Assignment 4:

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.

Raw Text:

```
Samsung|Optima|14|Madhya Pradesh|132401|14200
Onida|Lucid|18|Uttar Pradesh|232401|16200
Akai|Decent|16|Kerala|922401|12200
Lava|Attention|20|Assam|454601|24200
Zen|Super|14|Maharashtra|619082|9200
Samsung|Optima|14|Madhya Pradesh|132401|14200
Onida|Lucid|18|Uttar Pradesh|232401|16200
Onida|Decent|14|Uttar Pradesh|232401|16200
Onida|NA|16|Kerala|922401|12200
Lava|Attention|20|Assam|454601|24200
Zen|Super|14|Maharashtra|619082|9200
Samsung|Optima|14|Madhya Pradesh|132401|14200
NA|Lucid|18|Uttar Pradesh|232401|16200
Samsung|Decent|16|Kerala|922401|12200
Lava|Attention|20|Assam|454601|24200
Samsung|Super|14|Maharashtra|619082|9200
Samsung|Super|14|Maharashtra|619082|9200
Samsung|Super|14|Maharashtra|619082|9200
~
~
~
```

Task 1: Write a Map Reduce program to filter out the invalid records. Map only job will fit for this context.

Code:

```
package assgn_filter_na;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
```

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;



public class Filter_NA {

        public static class ProductMapper
          extends Mapper<Object, Text, Text, IntWritable>
        {

                private final static IntWritable one = new IntWritable(1);
                Text myText = new Text();
                public void map(Object key, Text value, Context context)
                        throws IOException, InterruptedException
                        {
                         String allData = value.toString();             //Read all data
from file
                         String[] allLines = allData.split("\n");       //Split the file
by new line (\n)

                         for (String s: allLines) // process each line in the file
                         {
                                 String[] line = s.split("\\|");  //Parse each line by
pipe separator
                                 System.out.println(line);        //Used for debugging
but learned it does not work in Hadoop
                                 if (!line[0].toString().equals("NA") &&
!line[1].toString().equals("NA")) //Identify all lines with name (field 0) and product
(field 1) not equal to NA
                                 {
                                         Text lineOut = new Text(s); // Convert string to
text
                                         myText.set(lineOut);            // Set line to
Text function myTest
                                         System.out.println(myText); //Again for
debugging
                                         context.write(myText, one); // Write Context
out

                                 }
                         }

                        }
        }

        public static class ProductReducer
        extends Reducer<Text, IntWritable, Text, IntWritable>
{
                private final static IntWritable one = new IntWritable(1);
                Text Value = new Text();
        public void reduce(Text key, Iterable<IntWritable>values, Context context)
                throws IOException, InterruptedException
                {

                        context.write(key,  one); //Pass through code since assignment
was to filter NA from Company Name and Product Name

                }
}

        public static void main(String[] args) throws Exception {
                Configuration conf = new Configuration();
                Job job = Job.getInstance(conf, "filter NA");
                job.setJarByClass(Filter_NA.class);
                job.setMapperClass(ProductMapper.class);
```

```
                job.setCombinerClass(ProductReducer.class);
                job.setReducerClass(ProductReducer.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                System.exit(job.waitForCompletion(true)?0:1);;

        }

}
~

~
~
```

Command:

```
[acadgild@localhost Desktop]$ hadoop jar assgn_filter_na.jar /television.txt /assign_filter_out
```

Successful Execution:

```
acadgild@localhost:~/Desktop
18/09/27 04:11:25 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=373
                FILE: Number of bytes written=215709
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=834
                HDFS: Number of bytes written=335
                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)=9104
                Total time spent by all reduces in occupied slots (ms)=6828
                Total time spent by all map tasks (ms)=9104
                Total time spent by all reduce tasks (ms)=6828
                Total vcore-milliseconds taken by all map tasks=9104
                Total vcore-milliseconds taken by all reduce tasks=6828
                Total megabyte-milliseconds taken by all map tasks=9322496
                Total megabyte-milliseconds taken by all reduce tasks=6991872
        Map-Reduce Framework
                Map input records=18
                Map output records=16
                Map output bytes=710
                Map output materialized bytes=373
                Input split bytes=101
                Combine input records=16
                Combine output records=8
                Reduce input groups=8
                Reduce shuffle bytes=373
                Reduce input records=8
                Reduce output records=8
                Spilled Records=16
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=182
                CPU time spent (ms)=1800
                Physical memory (bytes) snapshot=286240768
                Virtual memory (bytes) snapshot=4118188032
                Total committed heap usage (bytes)=170004480
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=733
        File Output Format Counters
                Bytes Written=335
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$ |
```

Output: Note that the “NA” are filtered out

```
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$ hadoop dfs -cat /assign_filter_out/part-r-00000
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

18/09/27 04:12:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
Akai|Decent|16|Kerala|922401|12200      1
Lava|Attention|20|Assam|454601|24200    1
Onida|Decent|14|Uttar Pradesh|232401|16200      1
Onida|Lucid|18|Uttar Pradesh|232401|16200       1
Samsung|Decent|16|Kerala|922401|12200   1
Samsung|Optima|14|Madhya Pradesh|132401|14200   1
Samsung|Super|14|Maharashtra|619082|9200        1
Zen|Super|14|Maharashtra|619082|9200    1
[acadgild@localhost Desktop]$ |
```

Task 2: Write a Map Reduce program to calculate the total units sold for each Company.

Code:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class Calculate_Tot_Units {

        public static class UnitsMapper
          extends Mapper<Object, Text, Text, IntWritable>
        {

                private final static IntWritable one = new IntWritable(1);
                Text myText = new Text();

                public void map(Object key, Text value, Context context)
                        throws IOException, InterruptedException
                        {
                         String allData = value.toString();             //Read all data
from file
                         String[] allLines = allData.split("\n");       //Split the file
by new line (\n)

                         for (String s: allLines) // process each line in the file
                         {
                                 String[] line = s.split("\\|");  //Parse each line by
pipe separator
                                 String sCompany = line[0].toString(); // Get Company
Name from Field 1
                                 Text sOut = new Text(sCompany); // Convert string to text
                             myText.set(sOut);           // Set line to Text function
myTest
                             context.write(myText, one); // Write Context out
                         }

                        }
        }

        public static class UnitsReducer
        extends Reducer<Text, IntWritable, Text, IntWritable>
{
                private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable>values, Context context)
                throws IOException, InterruptedException
                {

                int sum= 0;
                for(IntWritable val: values)
                {
                        sum += val.get(); //Add each value based on the key Company name

                }
                result.set(sum);
                context.write(key, result);
```

```
                }
}


        public static void main(String[] args) throws Exception {
                Configuration conf = new Configuration();
                Job job = Job.getInstance(conf, "Unit Counter");
                job.setJarByClass(Calculate_Tot_Units.class);
                job.setMapperClass(UnitsMapper.class);
                job.setCombinerClass(UnitsReducer.class);
                job.setReducerClass(UnitsReducer.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                System.exit(job.waitForCompletion(true)?0:1);;


        }

}
~
~
"Calculate_Tot_Units.java" 81 lines, 2497 characters
```

Command:

```
[acadgild@localhost Desktop]$ hadoop jar UnitCounter.jar /television.txt /Assign4Task2
```

Successful Execution:

```
[acadgild@localhost Desktop]$ hadoop jar UnitCounter.jar /television.txt /Assign4Task2
18/09/27 06:37:20 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
18/09/27 06:37:21 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/09/27 06:37:22 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execut
e your application with ToolRunner to remedy this.
18/09/27 06:37:23 INFO input.FileInputFormat: Total input paths to process : 1
18/09/27 06:37:23 INFO mapreduce.JobSubmitter: number of splits:1
18/09/27 06:37:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1537999048223_0007
18/09/27 06:37:24 INFO impl.YarnClientImpl: Submitted application application_1537999048223_0007
18/09/27 06:37:24 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1537999048223_0007/
18/09/27 06:37:24 INFO mapreduce.Job: Running job: job_1537999048223_0007
18/09/27 06:37:34 INFO mapreduce.Job: Job job_1537999048223_0007 running in uber mode : false
18/09/27 06:37:34 INFO mapreduce.Job:  map 0% reduce 0%
18/09/27 06:37:43 INFO mapreduce.Job:  map 100% reduce 0%
18/09/27 06:37:52 INFO mapreduce.Job:  map 100% reduce 100%
18/09/27 06:37:52 INFO mapreduce.Job: Job job_1537999048223_0007 completed successfully
18/09/27 06:37:52 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=73
                FILE: Number of bytes written=215057
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=834
                HDFS: Number of bytes written=43
                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)=6334
                Total time spent by all reduces in occupied slots (ms)=6162
                Total time spent by all map tasks (ms)=6334
                Total time spent by all reduce tasks (ms)=6162
                Total vcore-milliseconds taken by all map tasks=6334
                Total vcore-milliseconds taken by all reduce tasks=6162
                Total megabyte-milliseconds taken by all map tasks=6486016
                Total megabyte-milliseconds taken by all reduce tasks=6309888
        Map-Reduce Framework
                Map input records=18
                Map output records=18
                Map output bytes=183
                Map output materialized bytes=73
                Input split bytes=101
                Combine input records=18
                Combine output records=6
                Reduce input groups=6
                Reduce shuffle bytes=73
                Reduce input records=6
                Reduce output records=6
                Spilled Records=12
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=152
                CPU time spent (ms)=1670
                Physical memory (bytes) snapshot=292900864
                Virtual memory (bytes) snapshot=4118200320
                Total committed heap usage (bytes)=170004480
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=733
        File Output Format Counters
                Bytes Written=43
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$ |
```

Results:

```
[acadgild@localhost Desktop]$ hadoop dfs -cat /Assign4Task2/part-r-00000
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

18/09/27 06:38:42 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Akai    1
Lava    3
NA      1
Onida   4
Samsung 7
Zen     2
[acadgild@localhost Desktop]$ |
```

Task 3: Write a Map Reduce program to calculate the total units sold in each state for Onida

company.

Code:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.Mapper.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class OnidaByState {

        public static class OnidaMapper
          extends Mapper<Object, Text, Text, IntWritable>
        {

                private final static IntWritable one = new IntWritable(1);
                Text myText = new Text();

                public void map(Object key, Text value, Context context)
                        throws IOException, InterruptedException
                        {
                         String allData = value.toString();             //Read all data
from file
                         String[] allLines = allData.split("\n");       //Split the file
by new line (\n)

                         for (String s: allLines) // process each line in the file
                         {
                                 String[] line = s.split("\\|");  //Parse each line by
pipe separator
                                 String sCompany = line[0].toString(); // Get Company
Name from Field 1
                                 String sState = line[3].toString(); //Get State
                                 if (line[0].toString().equals("Onida"))
                                 {
                                         Text sOut = new Text(sState); // Convert State
string to text
                                     myText.set(sOut);           // Set line to Text
function myTest
                                     context.write(myText, one); // Write Context out
                                 }

                         }

                        }
        }

        public static class OnidaReducer
        extends Reducer<Text, IntWritable, Text, IntWritable>
{
                private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable>values, Context context)
                throws IOException, InterruptedException
                {

                int sum= 0;
                for(IntWritable val: values)
                {
                        sum += val.get(); //Add each value based on the key State

                }
```

```
                result.set(sum);
                context.write(key, result);

                }
}

        public static void main(String[] args) throws Exception {
                Configuration conf = new Configuration();
                Job job = Job.getInstance(conf, "Onida Count by State");
                job.setJarByClass(OnidaByState.class);
                job.setMapperClass(OnidaMapper.class);
                job.setCombinerClass(OnidaReducer.class);
                job.setReducerClass(OnidaReducer.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                System.exit(job.waitForCompletion(true)?0:1);;

        }

}
"OnidaByState.java" 83 lines, 2614 characters
```

Command:

```
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$ hadoop jar OnidaCounter.jar /television.txt /Assign4Task3
```

Successful Execution:

```
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$ hadoop jar OnidaCounter.jar /television.txt /Assign4Task3
18/09/27 07:00:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
18/09/27 07:00:54 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/09/27 07:00:55 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execut
e your application with ToolRunner to remedy this.
18/09/27 07:00:56 INFO input.FileInputFormat: Total input paths to process : 1
18/09/27 07:00:56 INFO mapreduce.JobSubmitter: number of splits:1
18/09/27 07:00:56 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1537999048223_0009
18/09/27 07:00:57 INFO impl.YarnClientImpl: Submitted application application_1537999048223_0009
18/09/27 07:00:57 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1537999048223_0009/
18/09/27 07:00:57 INFO mapreduce.Job: Running job: job_1537999048223_0009
18/09/27 07:01:07 INFO mapreduce.Job: Job job_1537999048223_0009 running in uber mode : false
18/09/27 07:01:07 INFO mapreduce.Job:  map 0% reduce 0%
18/09/27 07:01:14 INFO mapreduce.Job:  map 100% reduce 0%
18/09/27 07:01:22 INFO mapreduce.Job:  map 100% reduce 100%
18/09/27 07:01:23 INFO mapreduce.Job: Job job_1537999048223_0009 completed successfully
18/09/27 07:01:24 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=39
                FILE: Number of bytes written=214963
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=834
                HDFS: Number of bytes written=25
                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)=5303
                Total time spent by all reduces in occupied slots (ms)=5924
                Total time spent by all map tasks (ms)=5303
                Total time spent by all reduce tasks (ms)=5924
                Total vcore-milliseconds taken by all map tasks=5303
                Total vcore-milliseconds taken by all reduce tasks=5924
                Total megabyte-milliseconds taken by all map tasks=5430272
                Total megabyte-milliseconds taken by all reduce tasks=6066176
        Map-Reduce Framework
                Map input records=18
                Map output records=4
                Map output bytes=65
                Map output materialized bytes=39
                Input split bytes=101
                Combine input records=4
                Combine output records=2
                Reduce input groups=2
                Reduce shuffle bytes=39
                Reduce input records=2
                Reduce output records=2
                Spilled Records=4
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=158
                CPU time spent (ms)=1600
                Physical memory (bytes) snapshot=289832960
                Virtual memory (bytes) snapshot=4118241280
                Total committed heap usage (bytes)=170004480
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=733
        File Output Format Counters
                Bytes Written=25
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$ |
```

Results:

```
[acadgild@localhost Desktop]$ hadoop dfs -cat /Assign4Task3/part-r-00000
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

18/09/27 07:02:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Kerala  1
Uttar Pradesh   3
[acadgild@localhost Desktop]$ |
```