

CSE 488 (Section 1) [Summer 2022]

Lab 02 Assignment Submission Report

Assignment Title: MapReduce Programming

Submitted by: Rizvee Hassan Prito 2019-3-60-041

1. Screenshots

Problem 1:

Driver class:

```
☐ Package Explorer ☎
                  □ 🖟 AVGdriver.java 🛭 🗓 AVGmapper.java 🔑 AVGreducer.java
            E 🕏 | 🔊
                         11 public class AVGdriver{
                         12
                              public static void main(String[] args) throws Exception {
   AVGdriver.java
                                \ensuremath{^{*}} Validate that two arguments were passed from the command line.
      AVGmapper.java
    D AVGreducer.java
  ▶ M JRE System Library [JavaSE-1.7]
  ▶ ■ Referenced Libraries
                         18
                               if (args.length != 2) {
                                  System.out.printf("Usage: StubDriver <input dir> <output dir>\n");
                         19
                                 System.exit(-1);
                        21
22
23
24
25
26
27
28
29
30
                               Configuration config= new Configuration();
                                Path input= new Path(args[0]);
                               Path output= new Path(args[1]);
                                \overset{'}{\ast} Instantiate a Job object for your job's configuration. ^{\ast\prime}
                         31
                                @SuppressWarnings("deprecation")
                        32
33
                               Job job = new Job(config, "WordCount2");
                      □ 🖟 AVGdriver.java 🛭 🗓 AVGmapper.java 🔑 AVGreducer.java
33
               🖹 🔄 🖢
                              34
* Specify the jar file that contains your driver, <u>mapper</u>, and reducer.
                              35
  マ 🔑 src
                                       * Hadoop will transfer this jar file to nodes in your cluster running
    36
                                       * mapper and reducer tasks.
                              37
    AVGdriver.java
     D AVGmapper.java
                              38
      39
                                      job.setJarByClass(AVGdriver.class);
  ▶ MIRE System Library [JavaSE-1.7]
                              40
                                      job.setMapperClass(AVGmapper.class);
  ▶ ■ Referenced Libraries
                              41
                                      job.setReducerClass(AVGreducer.class);
                              42
                                      job.setOutputKeyClass(Text.class);
                             43
                                      job.setOutputValueClass(DoubleWritable.class);
                             44
                                      FileInputFormat.addInputPath(job, input);
                             45
                                      FileOutputFormat.setOutputPath(job, output);
                             46
                                       * TODO implement
                             47
                             48
                                       */
                             49
                             50
                             51
                                       * Start the MapReduce job and wait for it to finish.
                             52
                                       * If it finishes successfully, return 0. If not, return 1.
                             53
                                      boolean success = job.waitForCompletion(true);
                             54
                             55
                                      System.exit(success ? 0 : 1);
                             56
```

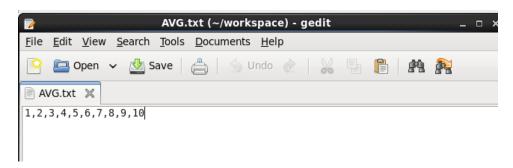
Mapper class:

```
1 import java.io.IOException;
           3 import org.apache.hadoop.io.DoubleWritable;
 4 import org.apache.hadoop.io.LongWritable;
  5 import org.apache.hadoop.io.Text;
   AVGdriver.java
    D AVGmapper.java
                        6 import org.apache.hadoop.mapreduce.Mapper;
    D AVGreducer.java
 ▶ ■ JRE System Library [JavaSE-1.7]
                        8 public class AVGmapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {
 ▶ ■ Referenced Libraries
                       10⊝
                            @Override
                            public void map(LongWritable key, Text value, Context context)
                       △11
                       12
                                throws IOException, InterruptedException {
                       13
                       14
                              String line = value.toString();
                       15
                              String[] words= line.split(",");
                       16
                              for (String word: words){
                       17
                                  Text outputKey = new Text("Avg");
                       18
                                  DoubleWritable outputValue = new DoubleWritable(Integer.parseInt(word));
                       19
                                  context.write(outputKey,outputValue);
                       20
                              }
                       21
                       22
                            }
                       23 }
```

Reducer class:

```
☐ Package Explorer 🛭
                 - -
                      🖹 🔄 | 🐉 🔻
                        1 import java.io.IOException;
3 import org.apache.hadoop.io.DoubleWritable;
 4 import org.apache.hadoop.io.IntWritable;
    AVGdriver.java
                       5 import org.apache.hadoop.io.Text;
    D AVGmapper.java
                        6 import org.apache.hadoop.mapreduce.Reducer;
 🛮 8 public class AVGreducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> [
 ▶ ➡ Referenced Libraries
                       10⊝
                           @Override
                            public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
                       11
                       12
                               throws IOException, InterruptedException {
                       13
                       14
                            double sum=0;
                            double count=0;
                       15
                       16
                            for (DoubleWritable value:values){
                                sum+=value.get();
                       17
                       18
                                count++;
                       19
                       20
                            context.write(key, new DoubleWritable(sum/count));
                       21
```

Input File:



Terminal:

```
עב/טט/עט טס:עס: אורט mapreduce.Job: map ט% reduce ט%
22/06/26 05:25:51 INFO mapreduce.Job: map 100% reduce 0%
22/06/26 05:26:02 INFO mapreduce.Job: map 100% reduce 100%
22/06/26 05:26:03 INFO mapreduce.Job: Job job 1656243722115 0002 completed successfully
22/06/26 05:26:04 INFO mapreduce.Job: Counters: 49
       File System Counters
               FILE: Number of bytes read=146
               FILE: Number of bytes written=223019
               FILE: Number of read operations=0
               FILE: Number of large read operations=0
               FILE: Number of write operations=0
               HDFS: Number of bytes read=139
               HDFS: Number of bytes written=8
               HDFS: Number of read operations=6
               HDFS: Number of large read operations=0
               HDFS: Number of write operations=2
       Job Counters
               Launched map tasks=1
               Launched reduce tasks=1
               Data-local map tasks=1
               Total time spent by all maps in occupied slots (ms)=6732
               Total time spent by all reduces in occupied slots (ms)=8178
               Total time spent by all map tasks (ms)=6732
               Total time spent by all reduce tasks (ms)=8178
               Total vcore-seconds taken by all map tasks=6732
               Total vcore-seconds taken by all reduce tasks=8178
               Total megabyte-seconds taken by all map tasks=6893568
               Total megabyte-seconds taken by all reduce tasks=8374272
                Total megabyte-seconds taken by all reduce tasks=8374272
        Map-Reduce Framework
                Map input records=1
                Map output records=10
                Map output bytes=120
                Map output materialized bytes=146
                Input split bytes=118
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=146
                Reduce input records=10
                Reduce output records=1
                Spilled Records=20
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=176
                CPU time spent (ms)=1650
                Physical memory (bytes) snapshot=334729216
                Virtual memory (bytes) snapshot=3008466944
                Total committed heap usage (bytes)=226365440
        Shuffle Errors
                BAD ID=0
                CONNECTION=0
                IO ERROR=0
                WRONG LENGTH=0
                WRONG MAP=0
                WRONG REDUCE=0
        File Input Format Counters
                Bytes Read=21
        File Output Format Counters
         Bytes Written=8
```

Output:

```
Home / user / cloudera / AVG / part-r-00000
```

```
Avg 5.5
```

Problem 2:

Driver class:

```
☐ Package Explorer 🛭
                     □ □ AVG2driver.java ☎ 🔝 AVG2mapper.java 🕡 AVG2reducer.java
              □ 🕏 👂 🔻
                            35
                                     ,* Specify the jar file that contains your driver, <u>mapper</u>, and reducer.

* <u>Hadoop</u> will transfer this jar file to nodes in your cluster running
▼ 🚰 AVG2
 * mapper and reducer tasks.
     AVG2driver.java
     AVG2mapper.java
                                     job.setJarByClass(AVG2driver.class);
     AVG2reducer.java
                                     job.setMapperClass(AVG2mapper.class);
 ▶ MIRE System Library [JavaSE-1.7]
                                     job.setReducerClass(AVG2reducer.class);
 ▶ ■ Referenced Libraries
                             43
                                     job.setOutputKeyClass(Text.class);
                             44
45
46
                                     job.setOutputValueClass(DoubleWritable.class);
                                     FileInputFormat.addInputPath(job, input);
                                     FileOutputFormat.setOutputPath(job, output);
                             47
                                      * TODO implement
                             48
                             49
                             50
                             51
                             52
53
                                      * Start the MapReduce job and wait for it to finish.
                                      * If it finishes successfully, return 0. If not, return 1.
                             54
                             55
                                     boolean success = job.waitForCompletion(true);
                             56
                                     System.exit(success ? 0 : 1);
```

Mapper class:

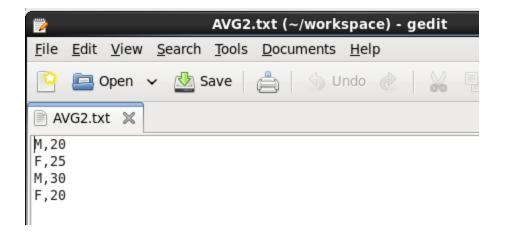
```
    AVG2driver.java
    AVG2mapper.java 
    AVG2reducer.java

            1 import java.io.IOException;
3 import org.apache.hadoop.io.DoubleWritable;
 4 import org.apache.hadoop.io.LongWritable;
  5 import org.apache.hadoop.io.Text;
    AVG2driver.java
    AVG2mapper.java
                          6 import org.apache.hadoop.mapreduce.Mapper;
    AVG2reducer.java
 ▶ Mark JRE System Library [JavaSE-1.7]
                          8 public class AVG2mapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {
 ▶ ■ Referenced Libraries
                          9
                        10°
11
12
                              public void map(LongWritable key, Text value, Context context)
                                  throws IOException, InterruptedException {
                         13
                         14
                                String line = value.toString();
                                String[] words= line.split("\n");
                         15
                        16
                                for (String word: words){
                         17
                                    String[] word2= line.split(",");
                         18
                                    Text outputKey = new Text(word2[0]);
                                    DoubleWritable outputValue = new DoubleWritable(Integer.parseInt(word2[1]));
                         19
                         20
                                     context.write(outputKey,outputValue);
                         21
                         22
                        23
                              }
                         24 }
```

Reducer class:

```
1 import java.io.IOException;
           3 import org.apache.hadoop.io.DoubleWritable;
 😘 4 import org.apache.hadoop.io.IntWritable;
  5 import org.apache.hadoop.io.Text;
   AVG2driver.java
                       6 import org.apache.hadoop.mapreduce.Reducer;
   D AVG2mapper.java
   AVG2reducer.java
 8 public class AVG2 reducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {
 ▶ ■ Referenced Libraries
                      10⊝
                          public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
                      12
                              throws IOException, InterruptedException {
                      13
                      14
                           double sum=0;
                      15
                           double count=0;
                      16
                           for (DoubleWritable value:values){
                      17
                               sum+=value.get();
                      18
                               count++;
                      19
                      20
                           context.write(key, new DoubleWritable(sum/count));
                      21
                     22 }
```

Input File:



Terminal:

```
22/06/26 08:26:17 INFO mapreduce.Job: Job job_1656256202977_0001 running in uber mode : false
22/06/26 08:26:17 INFO mapreduce.Job: map 0% reduce 0%
22/06/26 08:26:25 INFO mapreduce.Job: map 100% reduce 0%
22/06/26 08:26:36 INFO mapreduce.Job: map 100% reduce 100%
22/06/26 08:26:36 INFO mapreduce.Job: Job job_1656256202977_0001 completed successfully
22/06/26 08:26:36 INFO mapreduce.Job: Counters: 49
         File System Counters
                  FILE: Number of bytes read=54
                  FILE: Number of bytes written=222843
                  FILE: Number of read operations=0
FILE: Number of large read operations=0
                  FILE: Number of write operations=0
HDFS: Number of bytes read=139
                  HDFS: Number of bytes written=14
HDFS: Number of read operations=6
                  HDFS: Number of large read operations=0
HDFS: Number of write operations=2
         Job Counters
                  Launched map tasks=1
                  Launched reduce tasks=1
                  Data-local map tasks=1
                  Total time spent by all maps in occupied slots (ms)=6087
                  Total time spent by all reduces in occupied slots (ms)=7816
                  Total time spent by all map tasks (ms)=6087
Total time spent by all reduce tasks (ms)=7816
                  Total vcore-seconds taken by all map tasks=6087
                  Total vcore-seconds taken by all reduce tasks=7816
                  Total megabyte-seconds taken by all map tasks=6233088
                  Total megabyte-seconds taken by all reduce tasks=8003584
```

```
TOTAL MEGADYIC-SECONAS CAREN DY ALL TEAUCE CASES-0005507
Map-Reduce Framework
           Map input records=4
           Map output records=4
           Map output bytes=40
           Map output materialized bytes=54
           Input split bytes=119
           Combine input records=0
           Combine output records=0
           Reduce input groups=2
Reduce shuffle bytes=54
           Reduce input records=4
           Reduce output records=2
           Spilled Records=8
           Shuffled Maps =1
           Failed Shuffles=0
           Merged Map outputs=1
           GC time elapsed (ms)=176
           CPU time spent (ms)=1530
Physical memory (bytes) snapshot=347967488
Virtual memory (bytes) snapshot=3008630784
Total committed heap usage (bytes)=226365440
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
           WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=20
File Output Format Counters
Bytes Written=14
```

Output:

Problem 3:

Driver class:

```
☐ Package Explorer ☎
                    _ _

⚠ AgeGroupsdriver.java 
☒ ⚠ AgeGroupsmapper.java 
☒ AgeGroupsreducer.java
             E 😩 | 🐤
                           30
                                   @SuppressWarnings("deprecation")
31
                                   Job job = new Job(config, "AgeGroups");
 32
   33
    ▶ Jn AgeGroupsdriver.iava
                                    \ensuremath{^{*}} Specify the jar file that contains your driver, \underline{\ensuremath{\mathsf{mapper}}} , and reducer.
                           34
     AgeGroupsmapper.java
                                    * Hadoop will transfer this jar file to nodes in your cluster running
                           35
     AgeGroupsreducer.iava
                           36
                                      mapper and reducer tasks.
 ▶ ■ IRE System Library [JavaSE-1.7]
                           37
 ▶ ➡ Referenced Libraries
                                   job.setJarByClass(AgeGroupsdriver.class);
                           38
job.setMapperClass(AgeGroupsmapper.class);
                           39
 40
                                    job.setReducerClass(AgeGroupsreducer.class);
 ▶ ■ JRE System Library [JavaSE-1.7]
                           41
                                   job.setOutputKeyClass(Text.class);
 ▶ ➡ Referenced Libraries
                           42
                                   job.setOutputValueClass(IntWritable.class);
                                   FileInputFormat.addInputPath(job, input);
                           43
                           44
                                   FileOutputFormat.setOutputPath(job, output);
                           45
                                    * TODO implement
                          246
                           47
                           48
                           49
                                    * Start the MapReduce job and wait for it to finish.
                           50
                           51
                                    * If it finishes successfully, return 0. If not, return 1.
                           52
                           53
                                   boolean success = job.waitForCompletion(true);
                           54
                                   System.exit(success ? 0 : 1);
                          55
                          56 }
```

Mapper class:

```
☐ Package Explorer ☎
                   _ =
                         ⚠ AgeGroupsdriver.java
⚠ AgeGroupsmapper.java
☒ D AgeGroupsreducer.java
                           9 public class AgeGroupsmapper extends Mapper<LongWritable, Text, Text, IntWritable> {
            100
11
                               public void map(LongWritable key, Text value, Context context)
 12
                                   throws IOException, InterruptedException {
   AgeGroupsdriver.java
                          13
                          14
                                 String line = value.toString();
    AgeGroupsmapper.iava
                          15
                                 String[] words= line.split("\n");
    AgeGroupsreducer.java
                          16
                                 for (String word: words){
 String[] word2= line.split(",");
 ▶ ➡ Referenced Libraries
                          17
                                      if(Integer.parseInt(word2[1])>=1 & Integer.parseInt(word2[1])<=25){</pre>
18
                          19
                                          Text outputKey = new Text("1-25");
                                          IntWritable outputValue = new IntWritable(1);
                          20
                          21
                                          context.write(outputKey,outputValue);
                          22
                          23
                                      else if(Integer.parseInt(word2[1])>=26 & Integer.parseInt(word2[1])<=50){</pre>
                          24
                                         Text outputKey = new Text("26-50");
IntWritable outputValue = new IntWritable(1);
                          25
                          26
                                          context.write(outputKey,outputValue);
                          27
                          28
                                      else if(Integer.parseInt(word2[1])>=51 & Integer.parseInt(word2[1])<=75){</pre>
                                          Text outputKey = new Text("51-75");
                          29
                                          IntWritable outputValue = new IntWritable(1);
                          30
                                          context.write(outputKey,outputValue);
                         32
                                     }
                                  else if(Integer.parseInt(word2[1])>=76 & Integer.parseInt(word2[1])<=100){</pre>
                                      Text outputKey = new Text("76-100");
                       35
                                      IntWritable outputValue = new IntWritable(1);
                                      context.write(outputKey,outputValue);
```

Reducer class:

```
☐ Package Explorer ☎

1⊕ import java.io.I0Exception;

8 public class AgeGroupsreducer extends Reducer<Text, IntWritable, Text, IntWritable> {
 10⊝
  AgeGroupsdriver.java
                     11
12
    🕨 <u> </u> AgeGroupsmapper.java
                           public void reduce(Text key, Iterable<IntWritable> values, Context context)
    AgeGroupsreducer.java
                               throws IOException, InterruptedException {
                     13
 ▶ ■ JRE System Library [JavaSE-1.7]
 Referenced Libraries
                      14
                            int sum=0;
for (IntWritable value:values){
                      15
                      16
                               sum+=value.get();
                      17
                           context.write(key, new IntWritable(sum));
                      18
                     19 }
                      20 }
```

Input File:

Terminal:

```
22/06/26 08:51:50 INFO mapreduce.Job: map 0% reduce 0%
22/06/26 08:51:59 INFO mapreduce.Job: map 100% reduce 0%
22/06/26 08:52:10 INFO mapreduce.Job: map 100% reduce 100%
22/06/26 08:52:10 INFO mapreduce.Job: Job job 1656256202977 0003 completed successfully
22/06/26 08:52:11 INFO mapreduce.Job: Counters: 49
          File System Counters
                   FILE: Number of bytes read=89
                   FILE: Number of bytes written=222947
                   FILE: Number of read operations=0
                   FILE: Number of large read operations=0
                   FILE: Number of write operations=0
                   HDFS: Number of bytes read=159
                   HDFS: Number of bytes written=32
                   HDFS: Number of read operations=6
                   HDFS: Number of large read operations=0
                   HDFS: Number of write operations=2
          Job Counters
                   Launched map tasks=1
                   Launched reduce tasks=1
                   Data-local map tasks=1
                   Total time spent by all maps in occupied slots (ms)=6859
                   Total time spent by all reduces in occupied slots (ms)=7459
                   Total time spent by all map tasks (ms)=6859
                   Total time spent by all reduce tasks (ms)=7459
                   Total vcore-seconds taken by all map tasks=6859
                   Total vcore-seconds taken by all reduce tasks=7459
Total megabyte-seconds taken by all map tasks=7023616
                   Total megabyte-seconds taken by all reduce tasks=7638016
                 TOTAL MEGADYTE-SECONAS TAKEN DY ALL TEAUCE LASKS-7050010
        Map-Reduce Framework
                Map input records=7
Map output records=7
                 Map output bytes=69
                Map output materialized bytes=89
                Input split bytes=124
Combine input records=0
                 Combine output records=0
                Reduce input groups=4
Reduce shuffle bytes=89
                 Reduce input records=7
                Reduce output records=4
Spilled Records=14
                 Shuffled Maps =1
                Failed Shuffles=0
Merged Map outputs=1
                 GC time elapsed (ms)=182
                 CPU time spent (ms)=1550
                Physical memory (bytes) snapshot=357978112
Virtual memory (bytes) snapshot=3008475136
Total committed heap usage (bytes)=226365440
        Shuffle Errors
BAD_ID=0
                 CONNECTION=0
                IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
Bytes Read=35
        File Output Format Counters
                 Bytes Written=32 _
```

Output:

76-100 1

Learning Outcome:

From this lab we have learned how to solve problems by using MapReduce programming style and execute the program in Hadoop Framework.