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.output.FileOutputFormat;

import org.apache.hadoop.io.compress.GzipCodec;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.io.compress.CompressionCodec;

// vv MaxTemperatureWithCombiner

public class MaxTemperatureWithCombiner {

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

if (args.length != 2) {

System.err.println("Usage: MaxTemperatureWithCombiner <input path> " +

"<output path>");

System.exit(-1);

}

Configuration config = new Configuration();

config.setBoolean(Job.MAP\_OUTPUT\_COMPRESS, true);

config.setClass(Job.MAP\_OUTPUT\_COMPRESS\_CODEC, GzipCodec.class,CompressionCodec.class);

Job job = new Job(config);

job.setJarByClass(MaxTemperatureWithCombiner.class);

job.setJobName("Max and Min temperature Combiner for ITEM3");

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

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

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileOutputFormat.setCompressOutput(job, true);

FileOutputFormat.setOutputCompressorClass(job, GzipCodec.class);

job.setMapperClass(MaxTemperatureMapper.class);

/\*[\*/job.setCombinerClass(MaxTemperatureReducer.class)/\*]\*/;

job.setReducerClass(MaxTemperatureReducer.class);

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}