Table of Contents

HDFS

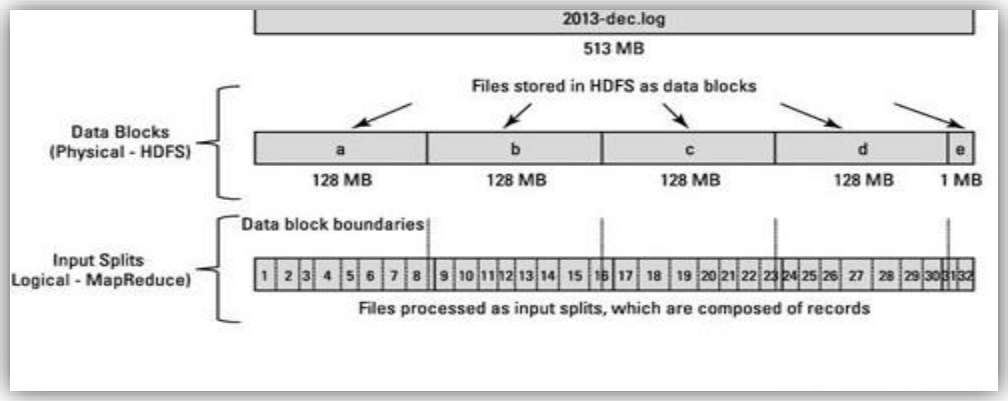
HDFS

- Follows ***Master*** and ***Slave*** topology.
- **Master**: NameNode
  - Stores ***meta-data*** i.e. number of blocks, their replications, replicas and other details.
- **Slave**: DataNode
  - Stores ***actual data*** in HDFS.
  - Sends information to NameNode about files and blocks stored in that node and responds to the NameNode.

Block Size (128 MB)

- HDFS Stores each ***file as blocks*** and distribute it across Hadoop clusters.
- This block is by default 128 MB.
- It's much larger then ***Linux*** (4 KB).

Input split



Hadoop Config Files

- **Code-site.xml**
  - Informs Hadoop daemon where ***NameNode*** runs in the cluster.
  - It contains the ***configuration settings*** for Hadoop Core such as I/O ***settings*** that are common to HDFS and MapReduce.
- **HDFS-site.xml**
  - The ***hdfs-site. xml file*** contains the ***configuration*** settings for HDFS daemons; the NameNode, the Secondary NameNode, and the DataNodes.
- **MAPRED-SITE. XML**
  - Required for ***runtime environment*** settings of a Hadoop.
  - It contains the configuration settings for MapReduce.

HDFS commands

(HDFS Commands, n.d.)

Hadoop ***fs*** command behaves like ***ls*** but recursively displays entries in all subdirectories of a path.

1. **To know the version of HDFS**

   ⇨ ***hdfs dfs version***

2. **Make directory**

   ⇨ ***mkdir < path >***

   - Ex: hdfs dfs –**mkdir** /user/folder/dir1

3. **Display command (list of contents of the directory)**

   ⇨ ***ls <path?***

   - Ex: hdfs dfs **–ls** /user/folder/dir1

4. **Copy file (move local file into distributed file system[dfs])**

   More robust as it helps to move multiple file paths at once to HDFS.

   As well as it reads input from stding and writes it directly to HDFS.

   ⇨ ***put <localsrc> <destination>***

   - Ex: hdfs dfs –**put** /home/folder/desktop/sample /user/folder/dir1

5. **copyFormLocal**

Similar to put command but *source is restricted* to local file reference.

⇨ ***copyFormLocal <local src> <destination>***

6. **get**

Copies the file or directory in HDFS identified by the source to the local file system path identified by local destination.

⇨ ***get [-crc] <src> <localDestination>***

- Ex: hdfs dfs -get /user/folder/dir2/sample /home/folder/Desktop

Other commands

https://data-flair.training/blogs/top-hadoop-hdfs-commands-tutorial/

HDFS Replication

- HDFS reliably stores large files across machines in large clusters.
- It stores each file in sequence of blocks.
- All blocks are of same size except the last block.
- Blocks of files are replicated for ***fault tolerance***.

HDFS HA

(HDFS HA, n.d.)

NameNode High Availability.

- HA provides options of ***running 2*** NameNode. (Active and Passive)

- **Active NameNode:** Handles all client operations.
- **Passive NameNode:** Standby NameNode with similar data as Active NameNode. Maintains enough space for fast failover.

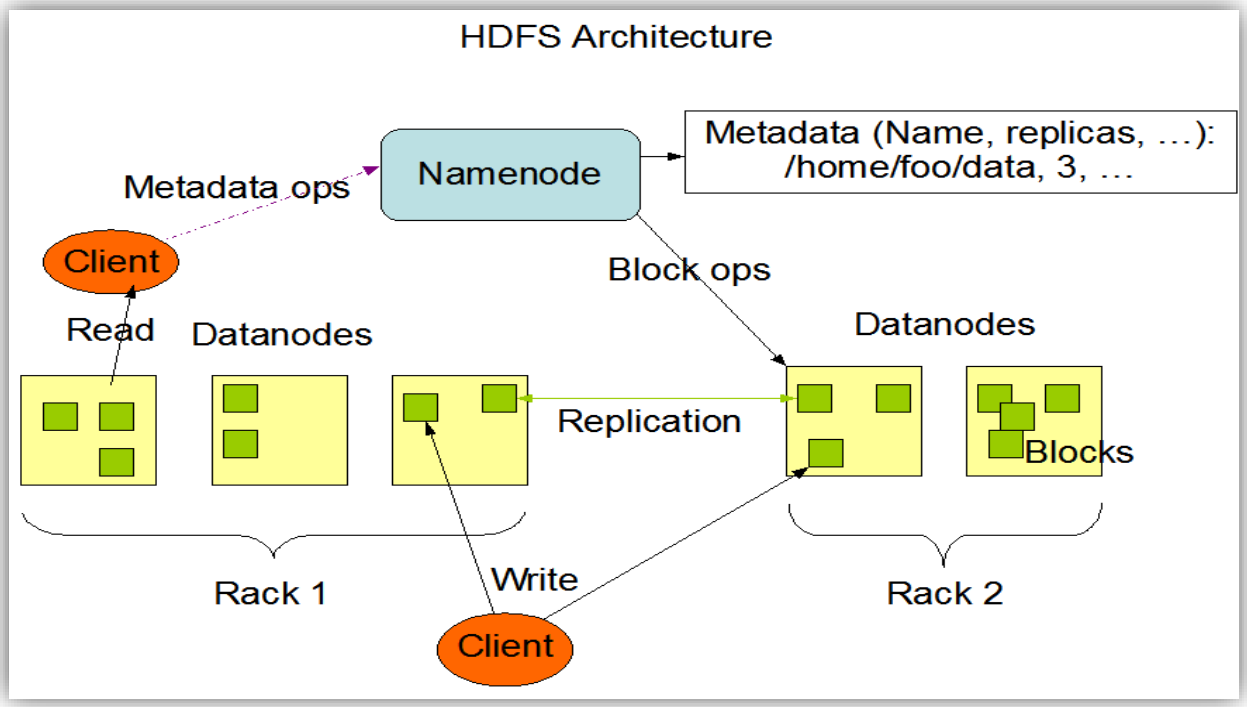
HDFS Failover

- In case of failure of active NameNode it ensures that ***standby*** NameNode has read all the edits from journal node before promoting itself as Active Node.

HDFS fencing

- It avoids ***Split-Brain-Scenario*** i.e. it helps to check that only one NameNode should be active at time to avoid corruption of data if two NameNodes are active.

Communication between NameNode and DataNode



- NameNode periodically receives ***heartbeat*** from DataNode.

- If heartbeat is not received for specific time then that DataNode is declared as ***dead***.
- System begins replicating the blocks that were stored on dead DataNode.
- The NameNode orchestrates (*co-ordinate elements to produce required effect*) the replication of data blocks from one DataNode to another.
- The replication data transfer ***happens directly*** between DataNodes and the data never passes through the NameNode.
- The replication data ***transfer happens directly between DataNodes*** and the data never passes through the NameNode.
- DataNode stores a block and forwards the request to the next block.

References

(n.d.). Retrieved from https://data-flair.training/blogs/hadoop-hdfs-namenode-high-availability/

(n.d.). Retrieved from https://data-flair.training/blogs/top-hadoop-hdfs-commands-tutorial/