HDFS Operation

These are following operations of HDFS performed: -

1. **Start HDFS**
2. **List the file in HDFS**
3. **Insert Data into the HDFS**
4. **Retrieving data HDFS**
5. **Stop HDFS**

1 Start HDFS:-

HDFS file system, open namenode(HDFS server) and execute the following commands :-

```
$ hadoop namenode –format
```

This command format the HDFS system and start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.

```
$start   -dfs.sh
```

2 Listing the files in HDFS:-

After starting the namenode server we can find the list of files in a directory, status of a file by using “**ls**” command as follows :-

```
$ HADOOP_HOME/bin/hadoop fs -ls
```

3 Insert Data into HDFS:-

This operation of HDFS is used to insert data of existing local system files into the HDFS distributed system. These are the following steps must be follow to insert a file “mca4.txt” file into the HDFS :-

Step 1:- We needs to create a directory for input the data file in existing user.

```
$ HADOOP_HOME/bin/hadoop fs -mkdir /usermca4/input
```

Step 2: - After create the input directory we needs to put a data file from local system to the Hadoop file system using put command.

```
$ HADOOP_HOME/bin/hadoop fs -put /home/file.txt /usermca4/input
```

Step 3: -After the above to operation verify the file using “ls” command

```
$ HADOOP_HOME/bin/hadoop fs -ls /usermca4/input
```

4 . Retrieving the data from DHFS file:-

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

We can view the data from HDFS using **cat** command.

```
$ HADOOP_HOME/bin/hadoop fs -cat /usermca4/output/outfile
```

Step 2

Get the file from HDFS to the local file system using **get** command.

```
$ $HADOOP_HOME/bin/hadoop fs -get /usermca4/output/ /home/hadoop_tp/
```

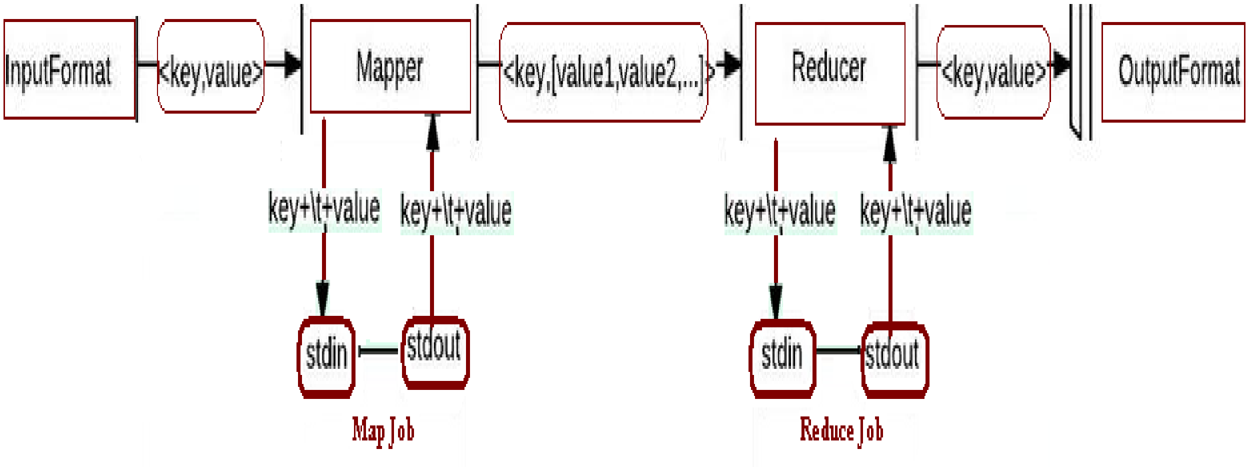
5 . Shutting Down the HDFS

We can shut down the HDFS by using the following command.

```
$ stop-dfs.sh
```

Hadoop Streaming

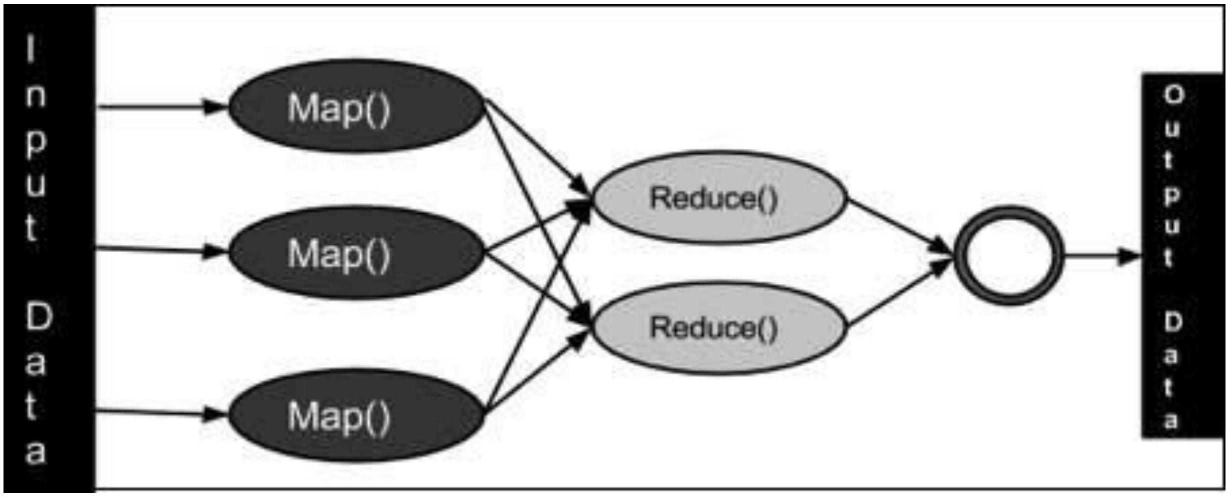
Hadoop streaming is a utility that comes with the Hadoop distribution. This utility allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer. The mapper and the reducer are read the input from standard input and emit the output to standard output. As the mapper task runs, it converts its inputs into lines and feed the lines to the standard input (STDIN) of the process. In the meantime, the mapper collects the line-oriented outputs from the standard output (STDOUT) of the process and converts each line into a key/value pair, which is collected as the output of the mapper. By default, the prefix of a line up to the first tab character is the key and the rest of the line (excluding the tab character) is the value.



What is MapReduce?

It is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.



MapReduce Processing

ALGORITHM MapReduce

1. MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

    - **Map stage** − The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

- **Reduce stage** − This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

2. During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
3. The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
4. Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
5. After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.