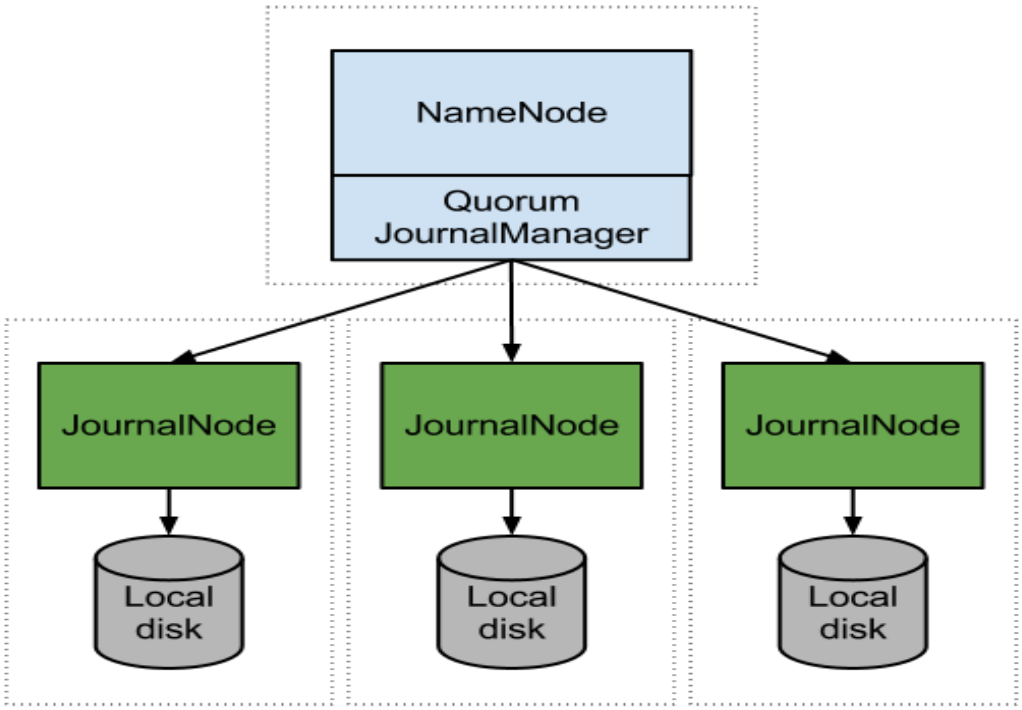
Name node in Hadoop Cluster is most important because it stores all the metadata, if it is down due to some unplanned event such as a machine crash, the whole Hadoop Cluster will be down as well.

There is a solution for this problem in Hadoop 2.0
Namenodein High Availability mode(HA)

High Availability feature, solves the downside by providing the choice of running two redundant Name Nodes in the same cluster in an Active/Passive way (one primary Name Node and other a hot standby Name Node)

Both Active and passive namenode share edits log. All namespace edits are logged to a shared NFS storage or on QJM(Quorum journal Manager) and there is only a single writer to this shared storage at any point of time. The standby Name Node reads from this storage and keeps updated metadata information for cluster. In case of Active Name Node failure, the standby Name Node becomesName Node and starts writing to the shared storage. There is only one write to the shared storage at any point of time.

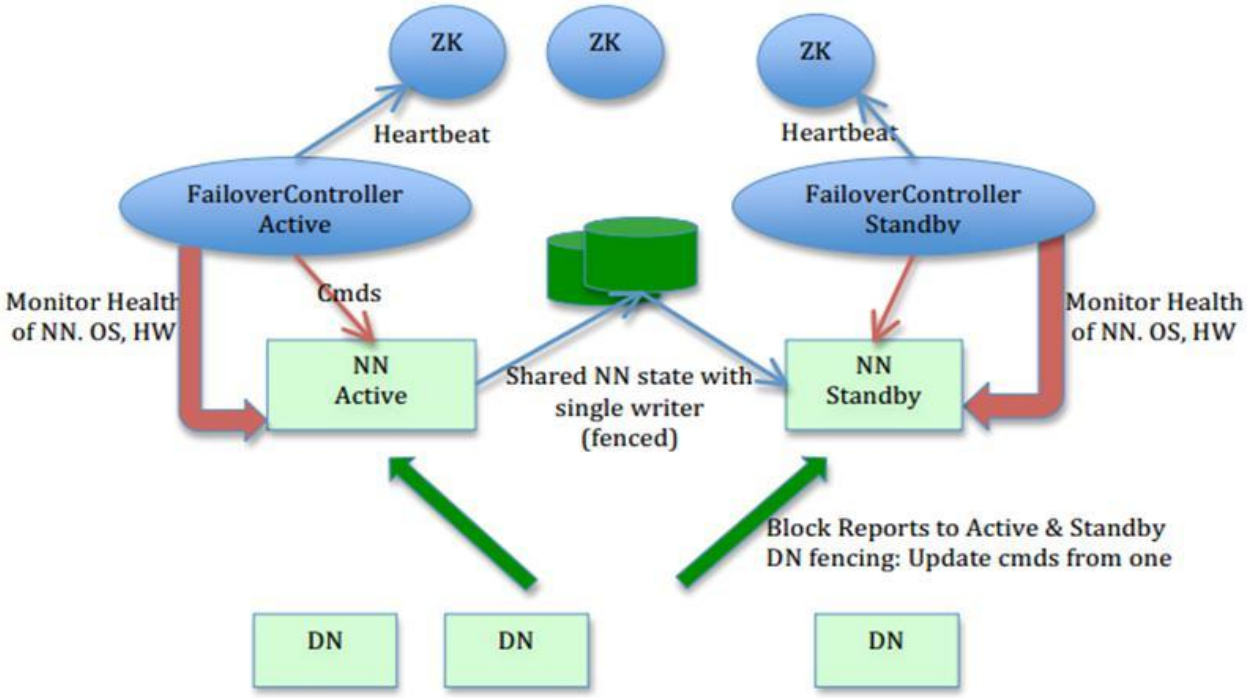
- Journal nodes are contacting Namenode regularly and from meta data they keep the edits
- Both Namemodes are constantly in sync as they collects edits from journal nodes
- Journal node tolerates at most (n-1)/2 failures
- JournalNodes will only allow a single NameNode to write at a time.
- JournalNodes fencing makes sure that if active namenode has failed there is no active session existing and saves system from split-brain scenario
- Whenever a NameNode becomes active, it generates an *epoch number*
- first active NameNode after the namespace is initialized starts with epoch number 1
- any failovers or restarts result in an increment of the epoch number
- When a new NameNode becomes active, it has an epoch number higher than any previous NameNode
- Call JournalNodes to increment their *promised epochs*



Zookeeper(Automatic Failover):- It is used for automatic failover whenever active namenode goes down it will make standby namenode as active namenode. And you don't have to do this thing manually zookeeper will do it for you. It runs on a collection of machines and is highly available

Zookeeper adds two new componenets

1. Zookeeper Quorum
2. Zkfailover Controller Process



- ZKFC pings its local NameNode on a periodic basis
- when the local NameNode is healthy, the ZKFC holds a session open in ZooKeeper.
- If the local NameNode is active, it also holds a special "lock" znode.
- if the session expires, the lock node will be automatically deleted

2 . In Hadoop 1.0 there is too much pressure on JobTracker

In Hadoop 1.0 JobTracker is responsible for both managing the cluster's resources and driving the execution of the MapReduce job.

There is a solution for this problem in Hadoop 2.0

In Hadoop 2.0 YARN splits up the two major functionsof JobTracker (resource management and job scheduling/monitoring) into two separate daemons:

Resource Manager:-It is a master daemon with which client interacts by way of submitting the jobs

- It's a Daemon which gets initialized by Hadoop admins
- Manages entire processing part of a submitted Hadoop 2.0 job
- Listens to heart beats from node manager
- Spawns 2 objects – scheduler and application manager

Scheduler :- It registers container because its it duty to assign container to whosoever request for container. It's the manager of all the containers in cluster

Node Manager :- It is a slave daemon and it spawns 2 objects container and app master
\
App Master:- It is one per job. It's a java program not a daemon. Its initialized when u submit the job

Hadoop 2.0 also includes some more features

1. You can run Non MapReduce Application on Hadoop 2.0

In Hadoop 1.0, you can onlyrun MapReduce framework jobs to process the data stored in HDFS. There were no other models (other than MapReduce) of data processing. For other processing way like Real-time or graph analysis on the similar data stored in HDFS, you need to take out that data to some alternate storage like HBase because Hadoop 1.0 was only supporting MapReduce Processing manner.

Hadoop 2.0 came up with new framework YARN (Yet another Resource Navigator), that has the ability to run Non-Map Reduce application. By using YARN API's other frameworks canrun on top of HDFS. This enablesrunning Non-MapReduce Big Data Applications on Hadoop. Spark, MPI and HAMA are few examples

2. Native Windows Support

Initially, Hadoop only used to support the UNIX family of operating systems. Butin Hadoop 2.0, the Windows operating system is also supported. This extends the reach of Hadoop significantly to a sizable Windows server Market.

3. Beyond Batch Oriented application: In Hadoop 2.0 we can run interactive and streaming applications as well where as in Hadoop 1.0 we can only have batch oriented applications

4. HDFS- Multiple Storage

When it comes to Hadoop 2.0 there is one more change and that is the support for heterogeneous storage. Hadoop 1.0 treated all storage devices (be it spinning disks or SSDs) on a DataNode as a single uniform pool; although one could store data on an SSD, one could not control which data. Where as in Heterogeneous storage in Hadoop 2.0 , the system can distinguish between storage types and also make the storage type information available to frameworks and applications so that they can take advantage of storage properties.

5. Faster access to data—Data Node caching

Users and applications (such as Hive, Pig or HBase) will determine currently a collection of files that require to be cached. For instance, dimension tables in Hive can be configured for caching in the DataNode RAM, enabling quick reads for Hive queries to these frequently looked up tables.

6. HDFS Snapshots

Hadoop 2 adds support for file system snapshots. A snapshot is a point-in-time image of the entire file system or a sub tree of a file system. A snapshot has many uses:

Protection against user errors: An admin can set up a process to take snapshots periodically. If a user accidentally deletes files, these can be restored from the snapshot that contains the files.

Backup: If an admin wants to back up the entire file system or a subtree in the file system, the admin takes a snapshot and uses it as the starting point of a full backup. Incremental backups are then taken by copying the difference between two snapshots.

Disaster recovery: Snapshots can be used for copying consistent point-in-time images over to a remote site for disaster recovery.

REFERENCES

[1]. http://saphanatutorial.com/hadoop-1-0-vs-hadoop-2-0/
[2]. http://crazyadmins.com/role-of-journal-nodes-in-ha/
[3]. http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yar site/HDFS High Availability with QJM. html# Deployment
[4]. http://blog.cloudera.com/blog/2012/10/quorum-based-journaling-in-cdh4-1/
[5]. https://issues.apache.org/jira/secure/attachment/12547598/qjournal-design.pdf