

Experiment No:13

Aim: Design a distributed application using MapReduce which processes a log file of a system.

Theory: In this tutorial, you will learn to use Hadoop with MapReduce Examples. The input data used is [SalesJan2009.csv](#). It contains Sales related information like Product name, price, payment mode, city, country of client etc. The goal is to **Find out Number of Products Sold in Each Country.**

First Hadoop MapReduce Program

Now in this MapReduce tutorial, we will create our first Java MapReduce program:

| | A | B | C | D | E | F | G | H | I | |
|----|------------------|----------|-------|-----------|------------|------------|------------|------------|------------------|-----|
| 1 | Transaction_date | Product | Price | Payment | Name | City | State | Country | Account_Created | Las |
| 2 | 01-02-2009 06:17 | Product1 | 1200 | Mastercar | carolina | Basildon | England | United Kir | 01-02-2009 06:00 | 0 |
| 3 | 01-02-2009 04:53 | Product1 | 1200 | Visa | Betina | Parkville | MO | United Sta | 01-02-2009 04:42 | 0 |
| 4 | 01-02-2009 13:08 | Product1 | 1200 | Mastercar | Federica e | Astoria | OR | United Sta | 01-01-2009 16:21 | 0 |
| 5 | 01-03-2009 14:44 | Product1 | 1200 | Visa | Gouya | Echuca | Victoria | Australia | 9/25/05 21:13 | 0 |
| 6 | 01-04-2009 12:56 | Product2 | 3600 | Visa | Gerd W | Cahaba He | AL | United Sta | 11/15/08 15:47 | 0 |
| 7 | 01-04-2009 13:19 | Product1 | 1200 | Visa | LAURENCE | Mickleton | NJ | United Sta | 9/24/08 15:19 | 0 |
| 8 | 01-04-2009 20:11 | Product1 | 1200 | Mastercar | Fleur | Peoria | IL | United Sta | 01-03-2009 09:38 | 0 |
| 9 | 01-02-2009 20:09 | Product1 | 1200 | Mastercar | adam | Martin | TN | United Sta | 01-02-2009 17:43 | 0 |
| 10 | 01-04-2009 13:17 | Product1 | 1200 | Mastercar | Renee Elis | Tel Aviv | Tel Aviv | Israel | 01-04-2009 13:03 | 0 |
| 11 | 01-04-2009 14:11 | Product1 | 1200 | Visa | Aidan | Chatou | Ile-de-Fra | France | 06-03-2008 04:22 | 0 |
| 12 | 01-05-2009 02:42 | Product1 | 1200 | Diners | Stacy | New York | NY | United Sta | 01-05-2009 02:23 | 0 |
| 13 | 01-05-2009 05:39 | Product1 | 1200 | Amex | Heidi | Eindhoven | Noord-Br | Netherlan | 01-05-2009 04:55 | 0 |
| 14 | 01-02-2009 09:16 | Product1 | 1200 | Mastercar | Sean | Shavano | TX | United Sta | 01-02-2009 08:32 | 0 |
| 15 | 01-05-2009 10:08 | Product1 | 1200 | Visa | Georgia | Eagle | ID | United Sta | 11-11-2008 15:53 | 0 |
| 16 | 01-02-2009 14:18 | Product1 | 1200 | Visa | Richard | Riverside | NJ | United Sta | 12-09-2008 12:07 | 0 |
| 17 | 01-04-2009 01:05 | Product1 | 1200 | Diners | Leanne | Julianstov | Meath | Ireland | 01-04-2009 00:00 | 0 |
| 18 | 01-05-2009 11:27 | Product1 | 1200 | Visa | Isabel | Ottawa | Ontario | Canada | 01-05-2009 00:25 | 0 |

Data of SalesJan2009

Ensure you have Hadoop installed. Before you start with the actual process, change user to 'hduser' (id used while Hadoop configuration, you can switch to the userid used during your Hadoop programming config).

su - hduser_

```
guru99@guru99-VirtualBox:~$ su - hduser_
Password:
hduser_@guru99-VirtualBox:~$
```

Step 1)

Create a new directory with name **MapReduceTutorial** as shown in the below MapReduce example

```
sudo mkdir MapReduceTutorial
```

```
hduser_@guru99-VirtualBox:~$ sudo mkdir MapReduceTutorial
```

Give permissions

```
sudo chmod -R 777 MapReduceTutorial
```

```
hduser_@guru99-VirtualBox:~$ sudo chmod -R 777 MapReduceTutorial
```

SalesMapper.java

```
package
```

```
SalesCountry; import java.io.
```

```
IOException;
```

```
import  
org.apache.hadoop.io.IntWritable; import org  
.apache.hadoop.io.LongWritable; import org.  
apache.hadoop.io.Text;  
import org.apache.hadoop.mapred.*;
```

```
public class SalesMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text  
, IntWritable> {  
    private final static IntWritable one = new IntWritable(1);  
  
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> out  
put, Reporter reporter) throws IOException {
```

```
        String valueString = value.toString();  
        String[] SingleCountryData =  
        valueString.split(","); output.collect(new Text(SingleC  
ountryData[7]), one);
```

```
    }  
}
```

SalesCountryReducer.java

```
package SalesCountry;
```

```
import  
java.io.IOException; import  
java.util.*;
```

```
import org.apache.hadoop.io.IntWritable; i  
mport org.apache.hadoop.io.Text; import o
```

```

    rg.apache.hadoop.mapred.*;

public class SalesCountryReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text t_key, Iterator<IntWritable>
values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {

        Text key = t_key;
        int frequencyForCountry = 0;
        while (values.hasNext()) {
            // replace type of value with the actual type of our value
            IntWritable value = (IntWritable) values.next();
            frequencyForCountry += value.get();
        }
        output.collect(key, new IntWritable(frequencyForCountry));
    }
}

```

SalesCountryDriver.java

```

package SalesCountry;

import
org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

public class SalesCountryDriver {
    public static void main(String[] args)
    {
        JobClient my_client = new JobClient();
        // Create a configuration object for the job
        JobConf job_conf = new JobConf(SalesCountryDriver.class);

        // Set a name of the
        JobConf job_conf.setJobName("SalesPerCountry");

        // Specify data type of output key and value
        job_conf.setOutputKeyClass(Text.class);
        job_conf.setOutputValueClass(IntWritable.class);

        // Specify names of Mapper and Reducer
        job_conf.setMapperClass(SalesCountry.SalesMapper.class);
        job_conf.setReducerClass(SalesCountry.SalesCountryReducer.class);

        // Specify formats of the data type of Input and output
        job_conf.setInputFormat(TextInputFormat.class);
        job_conf.setOutputFormat(TextOutputFormat.class);

        // Set input and output directories using command line arguments,
    }
}

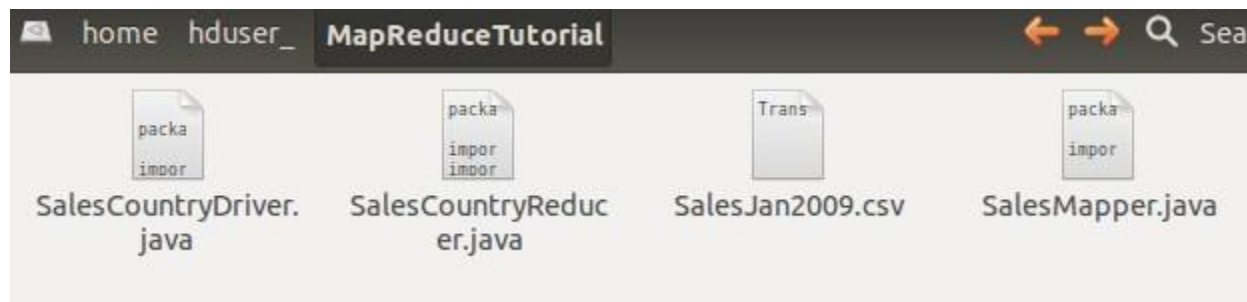
```

//arg[0]=nameofinputdirectoryonHDFS,andarg[1]=nameofoutputdirectorytobecreated to store the output file.

```
FileInputFormat.setInputPaths(job_conf, new
Path(args[0]));FileOutputFormat.setOutputPath(job_conf,new
Path(args[1]));

my_client.setConf(job_conf);
try {
    // Run the
    jobJobClient.runJob(job_co
nf);
} catch (Exception e)
    {e.printStackTrace()
    };
}
}
```

[DownloadFilesHere](#)



Check the file permissions of all these files

```
hduser_@guru99-VirtualBox:~/MapReduceTutorial$ ls -al
total 144
drwxrwxrwx 2 root    root    4096 May  5 15:00 .
drwxr-xr-x 6 hduser_ hadoop_ 4096 May  5 14:53 ..
-rw-rw-r-- 1 guru99  guru99  1367 May  5 02:28 SalesCountryDriver.java
-rw-rw-r-- 1 guru99  guru99   749 May  5 02:28 SalesCountryReducer.java
-rw-rw-r-- 1 guru99  guru99 123637 May  5 02:28 SalesJan2009.csv
-rw-rw-r-- 1 guru99  guru99   659 May  5 02:28 SalesMapper.java
```

and if 'read' permissions are missing then grant the same-

```
hduser_@guru99-VirtualBox:~/MapReduceTutorial$ sudo chmod +r *.*
```

Step 2)

Export classpath as shown in the below Hadoop example

export

```
CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.2.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-common-2.2.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-common-2.2.0.jar:~/MapReduceTutorial/SalesCountry/*:$HADOOP_HOME/lib/*"
```

```
hduser_@guru99-VirtualBox:~/MapReduceTutorial$ export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.2.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-common-2.2.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-common-2.2.0.jar:~/MapReduceTutorial/SalesCountry/*:$HADOOP_HOME/lib/*"
hduser_@guru99-VirtualBox:~/MapReduceTutorial$
```

Step 3)

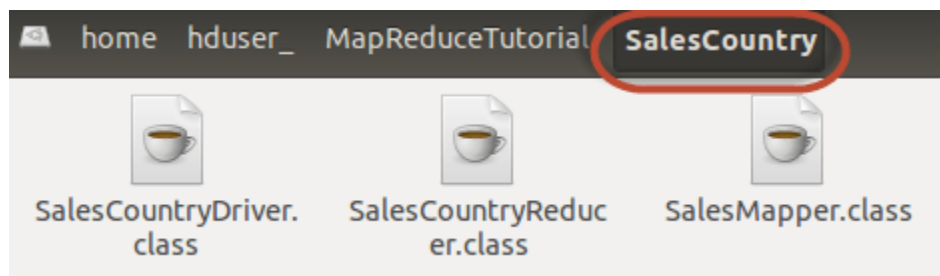
Compile **Java** files (these files are present in directory **Final-MapReduceHandsOn**). Its classfiles will be put in the package directory

```
javac -d SalesMapper.javaSalesCountryReducer.javaSalesCountryDriver.java
```

```
hduser_@guru99-VirtualBox:~/MapReduceTutorial$ javac -d . SalesMapper.java SalesCountryReducer.java SalesCountryDriver.java
/home/guru99/Downloads/hadoop/share/hadoop/common/hadoop-common-2.2.0.jar(org/apache/hadoop/fs/Path.class)
: warning: Cannot find annotation method 'value()' in type 'LimitedPrivate': class file for org.apache.hadoop.classification.InterfaceAudience not found
1 warning
hduser_@guru99-VirtualBox:~/MapReduceTutorial$
```

This warning can be safely ignored.

This compilation will create a directory in a current directory named with package names specified in the java source file (i.e. **SalesCountry** in our case) and put all compiled class files in it.

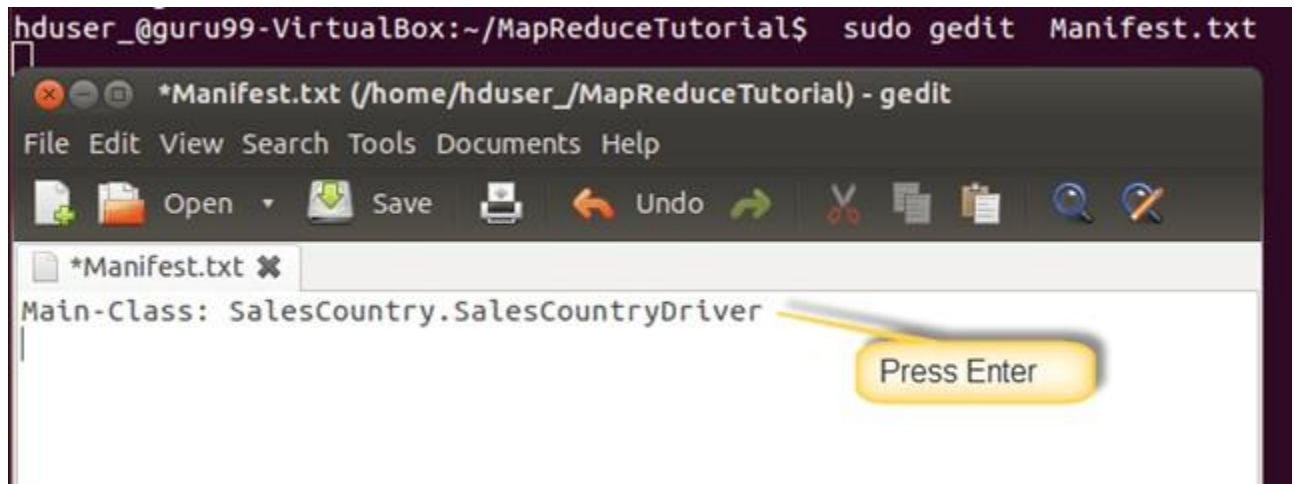


Step 4)

Create a new file **Manifest.txt**

```
sudo gedit
Manifest.txt add following
lines to it,
```

```
Main-Class:SalesCountry.SalesCountryDriver
```



SalesCountry.SalesCountryDriver is the name of the main class. Please note that you have to hit the enter key at the end of this line.

Step 5)

Create a Jar file

jar cfm ProductSalePerCountry.jar Manifest.txt SalesCountry/*.class

```
hduser_@guru99-VirtualBox:~/MapReduceTutorial$ jar cfm ProductSalePerCountry.jar Manifest.txt SalesCountry/*.class
```

Check that the jar file is created

```
hduser_@guru99-VirtualBox:~/MapReduceTutorial$ ls
Manifest.txt          SalesCountry          SalesCountryReducer.java  SalesMapper.java
ProductSalePerCountry.jar  SalesCountryDriver.java  SalesJan2009.csv
```

Step 6)

Start Hadoop

\$HADOOP_HOME/sbin/start-dfs.sh

\$HADOOP_HOME/sbin/start-yarn.sh

Step 7)

Copy the File **SalesJan2009.csv** into **~/inputMapReduce**

Now Use below command to copy **~/inputMapReduce** to HDFS.

\$HADOOP_HOME/bin/hdfs dfs -copyFromLocal ~/inputMapReduce /


```

hduser@guru99: ~/MapReduceTutorial
hduser@guru99:~/MapReduceTutorial$ $HADOOP_HOME/bin/hdfs dfs -copyFromLocal ~/inputMapReduce /
14/05/06 23:33:48 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hduser@guru99:~/MapReduceTutorial$

```

We can safely ignore this warning.

Verify whether a file is actually copied or not.

`$HADOOP_HOME/bin/hdfs dfs -ls /inputMapReduce`

```

hduser@guru99:~/MapReduceTutorial$ $HADOOP_HOME/bin/hdfs dfs -ls /inputMapReduce
14/05/06 23:35:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--  1 hduser supergroup      123637 2014-05-06 23:33 /inputMapReduce/SalesJan2009.csv
hduser@guru99:~/MapReduceTutorial$

```

Step 8)

Run MapReduce job

`$HADOOP_HOME/bin/hadoop jar ProductSalePerCountry.jar/inputMapReduce /mapreduce_output_sales`

```

hduser@guru99: ~/MapReduceTutorial
hduser@guru99:~/MapReduceTutorial$ $HADOOP_HOME/bin/hadoop jar ProductSalePerCountry.jar /inputMapReduce /mapreduce_output_sales

```

This will create an output directory named `mapreduce_output_sales` on HDFS. Contents of this directory will be a file containing product sales per country.

Step 9)

The result can be seen through command interface as,

`$HADOOP_HOME/bin/hdfs dfs -cat /mapreduce_output_sales/part-00000`

```
hduser@guru99: ~/MapReduceTutorial
hduser@guru99:~/MapReduceTutorial$ $HADOOP_HOME/bin/hdfs dfs -cat /mapreduce_output_sales/part-00000
14/05/02 13:03:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Argentina      1
Australia      38
Austria 7
Bahrain 1
Belgium 8
Bermuda 1
Brazil 5
Bulgaria      1
CO            1
Canada 76
Cayman Isls   1
```

Results can also be seen via a web interface as-

Open in a web browser.

The screenshot shows a web browser window with the address bar displaying 'localhost:50070/dfshealth.jsp'. The page title is 'NameNode 'localhost:54310' (active)'. Below the title, there is a table with the following information:

| | |
|----------------|---|
| Started: | Fri May 02 12:33:35 IST 2014 |
| Version: | 2.2.0, 1529768 |
| Compiled: | 2013-10-07T06:28Z by hortonmu from branch-2.2.0 |
| Cluster ID: | CID-a1832593-cb99-4642-b3a5-043b8e204dbb |
| Block Pool ID: | BP-657563107-127.0.1.1-1398775824455 |

Below the table, there are two links: [Browse the filesystem](#) and [NameNode Logs](#).

The 'Cluster Summary' section indicates that security is OFF and lists 13 files and directories, totaling 4 blocks. It also provides memory usage statistics: Heap Memory used is 30.93 MB (27% of committed), and Non Heap Memory used is 36.84 MB (98% of committed).

| | | |
|---------------------|---|----------|
| Configured Capacity | : | 35.26 GB |
| DFS Used | : | 300 KB |
| Non DFS Used | : | 6.62 GB |
| DFS Remaining | : | 28.64 GB |

Now select 'Browse the filesystem' and navigate to /mapreduce_output_sales

HDFS:/mapreduce_output_sales

localhost:50075/browseDirectory.jsp?dir=%2Fmapreduce_output_sales&namenodeinfoPort=50070&nnaddr=127.0.0.1:5

Contents of directory /mapreduce_output_sales

Goto : go

[Go to parent directory](#)

| Name | Type | Size | Replication | Block Size | Modification Time | Permission | Owner | Group |
|----------------------------|------|-------|-------------|------------|-------------------|------------|--------|------------|
| _SUCCESS | file | 0 B | 1 | 128 MB | 2014-05-02 12:58 | rw-r--r-- | hduser | supergroup |
| part-00000 | file | 661 B | 1 | 128 MB | 2014-05-02 12:58 | rw-r--r-- | hduser | supergroup |

[Go back to DFS home](#)

Local logs

[Log directory](#)

[Hadoop](#), 2014.

Open [part-r-00000](#)

HDFS:/mapreduce_output_sal...

localhost:50075/browseBlock.jsp?blockId=1073741836&blockSize=661&genstamp=1012&filename=%2Fmapreduce_outj

File: /mapreduce_output_sales/part-00000

Goto : go

[Go back to dir listing](#)

[Advanced view/download options](#)

```

Argentina      1
Australia     38
Austria       7
Bahrain        1
Belgium        8
Bermuda        1
Brazil         5
Bulgaria        1
CO              1
Canada        76
Cayman Isls    1
China          1
Costa Rica     1
Country        1
Czech Republic 3
Denmark        15
Dominican Republic 1
Finland        2
France         27
Germany        25
Greece         1
Guatemala      1
Hong Kong      1

```