GANDHINAGAR INSTITUTE OF TECHNOLGY

Information Technology Department

Data Mining & Business Intelligence (2170715)

HDFS

Prepared By:

Patel Maulik Satishkumar (150124116006)

Guided By: **Prof. Rahul A. Vaghela**

Content

- Introduction to Big Data - Hadoop
- What is HDFS ?
- Features of HDFS
- HDFS Architecture
- Elements of HDFS
- Goals of HDFS
- HDFS Operations
- Reference

Introduction to Big Data - Hadoop

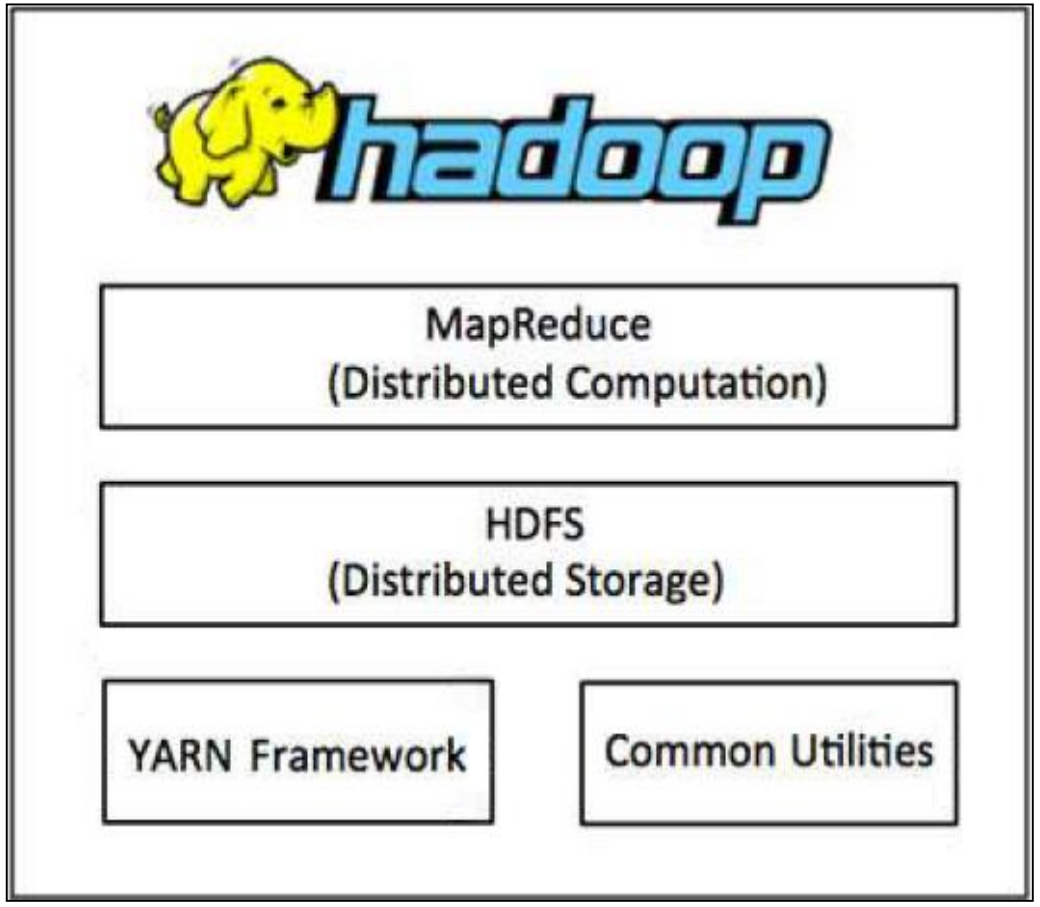
What is Big Data ?

- ✓ Big Data is a collection of large datasets that cannot be processed using traditional computing techniques.
- ✓ Big data involves the data produced by different devices and applications. The data in it will be of three types.
  - Structured data: Relational data.
  - Semi Structured data: XML data.
  - Unstructured data: Word, PDF, Text, Media Logs.

What is Hadoop ?

- ✓ Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models.

Introduction to Big Data - Hadoop



Hadoop Architecture

Hadoop MapReduce :

- This is YARN-based system for parallel processing of large data sets.

HDFS :

- A distributed file system that provides high throughput access to application data.

Hadoop YARN :

- This is a framework for job scheduling and cluster resource management.

Hadoop Common :

- These are Java libraries and utilities required by other Hadoop modules.

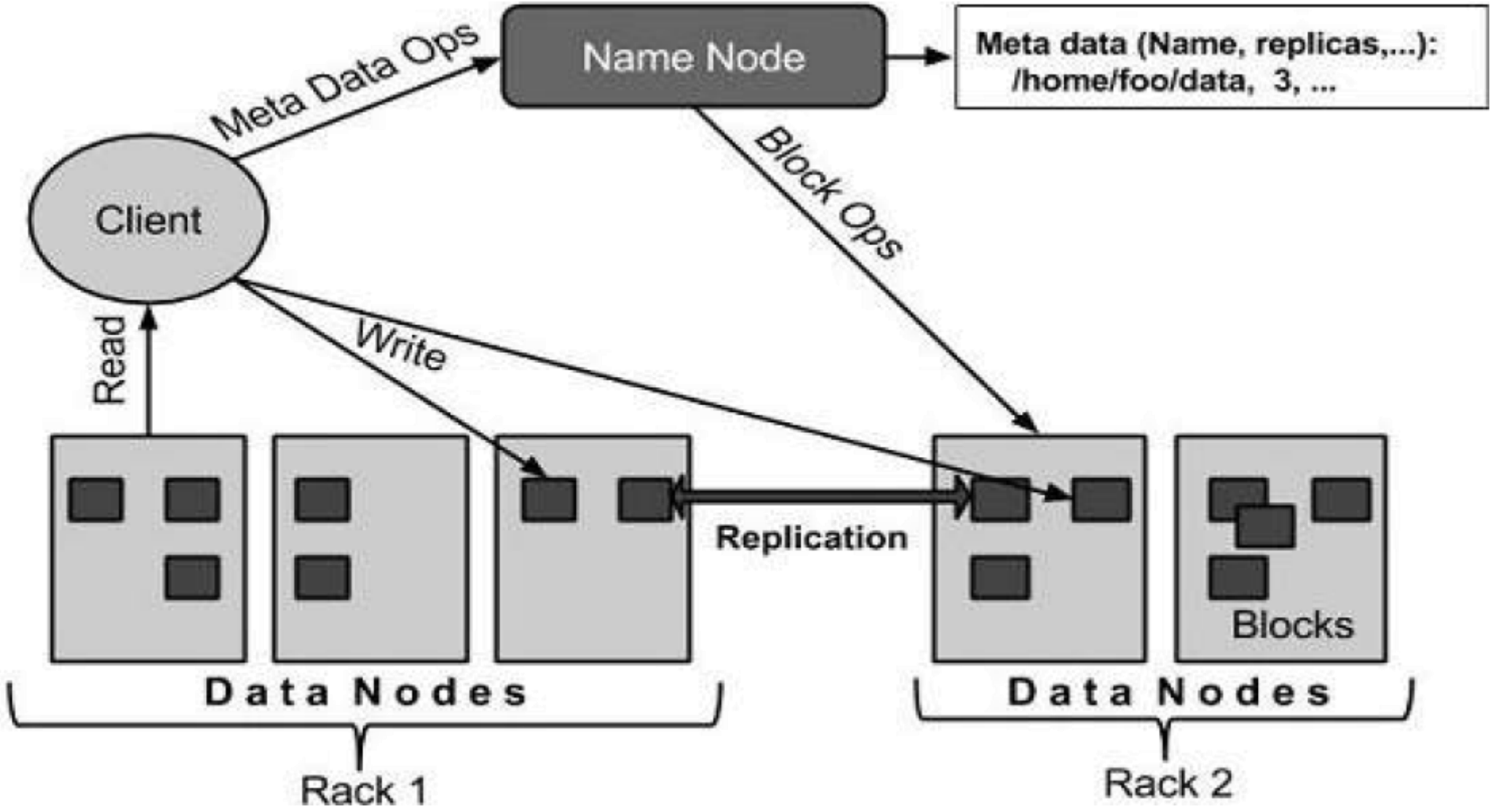
What is **HDFS** ?

- **HDFS** stands for **H**adoop **D**istributed **F**ile **S**ystem.
- 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.
- To store such huge data , the files are stored across multiple machines.
- These files are stored in redundant fashion to rescue the system from possible data losses in case of failure.
- 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.
- Hadoop HDFS is Master-Slave architecture , so the processing speed is very high & system failure rate is very low.

HDFS Architecture (master-slave architecture)



Elements of HDFS

1) 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 anddirectories.

Elements of HDFS

2) 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.

Elements of HDFS

3) 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 mechanism 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

```
$ hadoop namenode –format
$ start-dfs.sh          or          $ start-all.sh
```

Listing files in HDFS

```
$ $ HADOOP_HOME/bin/hadoop  fs  -ls  <args>
```

Inserting Data into HDFS

```
$ $ HADOOP_HOME/bin/hadoop  fs –mkdir  /user/dir_name
$ $ HADOOP_HOME/bin/hadoop  fs -put  /home/file.txt  /user/dir_name
$ $ HADOOP_HOME/bin/hadoop  fs -ls  /user/dir_name
```

HDFS Operations (Commands)

Retrieving Data from HDFS

```
$ $ HADOOP_HOME/bin/hadoop fs  -cat  /user/dir_name/file
$ $ HADOOP_HOME/bin/hadoop fs  -get  /user/output/   /home/hadoop_tp/
```

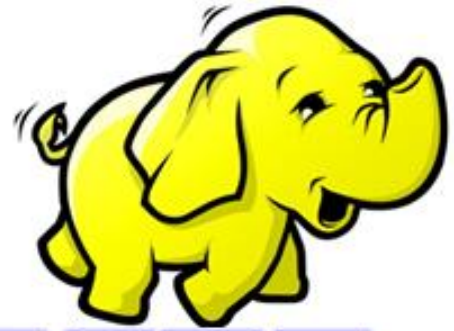
Shutting Down the HDFS

```
$ stop-dfs.sh          or          $ stop-all.sh
```

Reference

- Accessed [05/08/2018]. Available : **https://www.tutorialspoint.com**
- Accessed [05/08/2018]. Available : **https://www.youtube.com**
- Also Available on my github site:

maulikpatel295.github.io/ALA/2160715_150124116006.pdf



THANK YOU