Matrix-Vector Multiplication[*]

Montasser AKERMI

Last Update: March, 2025

[*]The course material is hosted at `https://akermi.org/`.

Learning objectives

Write a MapReduce application that computes the product of matrix and a vector.

Input Data

MatrixName;LineNumber(;ColumnNumber);Value

```
M;1;1;4
M;1;2;2
M;1;3;1
M;2;1;3
M;2;2;4
M;2;3;5
M;3;1;2
M;3;2;1
M;3;3;1
V;1;3
V;2;2
V;3;4
```

Output Data

LineNumber;Value

```
1;20
2;37
3;12
```

Solution

Listing 1: FirstMapper.java

```java
import java.io.IOException;

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

public class FirstMapper
    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 {
    String[] line = value.toString().split(";");

    if ("M".equals(line[0])) {
      outputKey.set(line[2]);
      outputValue.set(String.format("M;%s;%s", line[1], line[3]));
      context.write(outputKey, outputValue);
    } else {
      outputKey.set(line[1]);
      outputValue.set(String.format("V;%s", line[2]));
      context.write(outputKey, outputValue);
    }
  }

}
```

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> {

  private Text outputValue = new Text();

  public void reduce(Text key, Iterable<Text> values,
                     Context context) throws IOException, InterruptedException {
    float v = 0.0f;
    List<List<String>> matrixEls = new ArrayList<List<String>>();
    ;
    float product = 0.0f;

    for (Text value : values) {
      String[] element = value.toString().split(";");
      if ("M".equals(element[0])) {
        matrixEls.add(Arrays.asList(element));
      } else {
        v = Float.parseFloat(element[1]);
      }
    }

    if (v != 0) {
      for (List<String> el : matrixEls) {
        product = Float.parseFloat(el.get(2)) * v;
        outputValue.set(String.format("%s;%.2f", el.get(1), product));
        context.write(null, outputValue);
      }
    }
  }

}
```

Listing 3: SecondMapper.java

```
import java.io.IOException;

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

```

```java
public class SecondMapper
    extends Mapper<Object, Text, Text, Text> {

  private Text row = new Text();
  private Text outputValue = new Text();

  public void map(Object key, Text value, Context context
  ) throws IOException, InterruptedException {
    String[] line = value.toString().split(";");
    row.set(line[0]);
    outputValue.set(line[1]);
    context.write(row, outputValue);
  }

}
```

Listing 4: SecondReducer.java

```java
import java.io.IOException;

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

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

  private Text outputValue = new Text();

  public void reduce(Text key, Iterable<Text> values,
                     Context context) throws IOException, InterruptedException {
    float sum = 0.0f;

    for (Text val : values) {
      sum += Float.parseFloat(val.toString());
    }
    outputValue.set(String.valueOf(sum));
    context.write(key, outputValue);
  }
}
```

Listing 5: MatrixVectorMultiplication.java

```java
package tp;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
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;

public class MatrixVectorMultiplication {

  public static void main(String[] args)
      throws IOException, ClassNotFoundException, InterruptedException {
    Configuration conf1 = new Configuration();
    Job job1 = Job.getInstance(conf1, "Matrix Vector Multiplication - Step 1");
    job1.setJarByClass(MatrixVectorMultiplication.class);
    job1.setMapperClass(FirstMapper.class);
    job1.setReducerClass(FirstReducer.class);
    job1.setOutputKeyClass(Text.class);
    job1.setOutputValueClass(Text.class);
    FileInputFormat.addInputPath(job1, new Path(args[0]));
    FileOutputFormat.setOutputPath(job1, new Path(args[1]));
    job1.waitForCompletion(true);

    Configuration conf2 = new Configuration();
    Job job2 = Job.getInstance(conf2, "Matrix Vector Multiplication - Step 2");
    job2.setJarByClass(MatrixVectorMultiplication.class);
    job2.setMapperClass(SecondMapper.class);
    job2.setReducerClass(SecondReducer.class);
    job2.setOutputKeyClass(Text.class);
    job2.setOutputValueClass(Text.class);
    FileInputFormat.addInputPath(job2, new Path(args[1]));
    FileOutputFormat.setOutputPath(job2, new Path(args[2]));
    job2.waitForCompletion(true);
  }
}
```

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