HDFS

BY
PROF. PRAKASH PATEL
ASST.PROF.-IT DEPT.
GIT

What is HDFS?

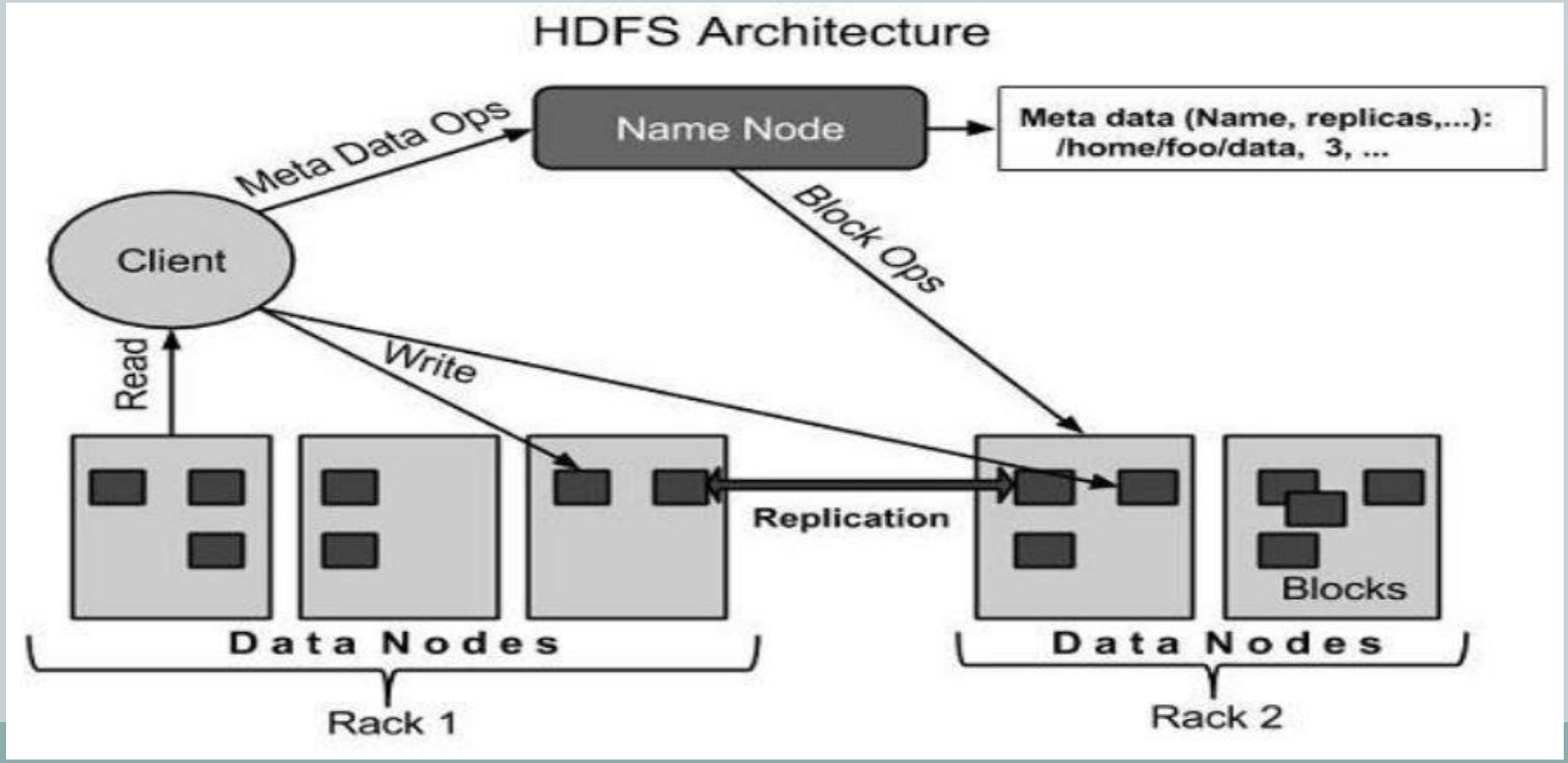
- Hadoop File System was developed using distributed file system design.
- It is run on commodity hardware. HDFS is highly fault-tolerant and designed using low-cost hardware.
- HDFS holds very large amount of data and provides easier access.
- HDFS also makes applications available to parallel processing.

Features of HDFS

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

Architecture of HDFS

- Given below is the architecture of a Hadoop File System.



Elements of HDFS

- **Namenode**
  - The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software.
  - It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks:
  - Manages the file system namespace.
  - Regulates client's access to files.
  - It also executes file system operations such as renaming, closing, and opening files and directories.

Datanode

- The datanode is a commodity hardware having the GNU/Linux operating system and datanode software.
- For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.
  - Datanodes perform read-write operations on the file systems, as per client request.
  - They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

Block

- Generally the user data is stored in the files of HDFS.
- The file in a file system will be divided into one or more segments and/or stored in individual data nodes.
- These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a *Block*.
- The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goals of HDFS

- **Fault detection and recovery:** Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.
- **Huge datasets:** HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.
- **Hardware at data:** A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

HDFS Operations

- **Starting HDFS**
- Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command.
- $ hadoop namenode -format
- After formatting the HDFS, start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.
- $ start-dfs.sh

HDFS Operations

- **Listing Files in HDFS**
- After loading the information in the server, we can find the list of files in a directory, status of a file, using '**ls**'. Given below is the syntax of **ls** that you can pass to a directory or a filename as an argument.
- $ $HADOOP_HOME/bin/hadoop fs -ls <args>

HDFS Operations

- **Inserting Data into HDFS**
- Assume we have data in the file called file.txt in the local system.
  - **Step 1**
    - You have to create an input directory.
    - $ $HADOOP_HOME/bin/hadoop fs -mkdir /user/input
  - **Step 2**
    - Transfer and store a data file from local systems to the Hadoop file system using the put command.
    - $ $HADOOP_HOME/bin/hadoop fs -put /home/file.txt /user/input
  - **Step 3**
    - You can verify the file using ls command.
    - $ $HADOOP_HOME/bin/hadoop fs -ls /user/input

Retrieving Data from HDFS

- Assume we have a file in HDFS called **outfile**. Given below is a simple demonstration for retrieving the required file from the Hadoop file system.
- **Step 1**
- Initially, view the data from HDFS using **cat** command.
- $ $HADOOP_HOME/bin/hadoop fs -cat /user/output/outfile
- **Step 2**
- Get the file from HDFS to the local file system using **get** command.
- $ $HADOOP_HOME/bin/hadoop fs -get /user/output/ /home/hadoop_tp/

Shutting Down the HDFS

- You can shut down the HDFS by using the following command.
- $ stop-dfs.sh