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.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

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

public class SimplestMatrixMultiplication {

public static class MatrixMapper extends Mapper<Object, Text, Text, Text> {

private Text outputKey = new Text();

private Text outputValue = new Text();

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

// Assuming input format: MatrixID,row,col,value

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

String matrixID = tokens[0];

String rowCol = tokens[1] + "," + tokens[2];

String cellValue = tokens[3];

// Emit intermediate key: row or col, value: matrixID,cellValue

outputKey.set(rowCol);

outputValue.set(matrixID + "," + cellValue);

context.write(outputKey, outputValue);

}

}

public static class MatrixReducer extends Reducer<Text, Text, Text, IntWritable> {

private IntWritable outputValue = new IntWritable();

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

int[] aRow = new int[2];

int[] bCol = new int[2];

for (Text value : values) {

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

int matrixID = tokens[0].equals("A") ? 0 : 1;

int cellValue = Integer.parseInt(tokens[1]);

if (matrixID == 0) {

aRow[key.charAt(0) - '0'] = cellValue;

} else {

bCol[key.charAt(2) - '0'] = cellValue;

}

}

int result = aRow[0] \* bCol[0] + aRow[1] \* bCol[1];

// Emit final key: key, value: result

context.write(key, new IntWritable(result));

}

}

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Simplest Matrix Multiplication");

job.setJarByClass(SimplestMatrixMultiplication.class);

job.setMapperClass(MatrixMapper.class);

job.setReducerClass(MatrixReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

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

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

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

}

}