Matrix Multiplication via Two Hadoop MapReduce Jobs*

Montasser AKERMI

Last Update: March, 2025

*The course material is hosted at https://akermi.org/c2/.

Learning objectives

Write a MapReduce application to compute the product of two matrices.

Input Data

MatrixName;LineNumber;ColumnNumber;Value

```
M;1;1;1
M;1;2;2
M;1;3;3
M;2;1;3
M;2;2;5
M;2;3;1
M;3;1;0
M;3;2;1
M;3;3;1
N;1;1;3
N;1;2;1
N;1;3;3
N;2;1;2
N;2;2;2
N;2;3;2
N;3;1;1
N;3;2;3
N;3;3;2
```

Output Data

LineNumber;ColumnNumber;Value

```
1;1;10.0
1;3;13.0
2;2;16.0
3;1;3.0
3;3;4.0
1;2;14.0
2;1;20.0
2;3;21.0
3;2;5.0
```

Solution

Listing 1: FirstMapper.java

```
1  import java.io.IOException;
2
3  import org.apache.hadoop.io.Text;
4  import org.apache.hadoop.mapreduce.Mapper;
5
6  public class FirstMapper
7      extends Mapper<Object, Text, Text, Text> {
8
9    public void map(Object key, Text value, Context context
10   ) throws IOException, InterruptedException {
```

```
    String line = value.toString();
    String[] element = line.split(";");

    if ("M".equals(element[0])) {
      context.write(new Text(element[2]), new Text(
          String.format("M;%s;%s", element[1], element[3])));
    } else {
      context.write(new Text(element[1]), new Text(
          String.format("N;%s;%s", element[2], element[3])));
    }
  }
}
```

Listing 2: FirstReducer.java

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class FirstReducer
    extends Reducer<Text, Text, Text, Text> {

  public void reduce(Text key, Iterable<Text> values,
                     Context context
  ) throws IOException, InterruptedException {
    List<List<String>> M = new ArrayList<List<String>>();
    List<List<String>> N = new ArrayList<List<String>>();

    for (Text val : values) {
      String[] element = val.toString().split(";");
      if ("M".equals(element[0])) {
        M.add(Arrays.asList(element));
      } else {
        N.add(Arrays.asList(element));
      }
    }

    for (List<String> elementM : M) {
      for (List<String> elementN : N) {
        float product = Float.parseFloat(elementM.get(2))
            * Float.parseFloat(elementN.get(2));
        context.write(null, new Text(String.format("%s;%s;%.1f",
            elementM.get(1), elementN.get(1), product)));
      }
    }
  }
}
```

Listing 3: SecondMapper.java

```
import java.io.IOException;
```

```
2
3  import org.apache.hadoop.io.Text;
4  import org.apache.hadoop.mapreduce.Mapper;
5
6  public class SecondMapper
7      extends Mapper<Object, Text, Text, Text> {
8
9    public void map(Object key, Text value, Context context
10   ) throws IOException, InterruptedException {
11     String[] element = value.toString().split(";");
12     context.write(new Text(String.format("%s;%s",
13         element[0], element[1])), new Text(element[2]));
14   }
15 }
```

Listing 4: SecondReducer.java

```
1  import java.io.IOException;
2
3  import org.apache.hadoop.io.Text;
4  import org.apache.hadoop.mapreduce.Reducer;
5
6  public class SecondReducer
7      extends Reducer<Text, Text, Text, Text> {
8    public void reduce(Text key, Iterable<Text> values,
9                       Context context
10   ) throws IOException, InterruptedException {
11     float sum = 0.0f;
12
13     for (Text value : values) {
14       sum += Float.parseFloat(value.toString());
15     }
16
17     context.write(null, new Text(String.format(
18         "%s;%.1f", key.toString(), sum)));
19   }
20 }
```

Listing 5: MatrixMultiplication.java

```
1  import java.io.IOException;
2
3  import org.apache.hadoop.conf.Configuration;
4  import org.apache.hadoop.fs.Path;
5  import org.apache.hadoop.io.Text;
6  import org.apache.hadoop.mapreduce.Job;
7  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
8  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
9
10 public class MatrixMultiplication {
11
12   public static void main(String[] args)
13       throws IOException, ClassNotFoundException, InterruptedException {
14     Configuration conf1 = new Configuration();
15     Job job1 = Job.getInstance(conf1, "Matrix Multiplication - Step 1");
```

```
16      job1.setJarByClass(MatrixMultiplication.class);
17      job1.setMapperClass(FirstMapper.class);
18      job1.setReducerClass(FirstReducer.class);
19      job1.setOutputKeyClass(Text.class);
20      job1.setOutputValueClass(Text.class);
21      FileInputFormat.addInputPath(job1, new Path(args[0]));
22      FileOutputFormat.setOutputPath(job1, new Path(args[1]));
23
24      job1.waitForCompletion(true);
25
26      Configuration conf2 = new Configuration();
27      Job job2 = Job.getInstance(conf2, "Matrix Multiplication - Step 2");
28      job2.setJarByClass(MatrixMultiplication.class);
29      job2.setMapperClass(SecondMapper.class);
30      job2.setReducerClass(SecondReducer.class);
31      job2.setOutputKeyClass(Text.class);
32      job2.setOutputValueClass(Text.class);
33      FileInputFormat.addInputPath(job2, new Path(args[1]));
34      FileOutputFormat.setOutputPath(job2, new Path(args[2]));
35      System.exit(job2.waitForCompletion(true) ? 0 : 1);
36    }
37  }
```

Note: It is highly recommended to manually write the code instead of copy/pasting.