Project ADA

Airline Delay Analysis

INFO 7250 Project

# Analysis with MapReduce

The following is the Analysis with MapReduce implemented so far.

MapReduce to find the average departure time of each carrier(airline company) of one specific year.

# Source Codes

## Main.java

import Mappers.CarrierDepaDelayMapper;  
import Reducers.CarrierDepaDelayReducer;  
import org.apache.hadoop.conf.Configuration;  
import org.apache.hadoop.conf.Configured;  
import org.apache.hadoop.fs.Path;  
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.util.Tool;  
import org.apache.hadoop.util.ToolRunner;  
  
public class ADAMain extends Configured implements Tool {  
 @Override  
 public int run(String[] strings) throws Exception {  
 Configuration conf = new Configuration();  
 Job job = new Job(conf,"play with secondary sort");  
  
 job.setJarByClass(ADAMain.class);  
  
*// job.setGroupingComparatorClass(.class);  
// job.setSortComparatorClass(.class);  
// job.setPartitionerClass(.class);* job.setInputFormatClass(TextInputFormat.class);  
 job.setOutputKeyClass(Text.class);  
 job.setOutputValueClass(Text.class);  
  
 job.setMapperClass(CarrierDepaDelayMapper.class);  
*// job.setCombinerClass(.class);* job.setReducerClass(CarrierDepaDelayReducer.class);  
  
 FileInputFormat.*addInputPath*(job,new Path(strings[0]));  
 FileOutputFormat.*setOutputPath*(job,new Path(strings[1]));  
  
  
 boolean b = job.waitForCompletion(false);  
  
 return b?0:1;  
 }  
  
 public static void main(String[] args) throws Exception {  
 int exit = ToolRunner.*run*(new ADAMain(),args);  
 System.*exit*(exit);  
  
 }  
}

## CarrierDepaDelayMapper.java

package Mappers;  
  
import org.apache.hadoop.io.DoubleWritable;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.Mapper;  
  
import java.io.IOException;  
  
public class CarrierDepaDelayMapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {  
 private Text carrierText = new Text();  
 private DoubleWritable depaDelayDouble = new DoubleWritable();  
  
 @Override  
 protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, DoubleWritable>.Context context) throws IOException, InterruptedException {  
 String[] flight\_data = value.toString().split(",");  
 String carrier = flight\_data[1];  
 double depaDelay = Double.*parseDouble*(flight\_data[7]);  
  
 carrierText.set(carrier);  
 depaDelayDouble.set(depaDelay);  
  
 context.write(carrierText, depaDelayDouble);  
 }  
}

## CarrierDepaDelayReducer.java

package Reducers;  
  
import org.apache.hadoop.io.DoubleWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.Reducer;  
  
import java.io.IOException;  
  
public class CarrierDepaDelayReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {  
 private DoubleWritable avgDepaDeplayDouble = new DoubleWritable();  
 @Override  
 protected void reduce(Text key, Iterable<DoubleWritable> values, Reducer<Text, DoubleWritable, Text, DoubleWritable>.Context context) throws IOException, InterruptedException {  
 double avgDepaDelay = 0.0;  
 int count = 0;  
 for (DoubleWritable doubleWritable : values) {  
 avgDepaDelay += doubleWritable.get();  
 count++;  
 }  
 avgDepaDelay = avgDepaDelay / count;  
 avgDepaDeplayDouble.set(avgDepaDelay);  
 context.write(key,avgDepaDeplayDouble);  
 }  
}