Assignment_4.1
MapReduce_Introduction

Problem Statement :

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.

<u>Task 1:</u>

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

In input file 'television.txt', there are total 18 records. Out of these, there are 2 records with 'NA' value in either Company Name or Product Name.

Hence we are getting 16 records below after removing these two invalid records.

Here we have splitted content in input file into array of lines. Then these in turn splitted into array of words by using '|'. We have used if condition to remove invalid 'NA' records and for output key, we have declared output key as **NullWritable** because we are not displaying output key in the output. We are displaying only output value from Mapper.

Assignment_4.1
MapReduce_Introduction

Driver code :-

```
package task1;
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.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;

public class task1 {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Usage: task1 <input path> <output path>");
      System.exit(-1);
    }

       //Job Related Configurations
       Configuration conf = new Configuration();
       Job job = new Job(conf, "Task1 Job");
       job.setJarByClass(task1.class);


    // As This is map only job, we have specified the number of reducer to 0
    job.setNumReduceTasks(0);

    //Provide paths to pick the input file for the job
    FileInputFormat.setInputPaths(job, new Path(args[0]));


    //Provide paths to pick the output file for the job, and delete it if already
present
       Path outputPath = new Path(args[1]);
       FileOutputFormat.setOutputPath(job, outputPath);
       outputPath.getFileSystem(conf).delete(outputPath, true);

    //To set the mapper of this job and there is no Reducer
    job.setMapperClass(task1Mapper.class);

     //set the input and output format class
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    //We set output key as NullWritable as we are not returning key
    job.setOutputKeyClass(NullWritable.class);
    job.setOutputValueClass(Text.class);

    //execute the job
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Assignment_4.1
MapReduce_Introduction

Mapper code :-

```
package task1;
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.util.*;

public class task1Mapper
  extends Mapper<LongWritable, Text, NullWritable, Text> {


  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

   // Here we are converting Text to String
       String content = value.toString();

         String[] linesArray = content.split("  ");

      for(String line : linesArray){

        //we are splitting line by pipe (|)
         String[] word = line.split("\\|");

         //we are assigning company and product values from word
         Text company = new Text(word[0]);
         Text product = new Text(word[1]);

         // Remove lines which have company or product as "NA"
         if(!((company.equals(new Text("NA")))||(product.equals(new
Text("NA")))))
         {

              // Here we are converting String to Test
              Text lineText = new Text(line);

              //we are writing only value as lineText and NullWritable.get()
returns no key
                context.write(NullWritable.get(),lineText);
        }

      }
  }

}
```

Assignment_4.1
MapReduce_Introduction

Here we have exported ‘Assignment4_task1.jar’ as JAR file and ‘/televison.txt’ is input file path and ‘/task1output’ is Output directory path. By using below command, we are running JAR.

```
[acadgild@localhost ~]$ hadoop jar Assignment4_task1.jar  /television.txt /task1output
18/07/23 20:39:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
18/07/23 20:39:45 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/07/23 20:39:47 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
execute your application with ToolRunner to remedy this.
18/07/23 20:39:48 INFO input.FileInputFormat: Total input paths to process : 1
18/07/23 20:39:48 INFO mapreduce.JobSubmitter: number of splits:1
18/07/23 20:39:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1532349639129_0005
18/07/23 20:39:49 INFO impl.YarnClientImpl: Submitted application application_1532349639129_0005
18/07/23 20:39:49 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1532349639129_0005/
18/07/23 20:39:49 INFO mapreduce.Job: Running job: job_1532349639129_0005
18/07/23 20:40:04 INFO mapreduce.Job: Job job_1532349639129_0005 running in uber mode : false
18/07/23 20:40:04 INFO mapreduce.Job:  map 0% reduce 0%
18/07/23 20:40:14 INFO mapreduce.Job:  map 100% reduce 0%
18/07/23 20:40:14 INFO mapreduce.Job: Job job_1532349639129_0005 completed successfully
18/07/23 20:40:14 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=107350
                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)=7773
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=7773
                Total vcore-milliseconds taken by all map tasks=7773
                Total megabyte-milliseconds taken by all map tasks=7959552
        Map-Reduce Framework
                Map input records=18
                Map output records=16
                Input split bytes=101
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=92
                CPU time spent (ms)=830
                Physical memory (bytes) snapshot=95879168
                Virtual memory (bytes) snapshot=2056757248
                Total committed heap usage (bytes)=32571392
        File Input Format Counters
                Bytes Read=733
        File Output Format Counters
                Bytes Written=646
You have new mail in /var/spool/mail/acadgild
```

Then we have displayed list of files or directories under ‘/task1output’ output directory.

We could see content in file ‘part-m-00000’ using HDFS cat command.

Below output shows that there are no records with text “NA”. So we have filtered out invalid records.

```
[acadgild@localhost ~]$ hadoop fs -ls /task1output
18/07/23 20:41:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-07-23 20:40 /task1output/_SUCCESS
-rw-r--r--   1 acadgild supergroup        646 2018-07-23 20:40 /task1output/part-m-00000
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -cat /task1output/part-m-00000
18/07/23 20:42:12 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes 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
```

Assignment_4.1
MapReduce_Introduction

Task 2:

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

Driver code :-

```
package task2;

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.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;

public class task2 {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Usage: task2 <input path> <output path>");
      System.exit(-1);
    }

       //Job Related Configurations
       Configuration conf = new Configuration();
       Job job = new Job(conf, "Units count of companies");
       job.setJarByClass(task2.class);


    // Specify the number of reducer to 1
    job.setNumReduceTasks(1);

    //Provide paths to pick the input file for the job
    FileInputFormat.setInputPaths(job, new Path(args[0]));


    //Provide paths to pick the output file for the job, and delete it if already
present
       Path outputPath = new Path(args[1]);
       FileOutputFormat.setOutputPath(job, outputPath);
       outputPath.getFileSystem(conf).delete(outputPath, true);


    //To set the mapper and reducer of this job
    job.setMapperClass(task2Mapper.class);
    job.setReducerClass(task2Reducer.class);


     //set the input and output format class
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);
```

```
 //set up the output key and value classes
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    //execute the job
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Mapper code :-

```
package task2;
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.util.*;

public class task2Mapper
  extends Mapper<LongWritable, Text, Text, IntWritable> {

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


  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {

        // Here we are converting Text to String
       String content = value.toString();

        //we are splitting content into array of lines
        String[] linesArray = content.split("  ");

      for(String line : linesArray){

        //we are splitting line by pipe (|)
          String[] word = line.split("\\|");

          //we are assigning company and product values from word
          Text company = new Text(word[0]);
          Text product = new Text(word[1]);

          // Remove lines which have company or product as "NA"
          if(!((company.equals(new Text("NA")))||(product.equals(new
Text("NA")))))
          {

                context.write(company,one);
        }
            }
  }

}
```

Assignment_4.1
MapReduce_Introduction

Reducer code :-

```
package task2;

import java.io.IOException;

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


public class task2Reducer
  extends Reducer<Text, IntWritable, Text, IntWritable> {

  @Override
  public void reduce(Text key, Iterable<IntWritable> values,
      Context context)
      throws IOException, InterruptedException {
      System.out.println("From The Reducer=>"+key) ;

      int sum = 0;
      for (IntWritable value : values) {
             sum+=value.get();
       }
       context.write(key, new IntWritable(sum));
  }
}
```

Here we have exported ‘Assignment4_task2.jar’ as JAR file and ‘/televison.txt’ is input file path and ‘/task2output’ is Output directory path. By using below command, we are running JAR.

Assignment_4.1
MapReduce_Introduction

```
[acadgild@localhost ~]$ hadoop jar Assignment4_task2.jar  /television.txt /task2output
18/07/23 18:57:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
18/07/23 18:57:56 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/07/23 18:57:58 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
execute your application with ToolRunner to remedy this.
18/07/23 18:57:58 INFO input.FileInputFormat: Total input paths to process : 1
18/07/23 18:57:59 INFO mapreduce.JobSubmitter: number of splits:1
18/07/23 18:57:59 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1532349639129_0003
18/07/23 18:58:00 INFO impl.YarnClientImpl: Submitted application application_1532349639129_0003
18/07/23 18:58:00 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1532349639129_0003/
18/07/23 18:58:00 INFO mapreduce.Job: Running job: job_1532349639129_0003
18/07/23 18:58:15 INFO mapreduce.Job: Job job_1532349639129_0003 running in uber mode : false
18/07/23 18:58:15 INFO mapreduce.Job:  map 0% reduce 0%
18/07/23 18:58:26 INFO mapreduce.Job:  map 100% reduce 0%
18/07/23 18:58:38 INFO mapreduce.Job:  map 100% reduce 100%
18/07/23 18:58:39 INFO mapreduce.Job: Job job_1532349639129_0003 completed successfully
18/07/23 18:58:39 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=204
                FILE: Number of bytes written=215673
                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)=7636
                Total time spent by all reduces in occupied slots (ms)=9612
                Total time spent by all map tasks (ms)=7636
                Total time spent by all reduce tasks (ms)=9612
                Total megabyte-milliseconds taken by all map tasks=7819264
                Total megabyte-milliseconds taken by all reduce tasks=9842688
        Map-Reduce Framework
                Map input records=18
                Map output records=16
                Map output bytes=166
                Map output materialized bytes=204
                Input split bytes=101
                Combine input records=0
                Combine output records=0
                Reduce input groups=5
                Reduce shuffle bytes=204
                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)=272
                CPU time spent (ms)=2540
                Physical memory (bytes) snapshot=300527616
                Virtual memory (bytes) snapshot=4118192128
                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=38
```

Then we have displayed list of files or directories under '/task2output' output directory.

We could see content in file 'part-r-00000' using HDFS cat command.

Below output shows company names with total units sold by each Company.

Note here we have considered only valid 16 records, so we are getting below output

```
[acadgild@localhost ~]$ hadoop fs -ls /task2output
18/07/23 18:58:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-07-23 18:58 /task2output/_SUCCESS
-rw-r--r--   1 acadgild supergroup         38 2018-07-23 18:58 /task2output/part-r-00000
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -cat /task2output/part-r-00000
18/07/23 18:59:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
Akai    1
Lava    3
Onida   3
Samsung 7
Zen     2
```

Assignment_4.1
MapReduce_Introduction

Task 3:

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

Driver code :-

```
package task3;

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.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;



public class task3 {

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Usage: task3 <input path> <output path>");
      System.exit(-1);
    }

       //Job Related Configurations
       Configuration conf = new Configuration();
       Job job = new Job(conf, "Units count of companies");
       job.setJarByClass(task3.class);


    // Specify the number of reducer to 0
    job.setNumReduceTasks(1);

    //Provide paths to pick the input file for the job
    FileInputFormat.setInputPaths(job, new Path(args[0]));


    //Provide paths to pick the output file for the job, and delete it if already
present
       Path outputPath = new Path(args[1]);
       FileOutputFormat.setOutputPath(job, outputPath);
       outputPath.getFileSystem(conf).delete(outputPath, true);


    //To set the mapper and reducer of this job
    job.setMapperClass(task3Mapper.class);
```

```
    job.setReducerClass(task3Reducer.class);


     //set the input and output format class
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    //set up the output key and value classes
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    //execute the job
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Mapper code :-

```
package task3;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.util.*;

public class task3Mapper
  extends Mapper<LongWritable, Text, Text, IntWritable> {

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


  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {


       String content = value.toString();

         String[] linesArray = content.split("  ");

      for(String line : linesArray){


          String[] word = line.split("\\|");
          /*for(String s: lineArray){
              System.out.println(s);} -- for debuggin purpose */
          Text company = new Text(word[0]);
          Text product = new Text(word[1]);
          Text state = new Text(word[3]);
          // Remove lines which have company or product as "NA"
          if(!((company.equals(new Text("NA")))||(product.equals(new
Text("NA")))))
               if(company.equals(new Text("Onida")))
               {
```

```
                context.write(state,one);
                }

        }
    }

}
```

Reducer code :-

```
package task3;

import java.io.IOException;

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


public class task3Reducer
  extends Reducer<Text, IntWritable, Text, IntWritable> {

  @Override
  public void reduce(Text key, Iterable<IntWritable> values,
      Context context)
      throws IOException, InterruptedException {
      System.out.println("From The Reducer=>"+key) ;

      int sum = 0;
      for (IntWritable value : values) {
             sum+=value.get();
       }
       context.write(key, new IntWritable(sum));
  }
}
```

Here we have exported ‘Assignment4_task3.jar’ as JAR file and ‘/televison.txt’ is input file path and ‘/task3output’ is Output directory path. By using below command, we are running JAR.

Assignment_4.1
MapReduce_Introduction

```
acadgild@localhost ~]$ hadoop jar Assignment4_task3.jar  /television.txt /task3output
8/07/23 19:50:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
pplicable
8/07/23 19:50:23 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
8/07/23 19:50:25 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
xecute your application with ToolRunner to remedy this.
8/07/23 19:50:26 INFO input.FileInputFormat: Total input paths to process : 1
8/07/23 19:50:26 INFO mapreduce.JobSubmitter: number of splits:1
8/07/23 19:50:26 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1532349639129_0004
8/07/23 19:50:27 INFO impl.YarnClientImpl: Submitted application application_1532349639129_0004
8/07/23 19:50:27 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1532349639129_0004/
8/07/23 19:50:27 INFO mapreduce.Job: Running job: job_1532349639129_0004
8/07/23 19:50:41 INFO mapreduce.Job: Job job_1532349639129_0004 running in uber mode : false
8/07/23 19:50:41 INFO mapreduce.Job:  map 0% reduce 0%
8/07/23 19:50:53 INFO mapreduce.Job:  map 100% reduce 0%
8/07/23 19:51:05 INFO mapreduce.Job:  map 100% reduce 100%
8/07/23 19:51:06 INFO mapreduce.Job: Job job_1532349639129_0004 completed successfully
8/07/23 19:51:07 INFO mapreduce.Job: Counters: 49
       File System Counters
               FILE: Number of bytes read=66
               FILE: Number of bytes written=215397
               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=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)=9583
               Total time spent by all reduces in occupied slots (ms)=9314
               Total time spent by all map tasks (ms)=9583
               Total time spent by all reduce tasks (ms)=9314
               Total vcore-milliseconds taken by all map tasks=9583
               Total megabyte-milliseconds taken by all map tasks=9812992
               Total megabyte-milliseconds taken by all reduce tasks=9537536
       Map-Reduce Framework
               Map input records=18
               Map output records=3
               Map output bytes=54
               Map output materialized bytes=66
               Input split bytes=101
               Combine input records=0
               Combine output records=0
               Reduce input groups=1
               Reduce shuffle bytes=66
               Reduce input records=3
               Reduce output records=1
               Spilled Records=6
               Shuffled Maps =1
               Failed Shuffles=0
               Merged Map outputs=1
               GC time elapsed (ms)=334
               CPU time spent (ms)=2890
               Physical memory (bytes) snapshot=299065344
               Virtual memory (bytes) snapshot=4118192128
               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=16
ou have new mail in /var/spool/mail/acadgild
```

Then we have displayed list of files or directories under '/task3output' output directory.

We could see content in file 'part-r-00000' using HDFS cat command.

Below output shows total units sold in each state for Onida company.

Note,here we have considered only valid 16 records, so we are getting output as :

'Uttar Pradesh 3'.

```
[acadgild@localhost ~]$ hadoop fs -ls /task3output
18/07/23 19:51:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-07-23 19:51 /task3output/_SUCCESS
-rw-r--r--   1 acadgild supergroup         16 2018-07-23 19:51 /task3output/part-r-00000
[acadgild@localhost ~]$ hadoop fs -cat /task3output/part-r-00000
18/07/23 19:51:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
applicable
Uttar Pradesh   3
```