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

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

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

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

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

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

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

public class WeatherAnalyzer extends Configured implements Tool {

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

int exitCode = ToolRunner.run(new WeatherAnalyzer(), args);

System.exit(exitCode);

}

@Override

public int run(String[] args) throws Exception {

if (args.length != 2) {

System.err.println("Usage: WeatherAnalyzer <input\_path> <output\_path>");

return 1;

}

// Create a new job

Job job = Job.getInstance(getConf(), "Weather Analysis");

job.setJarByClass(WeatherAnalyzer.class);

// Set the mapper and reducer classes

job.setMapperClass(WeatherMapper.class);

job.setReducerClass(WeatherReducer.class);

// Set the output key-value classes

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(DoubleWritable.class);

// Set the input and output paths

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

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

// Wait for the job to complete and return the status

return job.waitForCompletion(true) ? 0 : 1;

}

}

import java.io.IOException;

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

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

public class WeatherReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {

private DoubleWritable outputValue = new DoubleWritable();

@Override

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

double sum = 0;

int count = 0;

// Calculate the sum and count for the given field

for (DoubleWritable value : values) {

sum += value.get();

count++;

}

// Calculate the average and emit the result

double average = sum / count;

outputValue.set(average);

context.write(key, outputValue);

}

}

import java.io.IOException;

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

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

public class WeatherMapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {

private Text outputKey = new Text();

private DoubleWritable outputValue = new DoubleWritable();

@Override

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

// Split the CSV line into individual fields

String[] fields = value.toString().split(",");

if (fields.length >= 5) {

// Extract the relevant fields (temperature, dew point, wind speed)

double temperature = Double.parseDouble(fields[2]);

double dewPoint = Double.parseDouble(fields[3]);

double windSpeed = Double.parseDouble(fields[4]);

// Emit key-value pairs (field name, field value)

outputKey.set("Temperature");

outputValue.set(temperature);

context.write(outputKey, outputValue);

outputKey.set("DewPoint");

outputValue.set(dewPoint);

context.write(outputKey, outputValue);

outputKey.set("WindSpeed");

outputValue.set(windSpeed);

context.write(outputKey, outputValue);

}

}

}