

Assignment-4

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

Mapreduce code written for the above task is as below:

```
package task1;

// imported all the built-in packages required for the task

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

public class InValid
{
    //Mapper class
    public static class Map extends Mapper <LongWritable ,/*Input
key Type */
Text,                               /*Input value Type*/
Text,                               /*Output key Type*/
Text>                               /*Output value Type*/
    {
        //Map function
        public void map(LongWritable key, Text value,
Context context) throws IOException, InterruptedException
        {

            // Given data is splitted into an array based on the
delimiter present in the data.

            String[] parts = value.toString().split("[|]");

            // If the company and product name not equals to "NA"
satisfies it will be sent to the output otherwise it won't.

            if(!parts[0].equals("NA") && !parts[1].equals("NA")) {

                context.write(value, null);
            }
        }
    }
}
```

```

//Main function
public static void main(String args[])throws Exception
{
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Invalid");

    job.setJarByClass(InValid.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setMapperClass(Map.class);
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

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

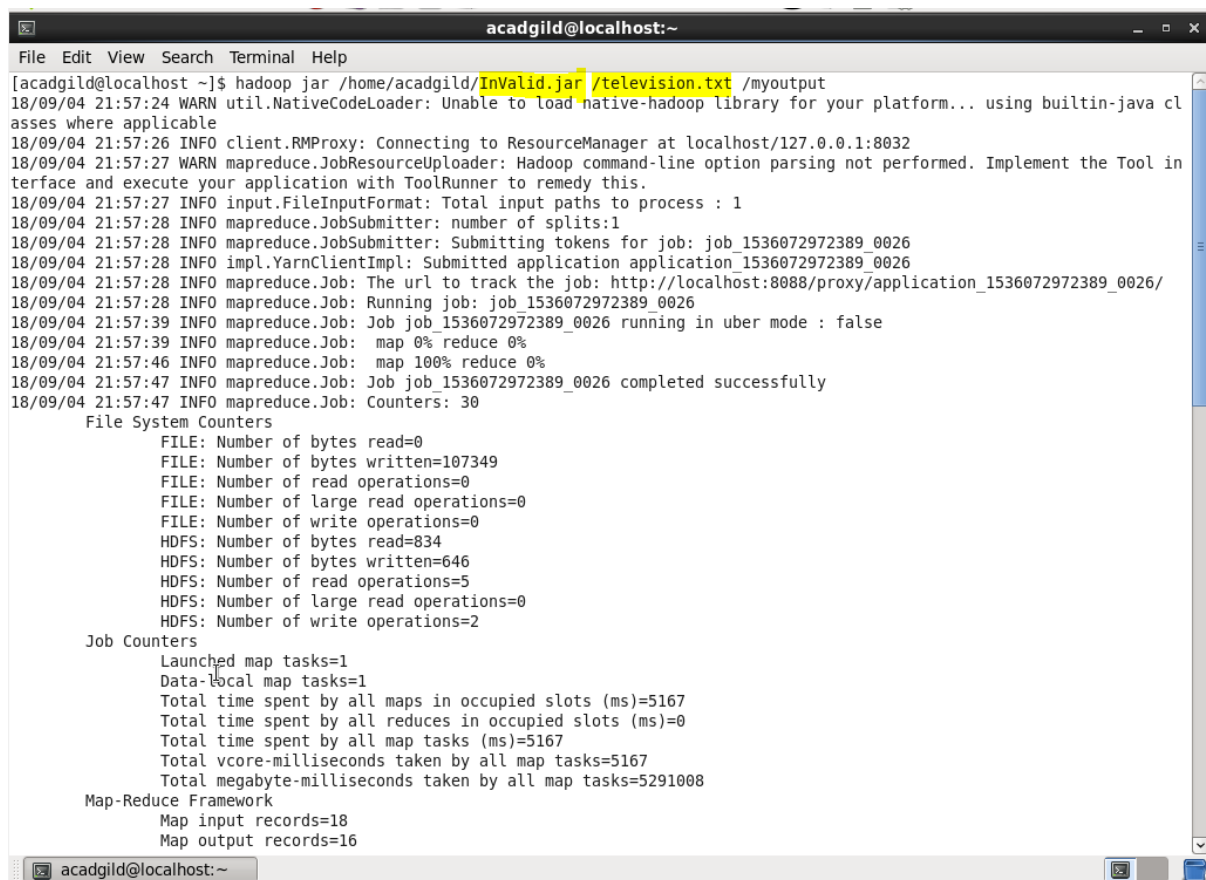
    //Reducer tasks are set to "0" as reducer is not required
for this task.
    job.setNumReduceTasks(0);

    Path outputPath = new Path(args[1]);
    outputPath.getFileSystem(conf).delete(outputPath, true);

    System.exit(job.waitForCompletion(true)? 0 :1);
}
}

```

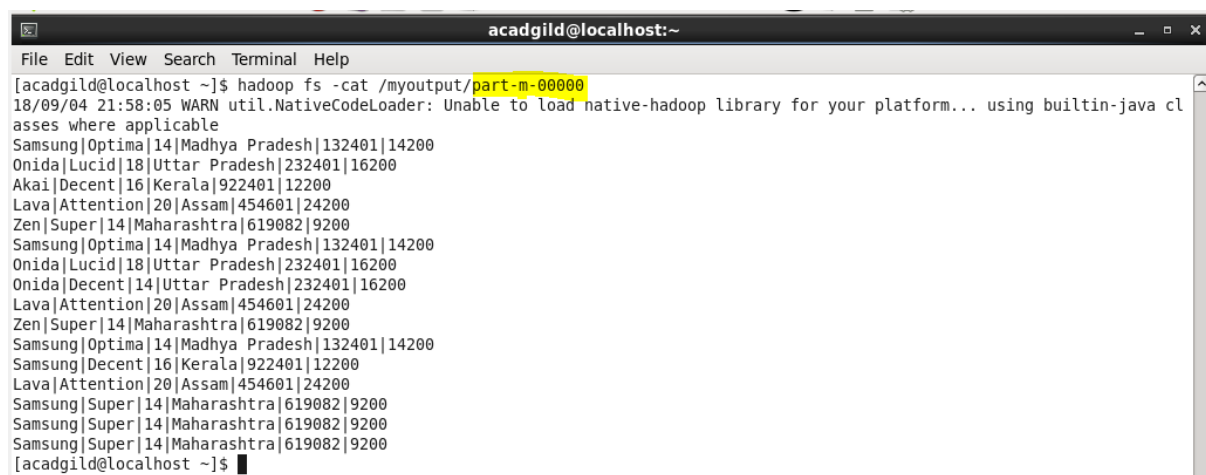
Executed the code **InValid.jar** with input file **television.txt**



```
acadgild@localhost:~  
File Edit View Search Terminal Help  
[acadgild@localhost ~]$ hadoop jar /home/acadgild/InValid.jar /television.txt /myoutput  
18/09/04 21:57:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl  
asses where applicable  
18/09/04 21:57:26 INFO client.RMPProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032  
18/09/04 21:57:27 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in  
terface and execute your application with ToolRunner to remedy this.  
18/09/04 21:57:27 INFO input.FileInputFormat: Total input paths to process : 1  
18/09/04 21:57:28 INFO mapreduce.JobSubmitter: number of splits:1  
18/09/04 21:57:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1536072972389_0026  
18/09/04 21:57:28 INFO impl.YarnClientImpl: Submitted application application_1536072972389_0026  
18/09/04 21:57:28 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1536072972389_0026/  
18/09/04 21:57:28 INFO mapreduce.Job: Running job: job_1536072972389_0026  
18/09/04 21:57:39 INFO mapreduce.Job: Job job_1536072972389_0026 running in uber mode : false  
18/09/04 21:57:39 INFO mapreduce.Job: map 0% reduce 0%  
18/09/04 21:57:46 INFO mapreduce.Job: map 100% reduce 0%  
18/09/04 21:57:47 INFO mapreduce.Job: Job job_1536072972389_0026 completed successfully  
18/09/04 21:57:47 INFO mapreduce.Job: Counters: 30  
File System Counters  
FILE: Number of bytes read=0  
FILE: Number of bytes written=107349  
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=646  
HDFS: Number of read operations=5  
HDFS: Number of large read operations=0  
HDFS: Number of write operations=2  
Job Counters  
Launched map tasks=1  
Data-local map tasks=1  
Total time spent by all maps in occupied slots (ms)=5167  
Total time spent by all reduces in occupied slots (ms)=0  
Total time spent by all map tasks (ms)=5167  
Total vcore-milliseconds taken by all map tasks=5167  
Total megabyte-milliseconds taken by all map tasks=5291008  
Map-Reduce Framework  
Map input records=18  
Map output records=16
```

Got the required output as shown in the below screenshot without “NA” records.

As there is no reduce code, the output file name will be with ‘m’ instead of ‘r’ as highlighted in the screenshot.



```
acadgild@localhost:~  
File Edit View Search Terminal Help  
[acadgild@localhost ~]$ hadoop fs -cat /myoutput/part-m-00000  
18/09/04 21:58:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl  
asses where applicable  
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  
Lava|Attention|20|Assam|454601|24200  
Zen|Super|14|Maharashtra|619082|9200  
Samsung|Optima|14|Madhya Pradesh|132401|14200  
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  
[acadgild@localhost ~]$
```

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

Mapreduce code written for the above task is as below:

```
package task1;

// imported all the built-in packages required for the task

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

public class Company
{
    //Mapper class
    public static class Map extends Mapper <LongWritable ,/*Input key
Type */
    Text,                      /*Input value Type*/
    Text,                      /*Output key Type*/
    IntWritable>              /*Output value Type*/
    {
        //Map function
        public void map(LongWritable key, Text value,
            Context context) throws IOException,
            InterruptedException
        {
            // Given data is splitted into an array based on the
            delimiter present in the data.
            String[] parts = value.toString().split("[|]");

            if(!parts[0].equals("NA")) {

                //If the company name is not equals "NA" satisfies then it
                will perform the following steps.

                value.set(parts[0]);
                //First position in the array is set "value".

                context.write(new Text(value), new IntWritable(1));

                //The value is passed as key to the output of Mapper and as
                the value is set with string, it is converted into Text. The value to
                the key in the output is passed as "1".

            }
        }
    }
}
```

```

//Reducer class (Received output from the mapper as Text and
intWritable)
    public static class Reduce extends Reducer< Text, IntWritable,
Text, IntWritable >
    {

        //Reduce function
        public void reduce( Text key, Iterable<IntWritable>
values,Context context) throws IOException, InterruptedException
        {
            int sum=0;

            //the below step will check for the each value.

            for(IntWritable x:values) {

                //each value will be added to the variable "sum"

                sum +=x.get();

            }
            context.write( key, new IntWritable(sum));

            //Now the output will be with key (company name) and value
(count of units)

        }
    }

//Main function
public static void main(String args[])throws Exception
{
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Company");

    job.setJarByClass(Company.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setMapperClass(Map.class);
    job.setCombinerClass(Reduce.class);
    job.setReducerClass(Reduce.class);
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

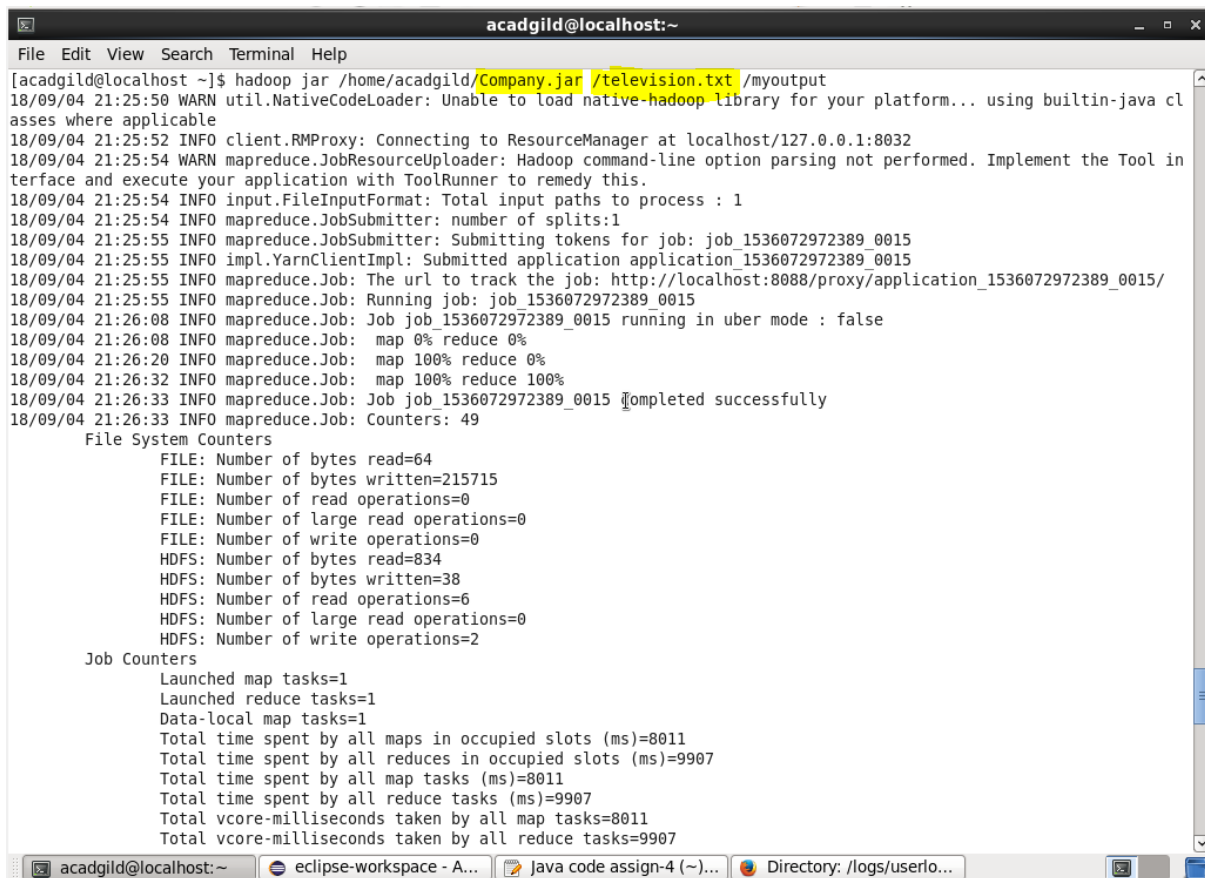
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    Path outputPath = new Path(args[1]);
    outputPath.getFileSystem(conf).delete(outputPath, true);

    System.exit(job.waitForCompletion(true)? 0 :1);
}
}

```

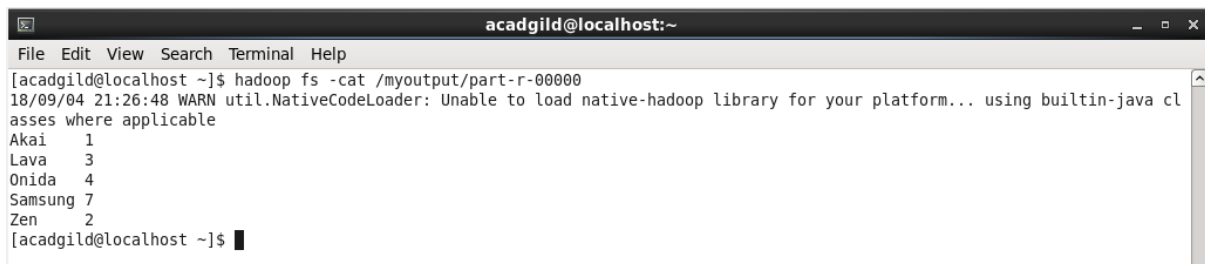
Executed the code **Company.jar** with input file **television.txt**



The screenshot shows a terminal window titled 'acadgild@localhost:~'. The command executed is 'hadoop jar /home/acadgild/Company.jar /television.txt /myoutput'. The output consists of various log messages from Hadoop, including warnings about native code loading, information about connecting to the ResourceManager, and details about the job submission and execution. The job is identified as 'job_1536072972389_0015'. The logs indicate that the job completed successfully. At the bottom of the terminal, there are tabs for 'eclipse-workspace - A...', 'Java code assign-4 (~)...', and 'Directory: /logs/userlo...'. The status bar at the very bottom shows 'acadgild@localhost:~'.

```
[acadgild@localhost ~]$ hadoop jar /home/acadgild/Company.jar /television.txt /myoutput
18/09/04 21:25:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/09/04 21:25:52 INFO client.RMPProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/09/04 21:25:54 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool in
terface and execute your application with ToolRunner to remedy this.
18/09/04 21:25:54 INFO input.FileInputFormat: Total input paths to process : 1
18/09/04 21:25:54 INFO mapreduce.JobSubmitter: number of splits:1
18/09/04 21:25:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1536072972389_0015
18/09/04 21:25:55 INFO impl.YarnClientImpl: Submitted application application_1536072972389_0015
18/09/04 21:25:55 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1536072972389_0015/
18/09/04 21:25:55 INFO mapreduce.Job: Running job: job_1536072972389_0015
18/09/04 21:26:08 INFO mapreduce.Job: Job job_1536072972389_0015 running in uber mode : false
18/09/04 21:26:08 INFO mapreduce.Job: map 0% reduce 0%
18/09/04 21:26:20 INFO mapreduce.Job: map 100% reduce 0%
18/09/04 21:26:32 INFO mapreduce.Job: map 100% reduce 100%
18/09/04 21:26:33 INFO mapreduce.Job: Job job_1536072972389_0015 completed successfully
18/09/04 21:26:33 INFO mapreduce.Job: Counters: 49
    File System Counters
      FILE: Number of bytes read=64
      FILE: Number of bytes written=215715
      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=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)=8011
      Total time spent by all reduces in occupied slots (ms)=9907
      Total time spent by all map tasks (ms)=8011
      Total time spent by all reduce tasks (ms)=9907
      Total vcore-milliseconds taken by all map tasks=8011
      Total vcore-milliseconds taken by all reduce tasks=9907
```

Got the required output as shown in the below screenshot.



The screenshot shows a terminal window titled 'acadgild@localhost:~'. The command executed is 'hadoop fs -cat /myoutput/part-r-00000'. The output is a list of names and their corresponding counts: Akai 1, Lava 3, Onida 4, Samsung 7, Zen 2. The terminal prompt is '[acadgild@localhost ~]\$'.

```
[acadgild@localhost ~]$ hadoop fs -cat /myoutput/part-r-00000
18/09/04 21:26:48 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Akai 1
Lava 3
Onida 4
Samsung 7
Zen 2
[acadgild@localhost ~]$
```

3). Write a Map Reduce program to calculate the total units sold in each state for Onida company.

Mapreduce code written for the above task is as below:

```
package task1;

// imported all the built-in packages required for the task

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

public class Onida
{
    //Mapper class
    public static class Map extends Mapper <LongWritable ,/*Input
key Type */
Text,
/*Input value Type*/
Text,
/*Output key Type*/
IntWritable>
/*Output value Type*/
{
        //Map function
        public void map(LongWritable key, Text value,
Context context) throws IOException, InterruptedException
        {

            //Given data is splitted into an array based on the
delimiter present in the data.

            String[] parts = value.toString().split("[|]");

            //If the company name equals to "Onida" satisfies it will
execute the statements in if condition.

            if(parts[0].equals("Onida")) {

                //The state name is set to the variable "value".

                value.set(parts[3]);

                context.write(new Text(value), new IntWritable(1));

                //The value is passed as key to the output of Mapper and as
the value is set with string it is converted into Text. The value to
the key in the output is passed as "1".
            }
        }
    }
}
```

```

    }
}

//Reducer class (Received output from the mapper as Text and
intWritable
    public static class Reduce extends Reducer< Text, IntWritable,
Text, IntWritable >
    {

        //Reduce function
        public void reduce( Text key, Iterable<IntWritable>
values,Context context) throws IOException, InterruptedException
        {
            int sum=0;

            //the below step will check for the each value.

            for(IntWritable x:values) {

                //each value will be added to the variable "sum"

                sum +=x.get();

            }
            context.write(key, new IntWritable(sum));

            //Now the output will be with key (company name) and value
(count of units)
        }
    }

//Main function
public static void main(String args[])throws Exception
{
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Onida");

    job.setJarByClass(Onida.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setMapperClass(Map.class);
    job.setCombinerClass(Reduce.class);
    job.setReducerClass(Reduce.class);
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

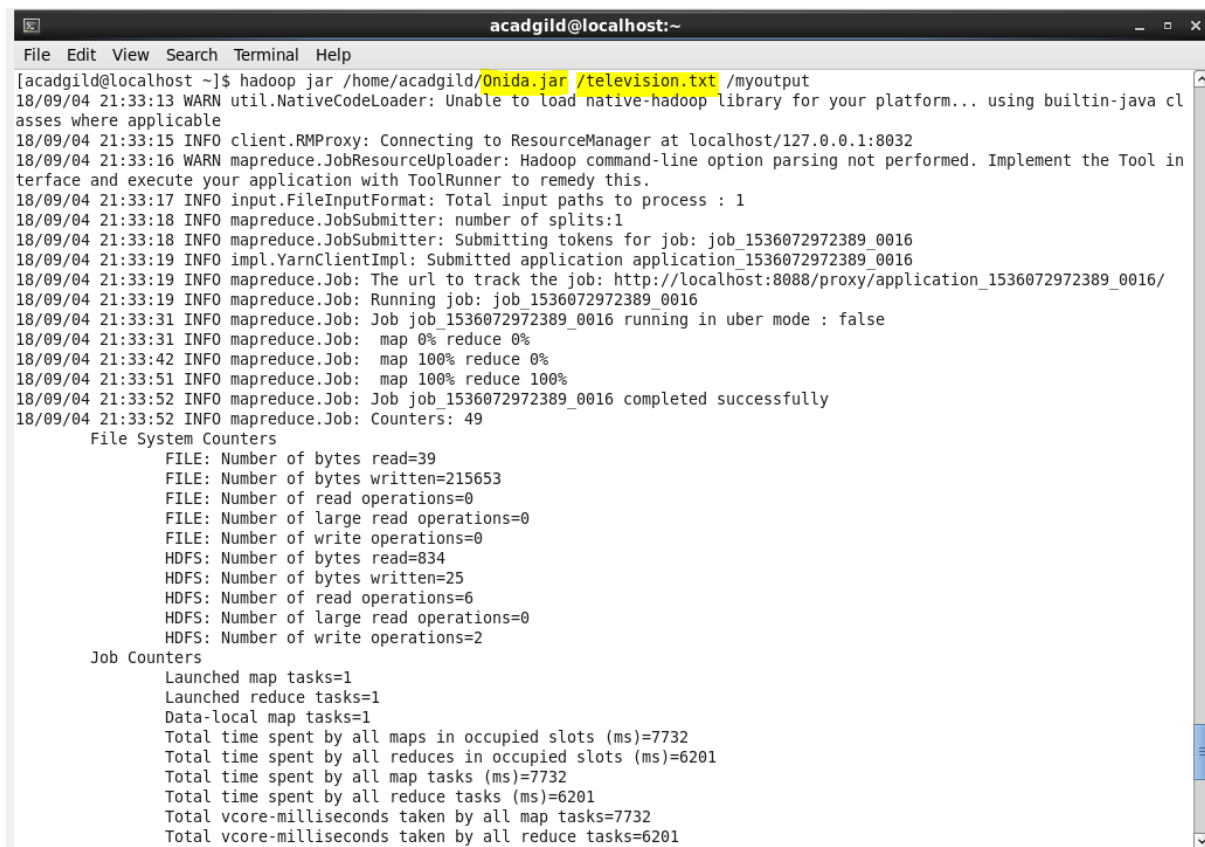
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    Path outputPath = new Path(args[1]);
    outputPath.getFileSystem(conf).delete(outputPath, true);

    System.exit(job.waitForCompletion(true)? 0 :1);
}
}

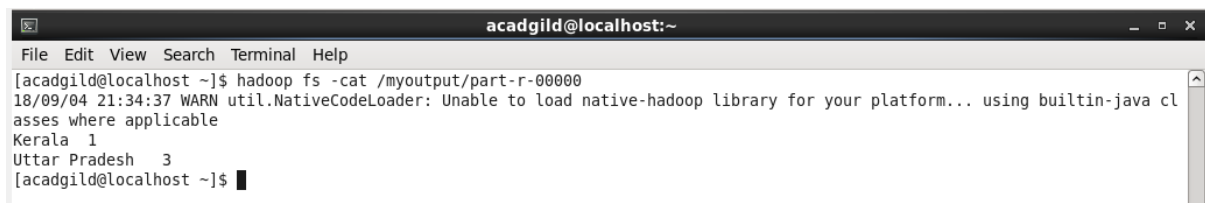
```


Executed the code **Onida.jar** with input file **television.txt**



```
acadgild@localhost:~  
File Edit View Search Terminal Help  
[acadgild@localhost ~]$ hadoop jar /home/acadgild/Onida.jar /television.txt /myoutput  
18/09/04 21:33:13 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
18/09/04 21:33:15 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032  
18/09/04 21:33:16 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.  
18/09/04 21:33:17 INFO input.FileInputFormat: Total input paths to process : 1  
18/09/04 21:33:18 INFO mapreduce.JobSubmitter: number of splits:1  
18/09/04 21:33:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1536072972389_0016  
18/09/04 21:33:19 INFO impl.YarnClientImpl: Submitted application application_1536072972389_0016  
18/09/04 21:33:19 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1536072972389_0016/  
18/09/04 21:33:19 INFO mapreduce.Job: Running job: job_1536072972389_0016  
18/09/04 21:33:31 INFO mapreduce.Job: Job job_1536072972389_0016 running in uber mode : false  
18/09/04 21:33:31 INFO mapreduce.Job: map 0% reduce 0%  
18/09/04 21:33:42 INFO mapreduce.Job: map 100% reduce 0%  
18/09/04 21:33:51 INFO mapreduce.Job: map 100% reduce 100%  
18/09/04 21:33:52 INFO mapreduce.Job: Job job_1536072972389_0016 completed successfully  
18/09/04 21:33:52 INFO mapreduce.Job: Counters: 49  
File System Counters  
FILE: Number of bytes read=39  
FILE: Number of bytes written=215653  
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)=7732  
Total time spent by all reduces in occupied slots (ms)=6201  
Total time spent by all map tasks (ms)=7732  
Total time spent by all reduce tasks (ms)=6201  
Total vcore-milliseconds taken by all map tasks=7732  
Total vcore-milliseconds taken by all reduce tasks=6201
```

Got the required output as shown in the below screenshot.



```
acadgild@localhost:~  
File Edit View Search Terminal Help  
[acadgild@localhost ~]$ hadoop fs -cat /myoutput/part-r-00000  
18/09/04 21:34:37 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 ~]$
```