DSE-3264 & Big Data Analytics Laboratory Manual

What is HDFS??

Hadoop comes with a distributed file system called HDFS. In HDFS data is distributed over several machines and replicated to ensure their durability to failure and high availability to parallel application. It is cost effective as it uses commodity hardware. It involves the concept of blocks, data nodes and node name.

HDFS building Blocks:

Blocks: A Block is the minimum amount of data that it can read or write.HDFS blocks are 128 MB by default and this is configurable.Files n HDFS are broken into block-sized chunks,which are stored as independent units.Unlike a file system, if the file is in HDFS is smaller than block size, then it does not occupy full block?s size, i.e. 5 MB of file stored in HDFS of block size 128 MB takes 5MB of space only.The HDFS block size is large just to minimize the cost of seek.

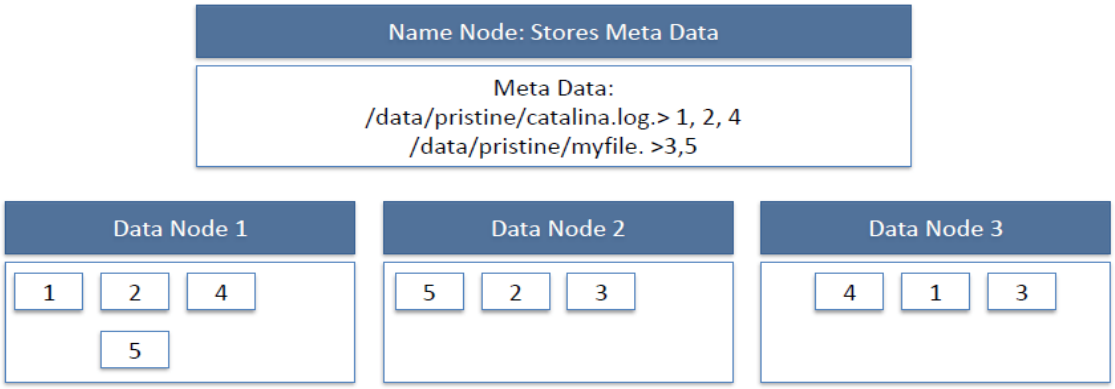
Name Node: HDFS works in master-worker pattern where the name node acts as master.Name Node is controller and manager of HDFS as it knows the status and the metadata of all the files in HDFS; the metadata information being file permission, names and location of each block.The metadata are small, so it is stored in the memory of name node,allowing faster access to data. Moreover the HDFS cluster is accessed by multiple clients concurrently,so all this information is handled bya single machine. The file system operations like opening, closing, renaming etc. are executed by it.

Data Node: They store and retrieve blocks when they are told to; by client or name node. They report back to name node periodically, with list of blocks that they are storing. The data node being a commodity hardware also does the work of block creation, deletion and replication as stated by the name node.

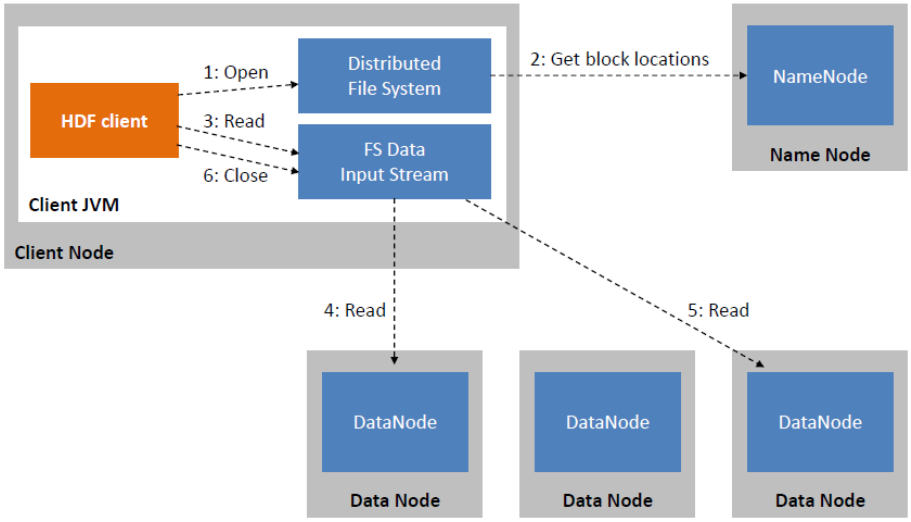
Secondary Name Node: It is a separate physical machine which acts as a helper of name node. It

performs periodic check points.It communicates with the name node and take snapshot of meta data which helps minimize downtime and loss of data.

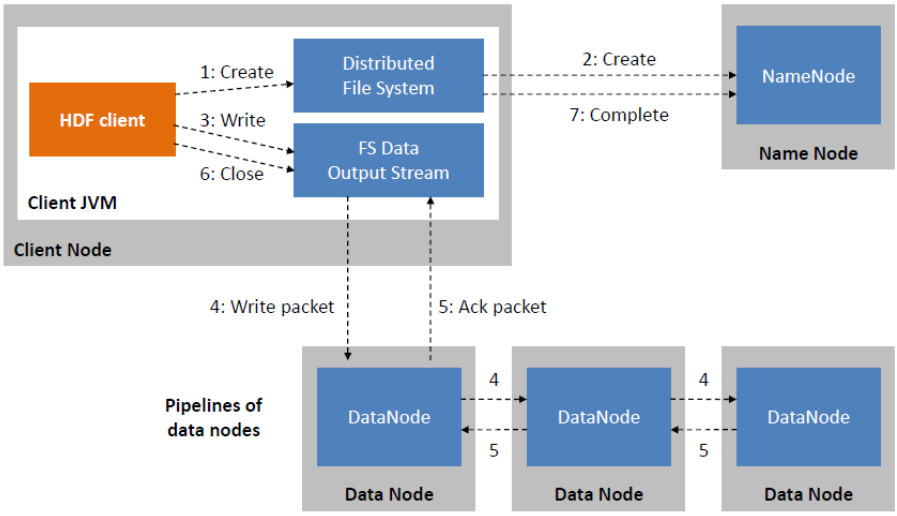
- **HDFS DataNode and NameNode Image:**



- **HDFS Read workflow**:



- **HDFS Write:**



- To use the HDFS commands, first you need to start the Hadoop services using the following command:

  $sbin/start-all.sh

- To check the Hadoop services are up and running use the following command:

  $jps

- Check the datanode service is up by running ***jps*** commond at client side.

```
suraj@suraj:~/hadoop-2.5.0-cdh5.3.2$ jps
2546 SecondaryNameNode
2404 DataNode
2295 NameNode
2760 ResourceManager
2874 NodeManager
4251 Jps
suraj@suraj:~/hadoop-2.5.0-cdh5.3.2$
```

- To perform various file operations use the prefix ***Hadoop fs/ hdfs dfs*** as a prefix for each command

- To create a directory. In Hadoop *dfs* there is no home directory by default. So let's first create it sing ***mkdir*** commond.

Week 1 Exercise: Hadoop Distributed File System

1. List all possible Linux file operations and execute each one of them in Linux CLI.
2. Interact with HDFS using command line interface to understand the basic working structure of Hadoop cluster. Using Hadoop CLI, demonstrate the following commands to:
    - Create a directory in HDFS.
    - create an empty file
    - copy files/folders from local file system to hdfs store.
    - print file contents.
    - copy files/folders from hdfs store to local file system.
    - move file from local to hdfs
    - copy files within hdfs
    - move files within hdfs
    - size of each file in directory
    - total size of directory/file
    - last modified time of directory or path
    - change the replication factor of a file/directory in HDFS.
    - List the contents of a directory in HDFS.
    - Remove a file from HDFS.
    - Change File Permissions
    - Changing File Ownership
    - Checksum Calculation
    - File Concatenation
    - File Compression/Decompression
    - File Block Location Information
    - File Encryption/Decryption
3. Use web interface to monitor Name node manager, resource manager, and Data node status.

Week 2: Exercise: MapReduce with Python

1. Consider the text file (consider larger file size) of your choice and perform word count using

MapReduce technique.

2. Perform Matrix operations using MapReduce by considering 3 * 3 matrix and perform following operations:
    i. Matrix addition and subtraction
    ii. Matrix Multiplication
    iii. Matrix transpose

    Note: Consider 3*3 matrix content as shown below

    a,0,0,10
    a,0,1,20
    a,0,2,30
    a,1,0,40
    a,1,1,50
    a,1,2,60
    a,2,0,70
    a,2,1,80
    a,2,2,90

    b,0,0,1
    b,0,1,2
    b,0,2,3
    b,1,0,4
    b,1,1,5
    b,1,2,6
    b,2,0,7
    b,2,1,8
    b,2,2,9

3. Create a text file containing the 20 student details such as registration number, name and marks (ex: 1001, john,45) .Write a MapReduce program to sort data by student name.

Week 3: Exercise: MapReduce with Python

1. Write a MapReduce program to find unit wise salary for the bellow given data.

EmpNo	EmpName	Unit	Designation	Salary
1001	John	IMST	TA	30000
1002	Jack	CLOUD	PM	80000
1003	Joshi	FNPR	TA	35000
1004	Jash	ECSSAP	PM	75000
1005	Yash	FSADM	SPM	60000
1006	Smith	ICS	TA	24000
1007	Lion	IMST	SPM	56000
1008	kate	FNPR	PM	76000
1009	cassy	MFGADM	TA	40000
1010	ronald	ECSSAP	SPM	65000

2. Consider the following sample text file to compute the the average, minimum and maximum recorded temperature by year wise using concept of Map Reduce.

Temperature.txt

2014 44

2013 42

2012 30

2013 44

2010 45

2014 38

2011 42

2010 44