Large Scale Data Management

Kagioglou Maria

14/02/2025

PART I

Aim

The aim of this part is to build and run a Hadoop MapReduce Application inside Docker.

Steps

1. `vagrant up` : Starts the virtual machine (VM) using Vagrant → uses the configuration in the Vagrantfile to set up and launch the VM.

2. `vagrant ssh` : Opens an SSH session to the running virtual machine created by Vagrant → this allows access the VM's terminal and execute commands within it.

3. `docker ps` : Lists all running Docker containers.

4. `cd /vagrant/hadoop-mapreduce-examples/` : Changes the working directory inside the VM to the hadoop-mapreduce-examples folder.

5. `wget https://www.gutenberg.org/cache/epub/1342/pg1342.txt-O/vagrant/PrideAndPrejudice.txt`: Downloads the text file of "Pride and Prejudice" from Project Gutenberg.

6. `docker cp /vagrant/hadoop-mapreduce-examples/PrideAndPrejudice.txt namenode:/tmp/`: Copies the "Pride and Prejudice" text file from the host system (inside /vagrant/...) to the namenode container. The file is placed in the /tmp/ directory within the namenode container.

7. `docker exec namenode hdfs dfs -mkdir -p /user/hdfs/input` : Creates a directory in HDFS (/user/hdfs/input) where input files for the MapReduce job will be stored.

8. `docker exec namenode hdfs dfs -put /tmp/PrideAndPrejudice.txt /user/hdfs/input/` : Uploads the "Pride and Prejudice" file to HDFS. → the file is moved from the namenode container's /tmp/ directory to HDFS under /user/hdfs/input/.

9. `docker exec namenode hdfs dfs -cat/user/hdfs/input/PrideAndPrejudice.txt | head -20` : Displays the first 20 lines of the uploaded file from HDFS

```
The Project Gutenberg eBook of Pride and Prejudice

This ebook is for the use of anyone anywhere in the United States and
most other parts of the world at no cost and with almost no restrictions
whatsoever. You may copy it, give it away or re-use it under the terms
of the Project Gutenberg License included with this ebook or online
at www.gutenberg.org. If you are not located in the United States,
you will have to check the laws of the country where you are located
before using this eBook.

Title: Pride and Prejudice

Author: Jane Austen

Release date: June 1, 1998 [eBook #1342]
Most recently updated: October 29, 2024

Language: English

Credits: Chuck Greif and the Online Distributed Proofreading Team at

http://www.pgdp.net

(This file was produced from images available at The Internet Archive)
```

10. `docker exec namenode hdfs dfs -rm -r /user/hdfs/output/` : Deletes the /user/hdfs/output/ directory in HDFS, because previously MobyDick was written in this directory.

11. `docker exec namenode hdfs dfs -ls /user/hdfs/output` : Lists the contents of the /user/hdfs/output/ directory in HDFS. This confirms whether the MapReduce job successfully wrote its output.

12. `find / -name "hadoop-mapreduce-examples*.jar"` : The hadoop-mapreduce-examples.jar file contains prebuilt MapReduce programs like wordcount, which are used for testing and learning. This command helps you locate the exact path of the JAR file, so you can use it in subsequent commands.

13. `docker exec -it namenode hadoop jar/opt/hadoop -3.2.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount/user/hdfs/input/PrideAndPrejudice.txt/user/hdfs/output/` : Runs the WordCount job. The output of this job will be stored in the /user/hdfs/output/ directory.

14. `docker exec namenode hdfs dfs -cat /user/hdfs/output/part-r-00000 | head -100` : Displays the first 100 lines of the MapReduce job's output file (part-r-00000).

```
#1342]   1
$5,000)  1
&        1
($1      1
(801)    1
```

```
(By      1
(I       1
(Lady    1
(This    1
(a)      1
(affection       1
(an      1
(and     4
(any     1
(arising         1
(b)      1
(by-the-bye,     1
(c)      1
(does    1
(for     4
(glancing        1
(her     1
(if      4
(it      1
(like    1
(most    1
(my      1
(not     1
(or      3
(though  1
(to      1
(trademark/copyright)    1
(unasked         1
(what    1
(which   1
(who,    1
(www.gutenberg.org),     1
(  the     1
***      4
*/       9
-        3
----     2
----,    1
----;    1
----shire        8
----shire,       2
----shire.       2
---- s h i r e        2
/*       9
1        1
1,       1
1.       1
1.A.     1
```

```
1.B.     1
1.C      1
1.C.     1
1.D.     1
1.E      1
1.E.     1
1.E.1    3
1.E.1.   2
1.E.2.   1
1.E.3.   1
1.E.4.   1
1.E.5.   1
1.E.6.   1
1.E.7    2
1.E.7.   1
1.E.8    2
1.E.8.   2
1.E.9.   3
1.F.     1
1.F.1.   1
1.F.2.   1
1.F.3,   3
1.F.3.   2
1.F.4.   1
1.F.5.   1
1.F.6.   1
10       1
108      1
113      1
118      1
12       1
132      1
139      1
143      1
146      1
148      1
15       1
1500     1
154      1
156      2
156.     1
161      1
166      1
168      1
175      1
177      1
1796,    1
```

Part II

Aim

In this part of the project, you will develop a MapReduce application that processes a car_price.csv file containing car sales records. For each unique seller and month pair (e.g., "kia motors america inc:2024-12"): we will find the car with the highest (selling price - MMR) difference and we will compute the average difference for all cars sold by the same seller in that month.

Steps

1. `docker cp car_prices.csv namenode:/tmp/car_prices.csv` : Copy CSV to Namenode Container.

2. `docker exec namenode hdfs dfs -put /tmp/car_prices.csv /user/hdfs/input/` : Put the file into HDFS.

3. `docker exec namenode hdfs dfs -ls /user/hdfs/input/` : Verify that the file is correctly uploaded by listing the files in HDFS.

4. `docker exec namenode hdfs dfs -cat /user/hdfs/input/car_prices.csv | head -10` : Displays the first 10 lines of the CSV file:

```
 year,make,model,trim,body,transmission,vin,state,condition,odometer,
color,interior,seller,mmr,sellingprice,saledate
 2015,Kia,Sorento,LX,SUV,automatic,5xyktca69fg566472,ca,5,16639,white,
black,kia motors america  inc,20500,21500,Tue Dec 16 2014 12:30:00 GMT
-0800 (PST)
 2015,Kia,Sorento,LX,SUV,automatic,5xyktca69fg561319,ca,5,9393,white,
beige,kia motors america  inc,20800,21500,Tue Dec 16 2014 12:30:00 GMT
-0800 (PST)
 2014,BMW,3 Series,328i SULEV,Sedan,automatic,wba3c1c51ek116351,ca
,45,1331,gray,black,financial services remarketing (lease),31900,30000,
Thu Jan 15 2015 04:30:00 GMT-0800 (PST)
 2015,Volvo,S60,T5,Sedan,automatic,yv1612tb4f1310987,ca,41,14282,white,
black,volvo na rep/world omni,27500,27750,Thu Jan 29 2015 04:30:00 GMT
-0800 (PST)
 ....
 ....
```

5. create java classes: Driver, Mapper-Reducer

6. Create `logback.xml` in `src/resources`. I used it to configure logging in a Hadoop MapReduce application and to debug the code.

7. Modify `pom.xml` to include `logback.xml`. This ensures proper logging configuration in the Hadoop MapReduce application, improving debugging and monitoring. The following dependency must be added to `pom.xml` to support Logback:

```
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
```

```
        <version>1.2.3</version>
    </dependency>
```

8. `mvn clean package` : each time a change in the code occurs
9. `docker exec -it namenode hdfs dfs -rm -r /user/hdfs/output/` : ensure that the output is empty before writing to it
10. `docker cp target/hadoop-map-reduce-examples-1.0-SNAPSHOT-jar-with-dependencies.jar namenode:/hadoop-map-reduce-examples-1.0-SNAPSHOT-jar-with-dependencies.jar`
11. `docker exec namenode hadoop jar /hadoop-map-reduce-examples-1.0-SNAPSHOT-jar-with-dependencies.jar gr.aueb.panagiotisl.mapreduce.wordcount.Driver /user/hdfs/input/car_prices.csv /user/hdfs/output/`
12. `docker exec namenode hdfs dfs -cat /user/hdfs/output/part-r-00000/` : to see the output
13. Below we can see part of the output, where we have defined in the reducer to return this result:

```
String result = carWithMaxDifference + ": "
                + String.format("%.1f", maxDifference)
                + ", avg: " + String.format("%.1f",
   averageDifference);
```

```
......
zimbrick eastside:2014-12:zimbrick eastside     1525.0: 1525.0,
   avg: 591.7
zimbrick eastside:2015-01:zimbrick eastside     1850.0: 1850.0,
   avg: 440.8
zimbrick eastside:2015-02:zimbrick eastside     2000.0: 2000.0,
   avg: 503.3
zimbrick eastside:2015-03:zimbrick eastside     1850.0: 1850.0,
   avg: 468.7
......
```