**Assignment 3.3**

1. Program for calculating the total units sold in each state for Onida.

**Job Program**:

**package** Onida;

**import** javax.security.auth.login.Configuration;

**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.IntWritable;

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

**public** **class** OnidaStateMain{

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

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

Job job = **new** ~~Job~~(conf, "Output Assignment3.1");

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

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

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

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

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

Job.setMapperClass(OnidaStateMapper.**class**);

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

Job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass (TextOutputFormat.**class**);

job.setNumReduceTasks(1);

job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass(TextOutputFormat.**class**);

FileInputFormat.*addInputPath*(job, **new** Path (args[0]));

FileOutputFormat.*setOutputPath*(job, **new** Path (args[1]));

job.waitForCompletion(**true**);

}

}

}

**Reducer Program**:

**package** Onida;

**import** java.io.IOException;

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

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

**import** org.apache.hadoop.mapreduce.\*;

**public** **class** OnidaStateReducer **extends** Reducer<Text, IntWritable, Text, IntWritable> {

**public** **void** reduce (Text statename, Iterable<IntWritable> values, Context context)

**throws** IOException, InterruptedException {

**int** sum= 0;

**for** (IntWritable value : values) {

sum+= value.get();

}

context.write(statename, **new** IntWritable(sum));

}

}

**Mapper Program:**

**package** Onida;

**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** OnidaStateMapper **extends** Mapper<LongWritable, Text, Text, IntWritable> {

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

**throws** IOException, InterruptedException {

String[] LineArray = value.toString().split("\\");

**if**(LineArray[0].equals("Onida")) {

Text statename = **new** Text(LineArray[3]);

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

context.write(statename, unit);

}

}

}

1. Program to get Total Units:

**Job Program:**

**package** Total;

**import** javax.security.auth.login.Configuration;

**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.IntWritable;

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

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.conf.\*;

**public** **class** TotalUnit{

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

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

Job job = **new** ~~Job~~(conf,"Total Assignment3.1");

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

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

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

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

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

Job.setMapperClass(TotalUnitMapper.**class**);

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

job.setNumReduceTasks(2);

job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass(TextOutputFormat.**class**);

FileInputFormat.*addInputPath*(job, **new** Path (args[0]));

FileOutputFormat.*setOutputPath*(job, **new** Path (args[1]));

job.waitForCompletion(**true**);

}

}

**Mapper Program:**

**package** Total;

**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** TotalUnitMapper **extends** Mapper<LongWritable, Text, Text, IntWritable> {

Text tvname;

//IntWritable unit;

**public** **void** setup(Context context) {

tvname = **new** Text();

}

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

**throws** IOException, InterruptedException {

String[] lineArray2 = value.toString().split("\\|");

**if**(!(lineArray2[0].equals("NA") || (lineArray2[1].equals("NA")))) {

tvname.set((lineArray2[0]));

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

}

}

}

**Reducer Program:**

package Total;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.\*;

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

public void reduce (Text tvname, Iterable<IntWritable> values, Context context)

throws IOException, InterruptedException {

int sum= 0;

for (IntWritable value : values) {

sum+= value.get(); }

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

}

}