
Lab 01: A Gentle Introduction to Hadoop

CSC14118 Introduction to Big Data 20KHMT1

The Girls

2023-02-17

Contents

1	Lab 02: MapReduce programming	3
1.1	Work assignment	3
1.2	Explain the code in detail.	3
1.2.1	Problem 1: WordCount	3
1.2.1.1	Mapper	3
1.2.1.2	Reducer	4
1.2.1.3	2.1.3. Guide to run the program	5
1.2.1.4	2.1.4. Self-evaluation	8
1.2.1.5	2.1.5. References	8
1.2.2	2.2. Problem 2: WordSizeWordCount Program	8
1.2.2.1	2.2.1. Mapper	8
1.2.2.2	2.2.2. Reducer	8
1.2.2.3	2.2.3. Guide to run the program	8
1.2.2.4	2.2.4. Self-evaluation	8
1.2.2.5	2.2.5. References	8
1.2.3	2.3. Problem 3: WeatherData Program	8
1.2.3.1	2.3.1. Mapper	8
1.2.3.2	2.3.2. Reducer	8
1.2.3.3	2.3.3. Guide to run the program	8
1.2.3.4	2.3.4. Self-evaluation	8
1.2.3.5	2.3.5. References	8
1.2.4	2.4. Problem 4: Patent Program	8
1.2.4.1	2.4.1. Mapper	8
1.2.4.2	2.4.2. Reducer	8
1.2.4.3	2.4.3. Guide to run the program	8
1.2.4.4	2.4.4. Self-evaluation	8
1.2.4.5	2.4.5. References	8
1.2.5	2.5. Problem 5: MaxTemp Program	8
1.2.5.1	2.5.1. Mapper	8
1.2.5.2	2.5.2. Reducer	8

1.2.5.3	2.5.3. Guide to run the program	8
1.2.5.4	2.5.4. Self-evaluation	8
1.2.5.5	2.5.5. References	8
1.2.6	2.6. Problem 6: AverageSalary Program	8
1.2.6.1	2.6.1. Mapper	8
1.2.6.2	2.6.2. Reducer	8
1.2.6.3	2.6.3. Guide to run the program	8
1.2.6.4	2.6.4. Self-evaluation	8
1.2.6.5	2.6.5. References	8
1.2.7	2.7. Problem 7: De Identify HealthCare Program	8
1.2.7.1	2.7.1. Mapper	8
1.2.7.2	2.7.2. Reducer	8
1.2.7.3	2.7.3. Guide to run the program	8
1.2.7.4	2.7.4. Self-evaluation	8
1.2.7.5	2.7.5. References	8
1.2.8	2.8. Problem 8: Music Track Program	8
1.2.8.1	2.8.1. Mapper	8
1.2.8.2	2.8.2. Reducer	8
1.2.8.3	2.8.3. Guide to run the program	8
1.2.8.4	2.8.4. Self-evaluation	8
1.2.8.5	2.8.5. References	8
1.2.9	2.9. Problem 9: Telecom Call Data Record Program	8
1.2.9.1	2.9.1. Mapper	8
1.2.9.2	2.9.2. Reducer	8
1.2.9.3	2.9.3. Guide to run the program	8
1.2.9.4	2.9.4. Self-evaluation	8
1.2.9.5	2.9.5. References	8
1.2.10	2.10. Problem 10: Count Connected Component Program	8
1.2.10.1	2.10.1. Mapper	8
1.2.10.2	2.10.2. Reducer	8
1.2.10.3	2.10.3. Guide to run the program	8
1.2.10.4	2.10.4. Self-evaluation	8
1.2.10.5	2.10.5. References	8
1.3	References	8

1 Lab 02: MapReduce programming

1.1 Work assignment

Student ID	Full name	Work assignment
20127011	Le Tan Dat	Problem 1, 4, report
20127438	Le Nguyen Nguyen Anh	Problem 2, 6, 8
20127458	Dang Tien Dat	Problem 3, 7, 10 report
20127627	Nguyen Quoc Thang	Problem 5, 9

-> Our team consulted the lab requirement file in drive lab 2, besides we also solved problems such as:

1.2 Explain the code in detail.

1.2.1 Problem 1: WordCount

1.2.1.1 Mapper

```
public static class WordCountMapper extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable number = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context) throws IOException,
        ↪ InterruptedException {
        StringTokenizer token = new StringTokenizer(value.toString());
        while (token.hasMoreTokens()) {
            word.set(token.nextToken());
            context.write(word, number);
        }
    }
}
```

```
}  
}  
}
```

- The mapper class is a subclass of the Mapper class.
- 2 variables are declared:
 - number is a constant variable with value 1 to count the number of words.
 - word is a variable of type Text to store the word.
- Idea of the mapper class:
 - The mapper class will read the input file line by line.
 - Then, the mapper class will split the line into words by using the StringTokenizer class.
 - After that, the mapper class will loop through the words and write the word and the number 1 to the context.
 - The context will be used to write the output file.

1.2.1.2 Reducer

```
public static class WordCountReducer extends Reducer<Text,IntWritable,Text,IntWritable> {  
    private IntWritable result = new IntWritable();  
  
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws  
        IOException, InterruptedException {  
        int sum = 0;  
        for (IntWritable val : values) {  
            sum += val.get();  
        }  
        result.set(sum);  
        context.write(key, result);  
    }  
}
```

- The reducer class is a subclass of the Reducer class to reduce the output of the mapper class.
- 1 variable is declared:
 - result is a variable of type IntWritable to store the number of words is counted.
- Idea of the reducer class:
 - The reducer class will read the output file of the mapper class line by line.
 - Then, the reducer class will split the line into words and the number of words by using the StringTokenizer class.

- After that, initialize the variable `sum` to 0. This variable is used to count the number of words.
- The reducer class will loop through the words and count the number of words by using the variable `sum`.
- Finally, set the value of the variable `result` to `sum` and write the word and the number of words to the context.

1.2.1.3 2.1.3. Guide to run the program

- Step 1: Create file `WordCount.java` in the folder `src` of the project.
- Step 2: Create file `wordcount.txt` in the folder `data` of the project and then put the file to the local HDFS by using the command

```
hdfs dfs -mkdir /input
```

```
hdfs dfs -put data/wordcount.txt /input
```

```
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing$ hdfs dfs -mkdir /input
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing$ hdfs dfs -put data/wordcount.txt /input
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing$ hdfs dfs -ls /input
Found 1 items
-rw-r--r--  1 dat20127458 supergroup    1305 2023-03-31 14:06 /input/wordcount.txt
```

Figure 1.1: Step 2

- Step 3: Compile and run the program by using the command
 - `javac` is a command-line tool that compiles Java source code into Java bytecode.

```
hadoop com.sun.tools.javac.Main WordCount.java
```

- `jar` is a command-line tool that creates a Java archive file (JAR) from a set of Java class files.

```
jar cf wc.jar WordCount*.class
```

- `hadoop` is a command-line tool that runs a MapReduce job. `hadoop jar wc.jar WordCount /input /output`

- Step 4: Check the result by using the command

```
hdfs dfs -cat /output/part-r-00000
```

```

dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing/src$ cd problem01
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing/src/problem01$ ls
WordCount.java
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing/src/problem01$ hadoop com.sun.tools.javac.Main WordCount.java
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing/src/problem01$ jar cf wc.jar WordCount*.class
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing/src/problem01$ hadoop jar wc.jar WordCount /input /output
2023-03-31 14:08:07,972 INFO Impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-03-31 14:08:08,170 INFO Impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-03-31 14:08:08,170 INFO Impl.MetricsSystemImpl: JobTracker metrics system started
2023-03-31 14:08:08,725 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execu
te your application with ToolRunner to remedy this.
2023-03-31 14:08:08,910 INFO input.FileInputFormat: Total input files to process : 1
2023-03-31 14:08:08,953 INFO mapreduce.JobSubmitter: number of splits:1
2023-03-31 14:08:09,135 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1220648815_0001
2023-03-31 14:08:09,135 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-03-31 14:08:09,355 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-03-31 14:08:09,356 INFO mapreduce.Job: Running job: job_local1220648815_0001
2023-03-31 14:08:09,361 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2023-03-31 14:08:09,385 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-03-31 14:08:09,385 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cl
eanup failures: false
2023-03-31 14:08:09,387 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2023-03-31 14:08:09,453 INFO mapred.LocalJobRunner: Waiting for map tasks
2023-03-31 14:08:09,456 INFO mapred.LocalJobRunner: Starting task: attempt local1220648815_0001_m_000000_0
2023-03-31 14:08:09,518 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2023-03-31 14:08:09,518 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cl
eanup failures: false
2023-03-31 14:08:09,553 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
2023-03-31 14:08:09,563 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/input/wordcount.txt:0+1305
2023-03-31 14:08:09,718 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2023-03-31 14:08:09,718 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2023-03-31 14:08:09,718 INFO mapred.MapTask: soft limit at 83886080
2023-03-31 14:08:09,718 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2023-03-31 14:08:09,718 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2023-03-31 14:08:09,725 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2023-03-31 14:08:10,365 INFO mapreduce.Job: Job job_local1220648815_0001 running in uber mode : false
2023-03-31 14:08:10,366 INFO mapreduce.Job: map 0% reduce 0%
2023-03-31 14:08:10,572 INFO mapred.LocalJobRunner:
2023-03-31 14:08:10,601 INFO mapred.MapTask: Starting flush of map output

```

Figure 1.2: Step 3

```
dat20127458@TienDat57:/mnt/d/WORK/Lab-BigData/Lab2_MapReduce-programing/src/problem01$ hadoop fs -cat /output/part-r-00000
In 1
Infinite, 1
Nobody 1
This 1
We 1
When 1
Whether 1
Worry, 1
Years 1
Youth 2
a 11
adventure 1
aerials 2
and 8
appetite 1
appetite, 1
are 4
as 3
at 2
back 1
beauty, 1
being's 1
body 1
bows 1
but 2
by 2
catch 1
center 1
cheeks, 1
cheer, 1
child-like 1
courage 2
covered 1
cynicism 1
deep 1
deserting 1
die 1
down, 1
dust. 1
ease. 1
eighty. 1
emotions; 1
enthusiasm 1
even 1
every 1
exists 1
fear, 1
for 1
```

Figure 1.3: Step 3

1.2.1.4 2.1.4. Self-evaluation**1.2.1.5 2.1.5. References****1.2.2 2.2. Problem 2: WordSizeWordCount Program****1.2.2.1 2.2.1. Mapper****1.2.2.2 2.2.2. Reducer****1.2.2.3 2.2.3. Guide to run the program****1.2.2.4 2.2.4. Self-evaluation****1.2.2.5 2.2.5. References****1.2.3 2.3. Problem 3: WeatherData Program****1.2.3.1 2.3.1. Mapper****1.2.3.2 2.3.2. Reducer****1.2.3.3 2.3.3. Guide to run the program****1.2.3.4 2.3.4. Self-evaluation****1.2.3.5 2.3.5. References****1.2.4 2.4. Problem 4: Patent Program****1.2.4.1 2.4.1. Mapper****1.2.4.2 2.4.2. Reducer****1.2.4.3 2.4.3. Guide to run the program****1.2.4.4 2.4.4. Self-evaluation****1.2.4.5 2.4.5. References****1.2.5 2.5. Problem 5: MaxTemp Program****1.2.5.1 2.5.1. Mapper****1.2.5.2 2.5.2. Reducer****1.2.5.3 2.5.3. Guide to run the program****1.2.5.4 2.5.4. Self-evaluation**

- https://docs.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_wordcount1.html
 - https://docs.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_wordcount2.html
 - https://docs.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_wordcount3.html
- Book: MapReduce Design Patterns [Donald Miner, Adam Shook, 2012]
- All of StackOverflow link related.
- Set up Hadoop Cluster
 - <https://www.linode.com/docs/guides/how-to-install-and-set-up-hadoop-cluster/>
 - <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/ClusterSetup.html>
- Slide of course.