MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Part B-01**

PART: B) ASSIGNMENT NO: 01

AIM: Write a code in JAVA for a simple WordCount application that counts the number of occurrences of each word in a given input set using the Hadoop MapReduce framework on local-standalone set-up.

OBJECTIVES:

- To learn and understand working MapReduce framework.
- To learn the concept of how to Input data to HDFS and view resultant file in the output folder.

APPARATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative.
- Front End: Java,Hadoop
- Dataset: Fruits.txt (or any other desired format)

PREREQUISITE:

Fundamentals of Hadoop commands and concept of MapReduce framework,HDFS.

THEORY:

- MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.
- MapReduce is a processing technique and a program model for distributed computing based on java.
- The MapReduce algorithm contains two important tasks, namely 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 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.
- The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.
- The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable- Comparable interface to facilitate sorting by the framework.
- Input and Output types of a MapReduce job:

	Input	**Output**
Map	<k1, v1>	list (<k2, v2>)
Reduce	<k2, list(v2)>	list (<k3, v3>)

(Input) <k1,v1> -> map -> <k2, v2>-> reduce -> <k3, v3> (Output).Some Terminologies:

Mapper - Mapper maps the input key/value pairs to a set of intermediate key/value pair.

NamedNode - Node that manages the Hadoop Distributed File System (HDFS).

DataNode - Node where data is presented in advance before any processing takes place.

MasterNode - Node where JobTracker runs and which accepts job requests from clients.

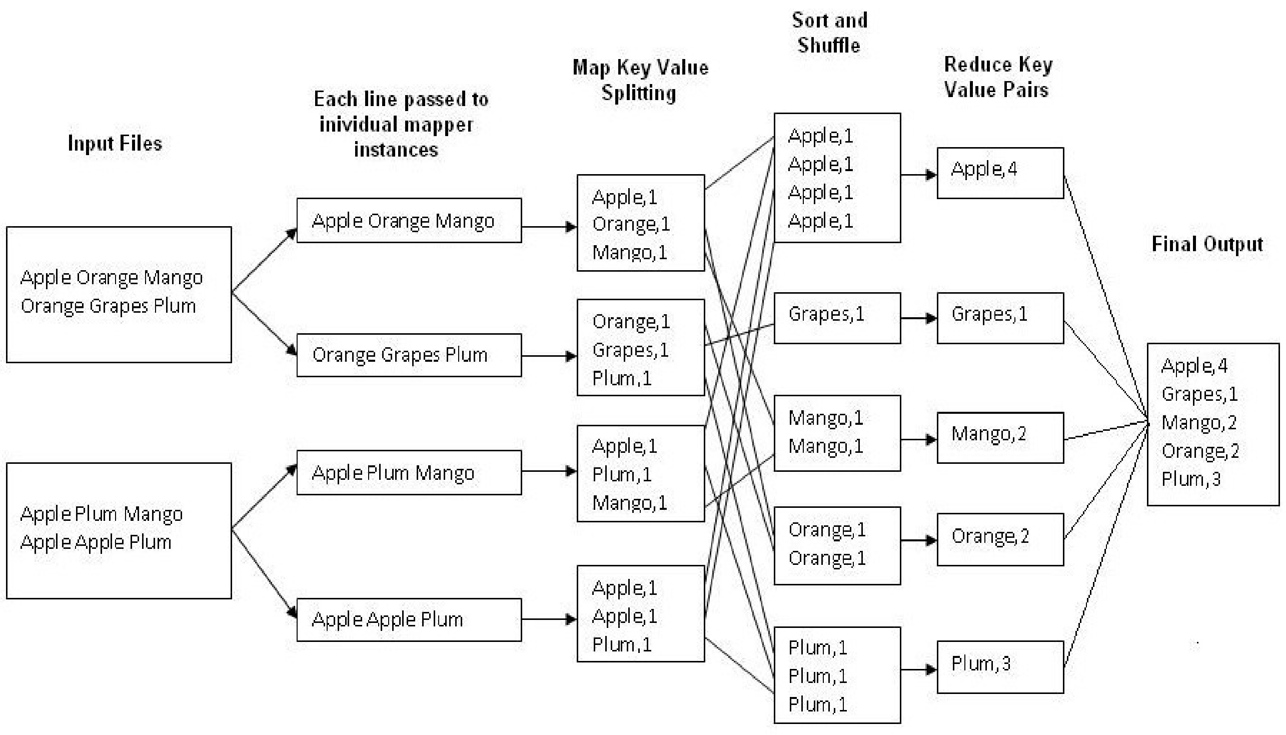
SlaveNode - Node where Map and Reduce program runs.

JobTracker - Schedules jobs and tracks the assign jobs to Task tracker.

Task Tracker - Tracks the task and reports status to JobTracker.

Job - A program is an execution of a Mapper and Reducer across a dataset.

Task - An execution of a Mapper or a Reducer on a slice of data.



IMPLEMENTATION:

• Let us assume we are in the home directory of a Hadoop user (e.g. /home/admin1).

• Follow the steps given below to compile and execute the above program.

Compilation and Execution steps:

• Step 1

The following command is to create a directory to store the compiled java classes.

```
$ mkdir words
```

• Step 2

Download hadoop-core-1.2.1.jar, which is used to compile and execute the MapReduce program. Visit the following link to download the jar.

Let us assume the downloaded folder is /home/words.

http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core/1.2.1

• Step 3

The following commands are used for compiling the wordcount.java program and creating a jar for the program.

```
$ javac -classpath hadoop-core-1.2.1.jar words/WordCount.java
$ jar -cvf words.jar -C words/ .
```

• Step 4

The following command is used to create an input directory in HDFS.

```
$hadoop fs -mkdir /input
```

• Step 5

The following command is used to copy input dataset file on HDFS.

```
$hadoop fs -put fruits.txt /input
```

• Step 6

The following command is used to verify the files in the input directory.

```
$hadoop fs -ls /input
```

• Step 7

The following command is used to run the Wordcount application by taking the input files from the input directory.

```
$hadoop jar words.jar WordCount /input /output
```

Wait for a while until the file is executed. After execution, the output will contain the number of input splits, the number of Map tasks, the number of reducer tasks, etc. The output directory must not be existing already.

• Step 8

The following command is used to verify the resultant files in the output folder.

```
$hadoop fs -ls /output
```

• Step 9

The following command is used to see the output in part-r-00000 file. This file is generated by HDFS.

```
$hadoop fs -cat /output/part-r-00000
```

• Step 10

The following command is used to copy the output file from HDFS to the local file system for analyzing.

```
$hadoop fs -get /output/part-r-00000
```

CONCLUSION:

QUESTIONS:

1. Explain hadoop daemons.
2. Explain HDFS.
3. Explain MapReduce Framework