TotalUnitsSoldByEachCompanyMapper.java:

import java.io.IOException;

import java.util.regex.Pattern;

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 TotalUnitsSoldByEachCompanyMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

public void map(LongWritable Key, Text Value, Context context) throws IOException, InterruptedException {

String[] lineArray = Value.toString().split(Pattern.quote("|"));

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

Text company\_Name = new Text(lineArray[0]);

IntWritable count = new IntWritable(1);

context.write(company\_Name, count);

}

}

}

TotalUnitsSoldByEachCompanyPartitioner.java:

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Partitioner;

public class TotalUnitsSoldByEachCompanyPartitioner extends Partitioner<Text, IntWritable> {

private static final String AF = "ABCDEF";

private static final String GL = "GHIJKL";

private static final String MR = "MNOPQR";

@Override

public int getPartition(Text key, IntWritable value, int arg2) {

String k = key.toString().toUpperCase().substring(0, 1);

if ( AF.contains(k) )

{

return 0;

}

else if ( GL.contains(k))

{

return 1;

}

else if ( MR.contains(k))

{

return 2;

}

else

{

return 3;

}

}

}

TotalUnitsSoldByEachCompanyReducer.java:

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

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

{

private IntWritable total = new IntWritable();

private Integer minValue = Integer.MIN\_VALUE;

@Override

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

{

Integer count = 0;

for ( IntWritable value : values )

{ if(value.get()>minValue){

count+=value.get();

}

}

total.set(count);

context.write(key, total);

}

}

TotalUnitsSoldByEachCompanyWithCombiner.java:

import java.io.IOException;

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 TotalUnitsSoldByEachCompanyWithCombiner {

public static void main(String[] args) throws IllegalArgumentException, IOException, ClassNotFoundException, InterruptedException {

// TODO Auto-generated method stub

Configuration conf = new Configuration();

Job job = new Job(conf, "TotalUnitsSoldByEachCompanyWithCombiner");

job.setJarByClass(TotalUnitsSoldByEachCompanyWithCombiner.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setMapperClass(TotalUnitsSoldByEachCompanyMapper.class);

//only added a combiner class which is same class as reducer

job.setCombinerClass(TotalUnitsSoldByEachCompanyReducer.class);

job.setReducerClass(TotalUnitsSoldByEachCompanyReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

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);

}

}

TotalUnitsSoldByEachCompanyWithReducers.java:

import java.io.IOException;

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 TotalUnitsSoldByEachCompanyWithReducers {

public static void main(String[] args) throws IllegalArgumentException, IOException, ClassNotFoundException, InterruptedException {

// TODO Auto-generated method stub

Configuration conf = new Configuration();

Job job = new Job(conf, "TotalUnitsSoldByEachCompanyWithReducers");

job.setJarByClass(TotalUnitsSoldByEachCompanyWithReducers.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setMapperClass(TotalUnitsSoldByEachCompanyMapper.class);

job.setPartitionerClass(TotalUnitsSoldByEachCompanyPartitioner.class);

job.setNumReduceTasks(4);

job.setReducerClass(TotalUnitsSoldByEachCompanyReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

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);

}

}