**Problem:**

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**

Copying the data file in Hadoop file system

[cloudera@quickstart ~]$ hadoop fs -put Desktop/television.txt /Ajay

**Data Files**

[cloudera@quickstart ~]$ hadoop fs -cat /Ajay/television.txt

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[cloudera@quickstart ~]$

**Mapper Program**

**package** com.test.television;

**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.mapreduce.\*;

**public** **class** Task1Mapper **extends** Mapper<LongWritable, Text, Text, IntWritable> {

**public** **void** map(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

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

String line = value.toString();

String[] lineArray = line.split("\\W+");

String compName= lineArray[0];

//IntWritable year = new IntWritable(Integer.parseInt(lineArray[0].split("-")[2]));

IntWritable temp = **new** IntWritable(1);

Text word = **new** Text();

word.set(compName);

context.write(word, temp);

}

}

**Reducer Program**

package com.test.television;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.io.Text;

public class Task1Reducer extends Reducer<Text, IntWritable, Text, IntWritable>

{

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

{

int sum = 0;

for (IntWritable value : values) {

//if (value.get() == 1) {

sum += value.get();

// }

}

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

}

}

**Driver Program**

**package** com.test.television;

**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.mapreduce.lib.input.KeyValueTextInputFormat;

//import org.apache.hadoop.mapreduce.lib.output.keyValueTextOutputFormat;

**import** org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.Text;

**public** **class** Task1 {

@SuppressWarnings("deprecation")

**public** **static** **void** main(String[] args) **throws** Exception {

Configuration conf = **new** Configuration();

Job job = **new** ~~Job~~(conf, "DemoTask1");

job.setJarByClass(Task1.**class**);

job.setMapOutputKeyClass(Text.**class**);

job.setMapOutputValueClass(IntWritable.**class**);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(IntWritable.**class**);

job.setMapperClass(Task1Mapper.**class**);

job.setReducerClass(Task1Reducer.**class**);

//job.setInputFormatClass(KeyValueTextInputFormat.class);

//job.setOutputFormatClass(KeyValueTextOutputFormat.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);

\*/

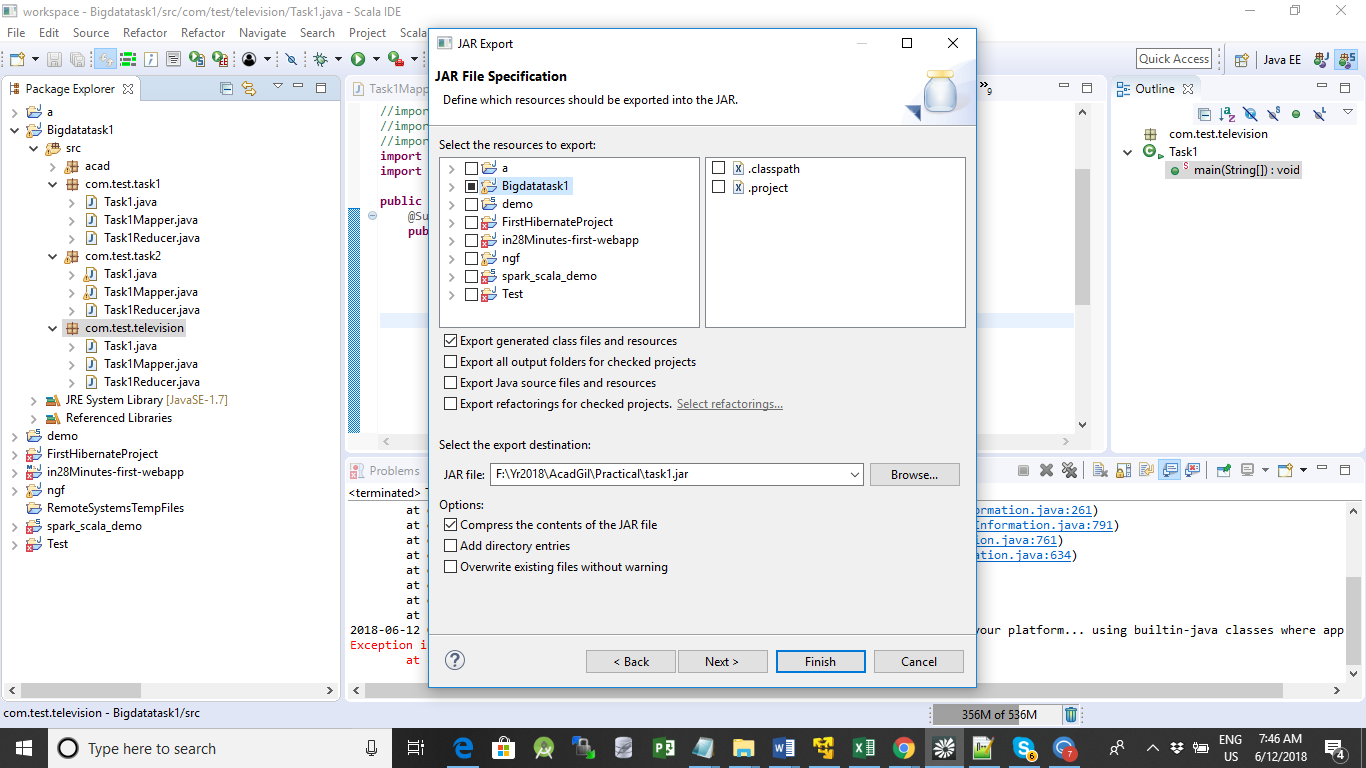
job.waitForCompletion(**true**);

}

}

**To run map reduce in Hadoop**

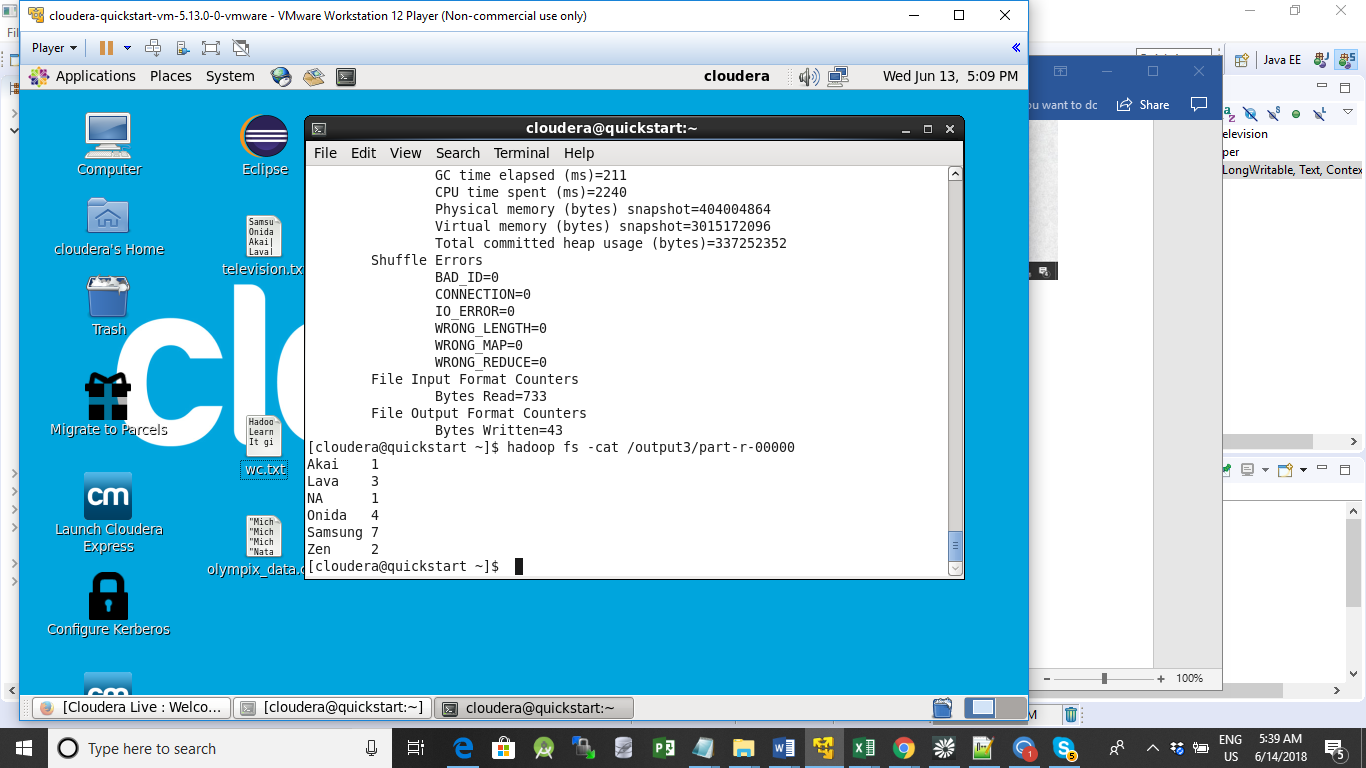
**First Copy the JAR file in Cloudera environment**



[cloudera@quickstart ~]$ mv Desktop/task1.jar .

mv: overwrite `./task1.jar'? y

**[cloudera@quickstart ~]$ hadoop jar task1.jar com.test.television.Task1 /Ajay/television.txt /output3**



**View the output file**

[cloudera@quickstart ~]$ hadoop fs -cat /output3/part-r-00000

[cloudera@quickstart ~]$ hadoop fs -cat /output3/part-r-00000

Akai 1

Lava 3

NA 1

Onida 4

Samsung 7

Zen 2

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

company.

For this changed the mapper program as below :

**Mapper Program**

**package** com.test.television;

**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.mapreduce.\*;

**public** **class** Task1Mapper **extends** Mapper<LongWritable, Text, Text, IntWritable> {

**public** **void** map(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

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

String line = value.toString();

String[] lineArray = line.split("\\W+");

String compName= lineArray[0];

String stateName= lineArray[3];

//IntWritable year = new IntWritable(Integer.parseInt(lineArray[0].split("-")[2]));

**if**(compName.equals("Onida"))

{

IntWritable temp = **new** IntWritable(1);

Text word = **new** Text();

word.set(stateName);

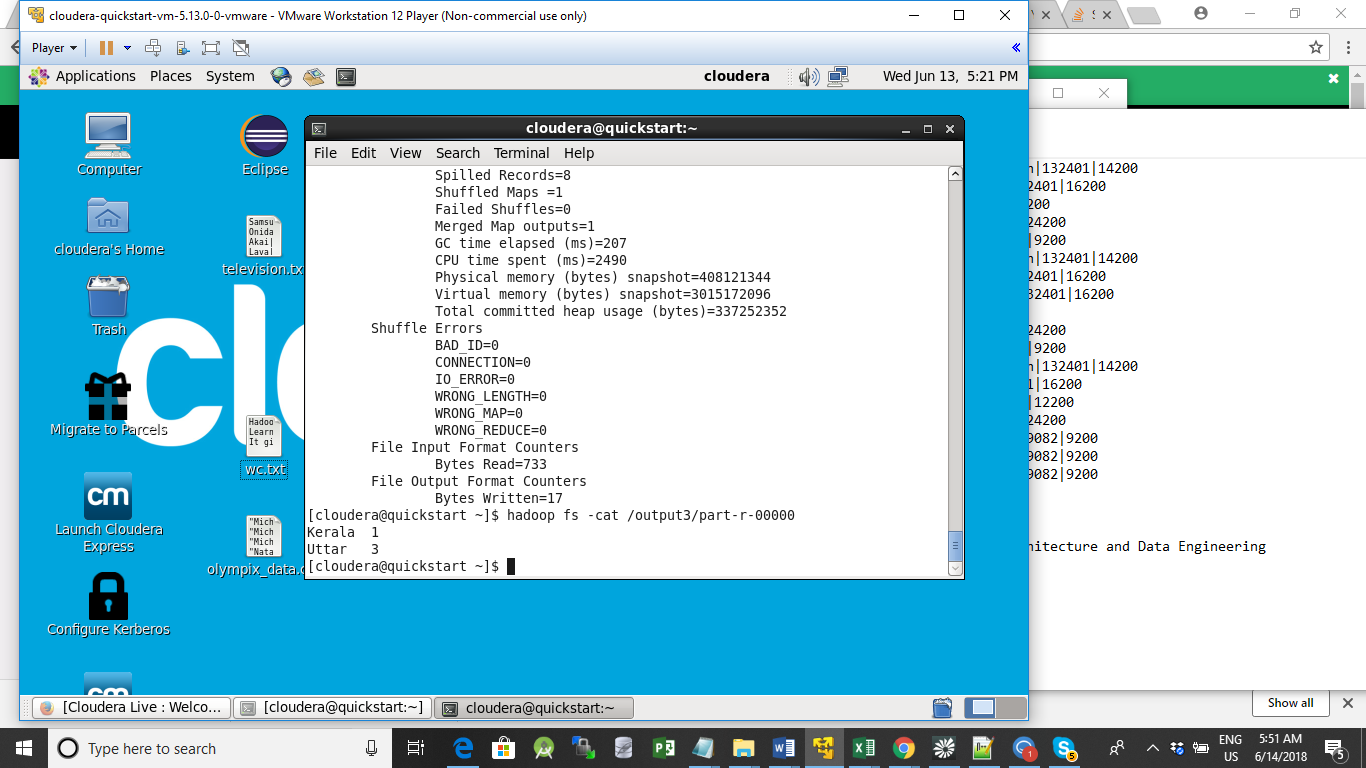
context.write(word, temp);

}

}

}

Output is :



**Output :**

Units sold for Onida product, state wide is as below:

[cloudera@quickstart ~]$ hadoop fs -cat /output3/part-r-00000

Kerala 1

Uttar 3