# Bigdata Assignment 1.7

We have a dataset of sales of different TV sets across different locations.
Records look like:
Samsung|Optima|14|Madhya Pradesh|132401|14200
The fields are arranged like:
Company Name|Product Name|Size in inches|State|Pin Code|Price
There are some invalid records which contain 'NA' in either Company Name or Product Name.
2. Write a Map Reduce program to calculate the total units sold for each Company.
3. Write a Map Reduce program to calculate the total units sold in each state for Onida
company.

Solution -

**2.**

**Driver code -**

```
package mapreduce;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

public class Task2 {
	public static void main(String[] args) throws Exception {
		Configuration conf = new Configuration();
		Job job = new Job(conf, "Task2");
		job.setJarByClass(Task2.class);
```

```
            //Key is text as it is the company_name
              job.setMapOutputKeyClass(Text.class);
              //Key is LongWritable as it is the no of units
              job.setMapOutputValueClass(LongWritable.class);

            //Key is text as it is the company_name
              job.setOutputKeyClass(Text.class);
              //Key is LongWritable as it is the no of units
              job.setOutputValueClass(LongWritable.class);
              job.setMapperClass(Task2mapper.class);
              job.setReducerClass(Task2reducer.class);

              job.setInputFormatClass(TextInputFormat.class);
              job.setOutputFormatClass(TextOutputFormat.class);

              FileInputFormat.addInputPath(job, new Path(args[0]));
              FileOutputFormat.setOutputPath(job,new Path(args[1]));

              /*
              Path out=new Path(args[1]);
              out.getFileSystem(conf).delete(out);
              */

              job.waitForCompletion(true);
        }
}
```

**Mapper Code -**

```
package mapreduce;

import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;

public class Task2mapper extends Mapper<LongWritable, Text, Text,
LongWritable> {

        @Override
```

```
	public void map(LongWritable key, Text value, Context context)
			throws IOException, InterruptedException {
		String[] lineArray = value.toString().split("\\|");
		//Checking if company name or product name must not equal to NA
		if(!(lineArray[0].equals("NA")||lineArray[1].equals("NA")))
		{
			context.write(new Text(lineArray[0]), new LongWritable(1));
		}

	}
}
```

**Reducer Code -**

```
package mapreduce;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class Task2reducer extends Reducer<Text, LongWritable, Text,
LongWritable>
{

	@Override
	public void reduce(Text key, Iterable<LongWritable> values,Context
context) throws IOException, InterruptedException
	{
		//Summing the values  that is the no of units for a particular key
which is the company name.
		long totalSales = 0;
		for(LongWritable value:values)
		{
			totalSales+= value.get();
		}
		context.write(key,new LongWritable(totalSales));
	}
}
```

Applications Places System Fri May 18, 8:22 PM Acadgild

acadgild@localhost:~/RITESH/A_1.7

```
[acadgild@localhost A_1.7]$ ls
assignment1.7  assignment1.7.jar
[acadgild@localhost A_1.7]$ hadoop fs -put assignment1.7.jar  /user/acadgild/hadoop
18/05/18 20:19:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
[acadgild@localhost A_1.7]$ hadoop fs -ls /user/acadgild/hadoop
18/05/18 20:20:12 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 6 items
-rw-r--r--   1 acadgild supergroup       2800 2018-05-18 13:33 /user/acadgild/hadoop/assignment1.6.jar
drwxr-xr-x   - acadgild supergroup          0 2018-05-18 13:34 /user/acadgild/hadoop/assignment1.6_output
-rw-r--r--   1 acadgild supergroup       3830 2018-05-18 20:20 /user/acadgild/hadoop/assignment1.7.jar
-rwxrwx---   1 acadgild supergroup        168 2018-05-17 13:31 /user/acadgild/hadoop/max-temp.txt
-rw-r--r--   1 acadgild supergroup        733 2018-05-18 13:33 /user/acadgild/hadoop/television.txt
-rw-r--r--   1 acadgild supergroup        227 2018-05-17 01:02 /user/acadgild/hadoop/word-count.txt
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost A_1.7]$ hadoop jar assignment1.7.jar mapreduce.Task2 /user/acadgild/hadoop/television.txt /user/acadgild/
hadoop/assignment1.7_output
18/05/18 20:20:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/05/18 20:20:28 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/05/18 20:20:29 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in
terface and execute your application with ToolRunner to remedy this.
18/05/18 20:20:29 INFO input.FileInputFormat: Total input paths to process : 1
18/05/18 20:20:29 INFO mapreduce.JobSubmitter: number of splits:1
18/05/18 20:20:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1526626806481_0011
18/05/18 20:20:30 INFO impl.YarnClientImpl: Submitted application application_1526626806481_0011
18/05/18 20:20:30 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1526626806481_0011/
18/05/18 20:20:30 INFO mapreduce.Job: Running job: job_1526626806481_0011
18/05/18 20:20:37 INFO mapreduce.Job: Job job_1526626806481_0011 running in uber mode : false
18/05/18 20:20:37 INFO mapreduce.Job:  map 0% reduce 0%
18/05/18 20:20:43 INFO mapreduce.Job:  map 100% reduce 0%
18/05/18 20:20:49 INFO mapreduce.Job:  map 100% reduce 100%
18/05/18 20:20:49 INFO mapreduce.Job: Job job_1526626806481_0011 completed successfully
18/05/18 20:20:49 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=268
                FILE: Number of bytes written=216603
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
```

Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox

Applications Places System Fri May 18, 8:23 PM Acadgild

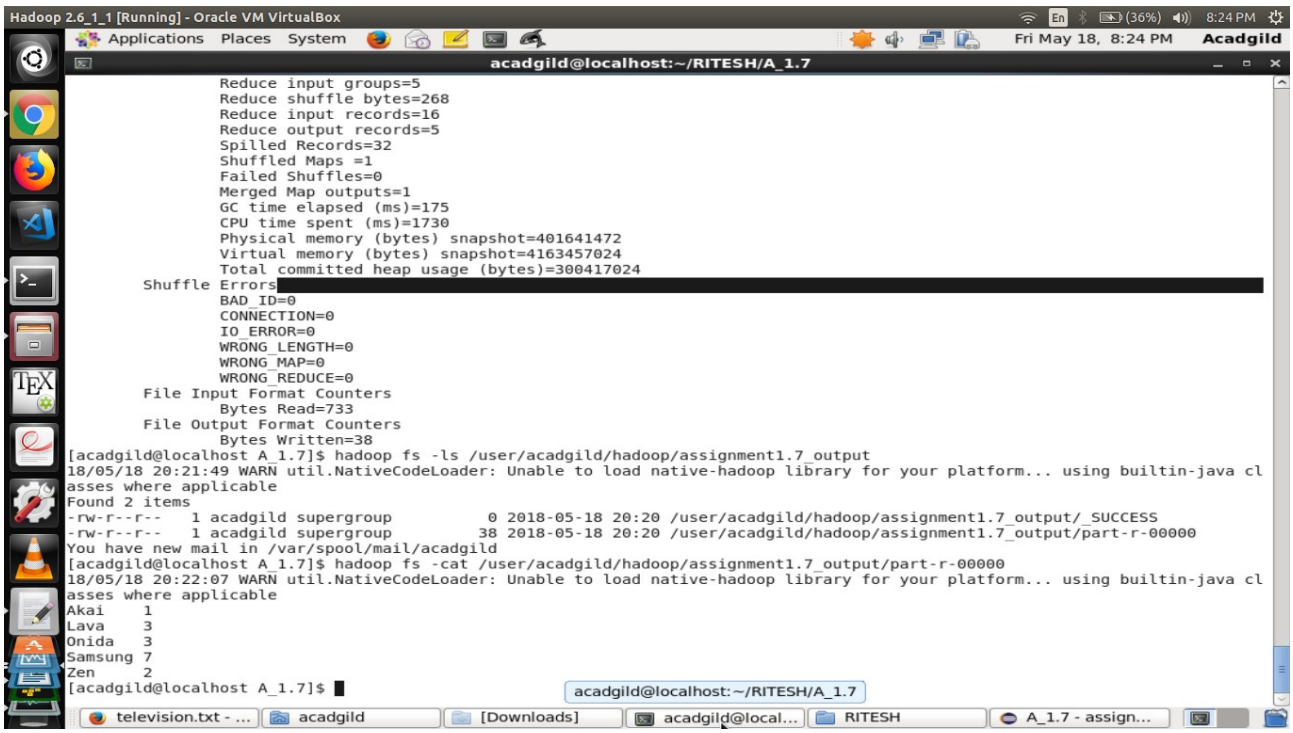
acadgild@localhost:~/RITESH/A_1.7

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=855
                HDFS: Number of bytes written=38
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3747
                Total time spent by all reduces in occupied slots (ms)=4098
                Total time spent by all map tasks (ms)=3747
                Total time spent by all reduce tasks (ms)=4098
                Total vcore-milliseconds taken by all map tasks=3747
                Total vcore-milliseconds taken by all reduce tasks=4098
                Total megabyte-milliseconds taken by all map tasks=3836928
                Total megabyte-milliseconds taken by all reduce tasks=4196352
        Map-Reduce Framework
                Map input records=18
                Map output records=16
                Map output bytes=230
                Map output materialized bytes=268
                Input split bytes=122
                Combine input records=0
                Combine output records=0
                Reduce input groups=5
                Reduce shuffle bytes=268
                Reduce input records=16
                Reduce output records=5
                Spilled Records=32
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=175
                CPU time spent (ms)=1730
                Physical memory (bytes) snapshot=401641472
                Virtual memory (bytes) snapshot=4163457024
                Total committed heap usage (bytes)=300417024
```

television.txt - ... acadgild [Downloads] acadgild@local... RITESH A_1.7 - assign...

**Execution Steps:-**

• television.txt was put to hdfs.
• After the driver code and mapping code , a jar file of was generated.
• assignmen1.7.jar file(mapping code) was put to hdfs.
• Mapping task was performed by the following command – **hadoop jar assignmen1.7.jar mapreduce.Task2 /user/acadgild/hadoop/television.txt /user/acadgild/hadoop/assignment1.7_output**
• To see the content of the output following command was used - **hadoop cat /user/acadgild//hadoop/assignment1.7_output/part-r-00000**




**Output**

As in the above screen shot ,it is evident that the content of the output file shows the no of units sold by each company.

3.

Driver Code-

```
package mapreduce;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

public class Task3 {
	public static void main(String[] args) throws Exception {
		Configuration conf = new Configuration();
		Job job = new Job(conf, "Task3");
		job.setJarByClass(Task2.class);

	      //Key is text as it is the state name
		job.setMapOutputKeyClass(Text.class);
		//Key is LongWritable as it is the no of units  of Onida
		job.setMapOutputValueClass(LongWritable.class);

	      //Key is text as it is the state name
		job.setOutputKeyClass(Text.class);
		//Key is LongWritable as it is the no of units  of Onida
		job.setOutputValueClass(LongWritable.class);
		job.setMapperClass(Task3mapper.class);
		job.setReducerClass(Task3reducer.class);

		job.setInputFormatClass(TextInputFormat.class);
		job.setOutputFormatClass(TextOutputFormat.class);

		FileInputFormat.addInputPath(job, new Path(args[0]));
		FileOutputFormat.setOutputPath(job,new Path(args[1]));

		/*
		Path out=new Path(args[1]);
```

```
		out.getFileSystem(conf).delete(out);
		*/

		job.waitForCompletion(true);
	}
}
```

**Mapper Code-**

```
package mapreduce;

import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;

public class Task3mapper extends Mapper<LongWritable, Text, Text,
LongWritable> {

	@Override
	public void map(LongWritable key, Text value, Context context)
			throws IOException, InterruptedException {
		String[] lineArray = value.toString().split("\\|");
		//Checking if company name is Ondia or product name must not
equal to NA
		if((lineArray[0].equals("Onida")&& !lineArray[1].equals("NA")))
		{
			context.write(new Text(lineArray[3]), new LongWritable(1));
		}

	}
}
```

**Reducer Code -**

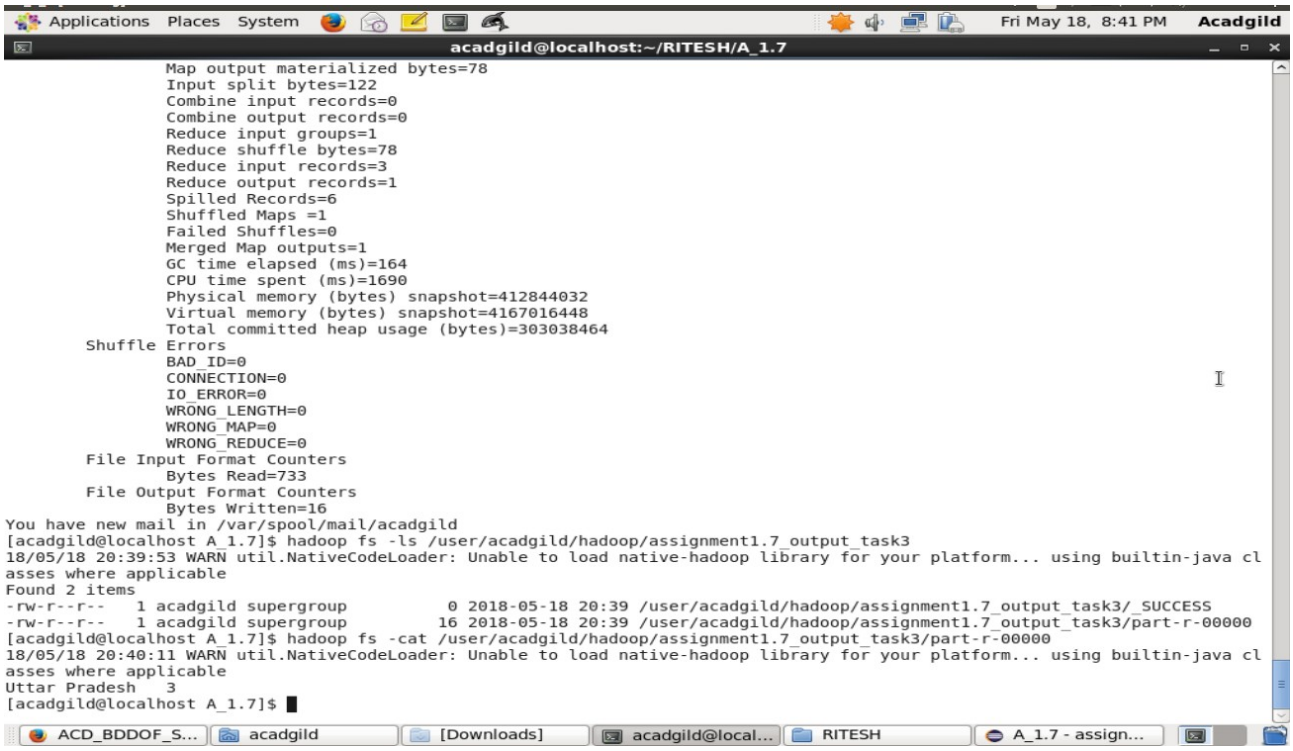
```
package mapreduce;

import java.io.IOException;
```

```
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class Task3reducer extends Reducer<Text, LongWritable, Text,
LongWritable>
{

	@Override
	public void reduce(Text key, Iterable<LongWritable> values,Context
context) throws IOException, InterruptedException
	{
		//Summing the values no of units for a particular key which is the
state name.
			long totalSales = 0;
			for(LongWritable value:values)
			{
				totalSales+= value.get();
			}
			context.write(key,new LongWritable(totalSales));
	}
}
```

```
Applications  Places  System                                   Fri May 18, 8:40 PM   Acadgild
acadgild@localhost:~/RITESH/A_1.7
[acadgild@localhost A_1.7]$ hadoop jar assignment1.7.jar mapreduce.Task3 /user/acadgild/hadoop/television.txt /user/acadgild/
hadoop/assignment1.7_output_task3
18/05/18 20:39:03 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/05/18 20:39:04 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/05/18 20:39:05 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in
terface and execute your application with ToolRunner to remedy this.
18/05/18 20:39:05 INFO input.FileInputFormat: Total input paths to process : 1
18/05/18 20:39:06 INFO mapreduce.JobSubmitter: number of splits:1
18/05/18 20:39:06 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1526626806481_0013
18/05/18 20:39:06 INFO impl.YarnClientImpl: Submitted application application_1526626806481_0013
18/05/18 20:39:06 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1526626806481_0013/
18/05/18 20:39:06 INFO mapreduce.Job: Running job: job_1526626806481_0013
18/05/18 20:39:13 INFO mapreduce.Job: Job job_1526626806481_0013 running in uber mode : false
18/05/18 20:39:13 INFO mapreduce.Job:  map 0% reduce 0%
18/05/18 20:39:19 INFO mapreduce.Job:  map 100% reduce 0%
18/05/18 20:39:25 INFO mapreduce.Job:  map 100% reduce 100%
18/05/18 20:39:25 INFO mapreduce.Job: Job job_1526626806481_0013 completed successfully
18/05/18 20:39:26 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=78
                FILE: Number of bytes written=216235
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=855
                HDFS: Number of bytes written=16
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3518
                Total time spent by all reduces in occupied slots (ms)=4047
                Total time spent by all map tasks (ms)=3518
                Total time spent by all reduce tasks (ms)=4047
                Total vcore-milliseconds taken by all map tasks=3518
                Total vcore-milliseconds taken by all reduce tasks=4047
```

**Execution Steps:-**

• television.txt was put to hdfs.
• After the driver code and mapping code , a jar file of was generated.
• assignmen1.7.jar file(mapping code) was put to hdfs.
• Mapping task was performed by the following command – **hadoop jar assignmen1.7.jar mapreduce.Task3 /user/acadgild/hadoop/television.txt /user/acadgild/hadoop/assignment1.7_output_task3**
• To see the content of the output following command was used - **hadoop cat /user/acadgild//hadoop/assignment1.7_output_task3/part-r-00000**




**Output**

As in the above screen shot ,it is evident that the content of the output file shows the no of units sold by company Onida in each state.