**ASSIGNMENT- 3.3**

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.

**Task\_sum\_mapper.java**

package assignment\_3;

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 Task\_sum\_mapper extends Mapper<LongWritable, Text, Text, IntWritable> {

Text Company;

IntWritable units;

@Override

public void setup(Context context){

Company = new Text();

units = new IntWritable();

}

@Override

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

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

if((lineArray.length>0)

&& (lineArray[0] !=null)

&& (lineArray[1] !=null)

&& (!lineArray[0].equalsIgnoreCase("NA"))

&& (!lineArray[1].equalsIgnoreCase("NA"))

){

Company.set(lineArray[0]);

units.set(Integer.parseInt(lineArray[2]));

context.write(new Text(Company), units);

}

}

}

**Task\_sum\_reducer.java**

package assignment\_3;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class Task\_sum\_reducer extends Reducer<Text, IntWritable, Text, IntWritable>{

Integer sum;

public void setup (Context context){

sum = new Integer(0);

}

//To get key wise sum of units

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

Integer sum = 0;

for(IntWritable value : values){

sum = sum + value.get();

}

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

}

}

**Task\_sum.java**

package assignment\_3;

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.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;

public class Task\_sum {

@SuppressWarnings("deprecation")

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

Configuration config = new Configuration();

Job job = new Job(config,"Demo\_job");

job.setJarByClass(Task\_sum.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setMapperClass(Task\_sum\_mapper.class);

job.setReducerClass(Task\_sum\_reducer.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);

}

}

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

**Task\_onida\_mapper.java**

|  |
| --- |
| package assignment\_3\_1; |
|  |  |
|  | 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.Mapper; |
|  |  |
|  | public class Task\_onida\_mapper extends Mapper<LongWritable, Text, Text, IntWritable>{ |
|  |  |
|  | Text State; |
|  | IntWritable units; |
|  |  |
|  | @Override |
|  | public void setup(Context context){ |
|  | State = new Text(); |
|  | units = new IntWritable(); |
|  | } |
|  |  |
|  | @Override |
|  | public void map (LongWritable key, Text value, Context context) throws IOException, InterruptedException { |
|  |  |
|  | String[] lineArray = value.toString().split("\\|"); |
|  |  |
|  | if((lineArray.length>0) |
|  | && (lineArray[0].matches("Onida")) |
|  | ){ |
|  |  |
|  | State.set(lineArray[0]+"|"+lineArray[3]); |
|  | units.set(Integer.parseInt(lineArray[2])); |
|  |  |
|  | context.write(new Text(State), units); |
|  |  |
|  | } |
|  | } |
|  | } |

**Task\_onida\_reducer.java**

package assignment\_3\_1;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class Task\_onida\_reducer extends Reducer<Text, IntWritable, Text, IntWritable> {

Integer sum;

public void setup (Context context){

sum = new Integer(0);

}

//To get key wise sum of units

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

Integer sum = 0;

for(IntWritable value : values){

sum = sum + value.get();

}

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

}

}

**Task\_onida.java**

package assignment\_3\_1;

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.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;

public class Task\_onida {

@SuppressWarnings("deprecation")

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

Configuration config = new Configuration();

Job job = new Job(config,"Demo\_job");

job.setJarByClass(Task\_onida.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setMapperClass(Task\_onida\_mapper.class);

job.setReducerClass(Task\_onida\_reducer.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);

}

}