StateRegionWiseTotalSalesPrice.java

import java.io.\*;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.MultipleInputs;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class StateRegionWiseTotalSalesPrice{

public static class StateRegionWiseTotalSalesPriceMapper extends Mapper<LongWritable, Text, Text,Text> {

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

String line = value.toString();

String[] record = line.split(",");

con.write(new Text(record[9]),new Text("sum\t" + record[15]));}

}

public static class StateRegionWiseTotalSalesPriceReducer extends Reducer <Text,Text ,Text,Text> {

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

double Totalprice=0.0;

for(Text t:values) {

String[] parts =t.toString().split("\t");

Totalprice +=Float.parseFloat(parts[1]);}

String str =String.format("%f",Totalprice);

con.write(key, new Text(str));} }

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

boolean recursive=true;

Configuration conf=new Configuration();

FileSystem fs=FileSystem.get(conf);

if(fs.exists(new Path(args[1])))

fs.delete(new Path(args[1]),recursive);

Job job=Job.getInstance(conf, " State Region wise Total Sales price ");

job.setJarByClass(StateRegionWiseTotalSalesPrice.class);

job.setMapperClass(StateRegionWiseTotalSalesPriceMapper.class);

job.setReducerClass(StateRegionWiseTotalSalesPriceReducer.class);

job.setNumReduceTasks(1);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

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

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

job.waitForCompletion(true);} }

COMMANDS

start-all.sh  
jps

export HADOOP\_CLASSPATH=$(hadoop classpath)

echo $HADOOP\_CLASSPATH

hadoop fs -mkdir hdfs://localhost:9000/datapurchase

hadoop fs -mkdir hdfs://localhost:9000/datapurchase/input

hadoop fs -put '/home/hadoop/datapurchase/input\_data/superstore.csv' /datapurchase/input

ls

javac -classpath ${HADOOP\_CLASSPATH} -d '/home/hadoop/datapurchase/classfiles' '/home/hadoop/datapurchase/StateRegionWiseTotalSalesPrice.java'

cd datapurchase

jar -cvf firstTutorial.jar -C classfiles/ .

hadoop jar '/home/hadoop/datapurchase/firstTutorial.jar' StateRegionWiseTotalSalesPrice /datapurchase/input /datapurchase/output

hadoop dfs -cat /datapurchase/output/\*







