Matrix Reducer program

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;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

public class MatrixMultiplication {

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

@Override

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

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

String matrixName = elements[0];

int i = Integer.parseInt(elements[1]);

int j = Integer.parseInt(elements[2]);

String cellValue = elements[3];

Configuration conf = context.getConfiguration();

int p = Integer.parseInt(conf.get("p")); // Number of columns in Matrix B

if (matrixName.equals("A")) {

// Emitting key: (i, k), value: A,j,value for all k

for (int k = 0; k < p; k++) {

context.write(new Text(i + "," + k), new Text("A," + j + "," + cellValue));

}

} else if (matrixName.equals("B")) {

// Emitting key: (i, k), value: B,j,value for all i

int m = Integer.parseInt(conf.get("m")); // Number of rows in Matrix A

for (int iIndex = 0; iIndex < m; iIndex++) {

context.write(new Text(iIndex + "," + j), new Text("B," + i + "," + cellValue));

}

}

}

}

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

@Override

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

Map<Integer, Integer> aMap = new HashMap<>();

Map<Integer, Integer> bMap = new HashMap<>();

// Separate values into A and B maps

for (Text val : values) {

String[] elements = val.toString().split(",");

String matrixName = elements[0];

int index = Integer.parseInt(elements[1]);

int value = Integer.parseInt(elements[2]);

if (matrixName.equals("A")) {

aMap.put(index, value);

} else if (matrixName.equals("B")) {

bMap.put(index, value);

}

}

// Calculate the result for cell C[i][k]

int result = 0;

for (Map.Entry<Integer, Integer> entry : aMap.entrySet()) {

int j = entry.getKey();

if (bMap.containsKey(j)) {

result += entry.getValue() \* bMap.get(j);

}

}

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

}

}

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

Configuration conf = new Configuration();

// Pass matrix dimensions to the configuration

conf.set("m", "2"); // Number of rows in matrix A

conf.set("n", "3"); // Number of columns in matrix A and rows in matrix B

conf.set("p", "2"); // Number of columns in matrix B

Job job = Job.getInstance(conf, "matrix multiplication");

job.setJarByClass(MatrixMultiplication.class);

job.setMapperClass(MatrixMapper.class);

job.setReducerClass(MatrixReducer.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(Text.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);

}

}

Matrix A File (e.g., matrix\_a.txt):

A,0,0,1

A,0,1,2

A,0,2,3

A,1,0,4

A,1,1,5

A,1,2,6

Matrix B File (e.g., matrix\_b.txt):

B,0,0,7

B,0,1,8

B,1,0,9

B,1,1,10

B,2,0,11

B,2,1,12

Run Command

hadoop jar matrix\_multiplication.jar MatrixMultiplication matrix\_a.txt matrix\_b.txt output\_dir