CS580-MiniProject 2

Name- Parneet Lnu

BNumber- B00816285

Email- plnu2@binghamton.edu

Q1. Can you figure out what are the main steps do we need to run a hadoop mapreduce task (i.e., wordcount here)?

Ans. In this case we are using a docker image to run hadoop mapreduce task.

- We pull the docker image.
- Then clone the Github repository.
- Then we create a Hadoop for all the Hadoop container to connect.
- Then we enter the hadoop-cluster-docker directory.
- We start Hadoop container clusters using the command sudo ./start-container.sh. The output is as follows:

  ```
  start hadoop-master container...
  start hadoop-slave1 container...
  start hadoop-slave2 container...
  ```
- After we have started the container and are in the root directory of Hadoop master container, we start the Hadoop clusters using the command ./start-hadoop.sh.
- Finally, we run the wordcount task with the command ./run-wordcount.sh.

Q2. What does this command mean — "hdfs dfs -put ./input/* input"? (Hint, HDFS is Hadoop's distributed file system.

Ans. The put command is used to copy single src, or multiple srcs from local file system to the destination file system, according to Apache Hadoop documentation. Put can also be used to read input from stdin and writes to destination file system if the source is set to "-". Here copying will fail if the file already exists unless -f flag is given.

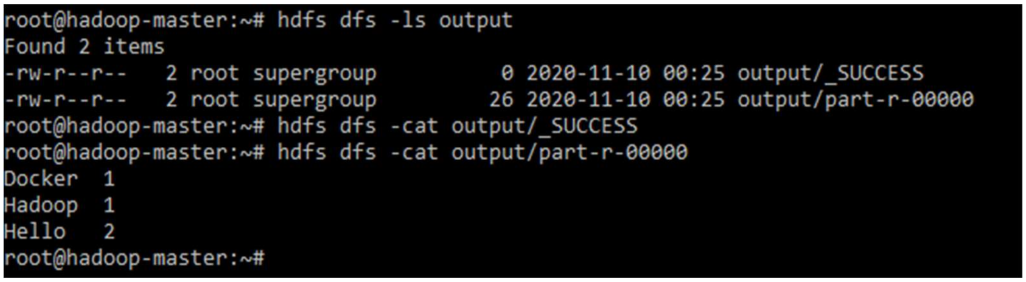
Q3. How many mappers and reducers are launched for executing the above wordcount program?

Ans. In the current wordcount program number of mappers used is 2 and number of reducers used is 1.

Q4. How much time do mappers and reducers spend for the above tasks, separately?

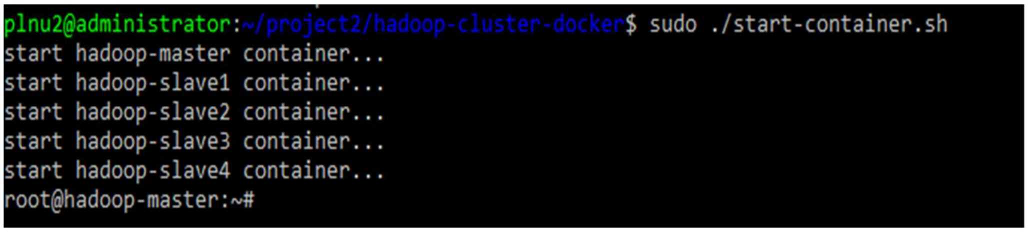
Ans. The time spent by map tasks is 10575 ms and the time spent by reduce tasks is 4538 ms.

Q5. After execution, what are the files in the output folder in HDFS, and what content do they contain?
Ans. Two files are created in the output folder namely _SUCCESS and part-r-00000. The contents of the files are shown in the screenshot below:



```
root@hadoop-master:~# hdfs dfs -ls output
Found 2 items
-rw-r--r--   2 root supergroup          0 2020-11-10 00:25 output/_SUCCESS
-rw-r--r--   2 root supergroup         26 2020-11-10 00:25 output/part-r-00000
root@hadoop-master:~# hdfs dfs -cat output/_SUCCESS
root@hadoop-master:~# hdfs dfs -cat output/part-r-00000
Docker  1
Hadoop  1
Hello   2
root@hadoop-master:~#
```

Q6. How many master and slave containers do you launch separately this time?
Ans. This time 1 master and 4 slave containers are launched.



```
plnu2@administrator:~/project2/hadoop-cluster-docker$ sudo ./start-container.sh
start hadoop-master container...
start hadoop-slave1 container...
start hadoop-slave2 container...
start hadoop-slave3 container...
start hadoop-slave4 container...
root@hadoop-master:~#
```

Q7. Please figure out what a master container/node and a slave container/node are used for.
Ans. **Master container** in Hadoop HDFS is the NameNode.

- It maintains and manages the slave containers/nodes.
- It records the metadata of all files stored in the cluster.
- It receives a heartbeat and block report from the slave nodes to ensure they are alive.
- It also takes care of the replication factor of all blocks.
- The slave node failure is also taken care by the NameNode.

Slave container in Hadoop HDFS is the DataNode.

- It is commodity hardware, not of high quality or availability.
- It is a block server that stores the data in the local file ext3 or ext4.
- It stores the actual data.
- It also performs the low level read and write from the file systems's clients.

Q8. How many mappers and reducers are launched for executing the above wordcount program?
Ans. In the current wordcount program number of mappers used is 3 and number of reducers used is 1.

Q9. How much time do mappers and reducers spend for the above tasks, separately?
Ans. The time spent by map tasks is 26767 ms and the time spent by reduce tasks is 4675 ms.

Q10. What are the two most frequently occurring words, and how many times do they occur?
Ans. The most frequently occurring words are “The”- occurring 42 times and “of”- occurring 27 times.

Q11. Please describe the basic steps in the map function of WordCount.java.
Ans. The basic steps are as follows:

- An object itr of StringTokenizer is created.
- A while loop is run until the itr has no more token i.e. till the text file has no more words.
- With each loop i.e. for each token a new key-value pair will be generated. So, each word will be a key and value will be 1.
- These key-value pairs are then written.

Q12. Please describe the basic steps in the reduce function of WordCount.java.
Ans. The basic steps are as follows:

- We loop through the keys with the object val of class IntWritable.
- For each key the sum is done on the number of entries in the value.
- This result i.e. the sum is stored for each key.

Q13. How many mappers and reducers are launched for executing the above wordcount program?
Ans. In the current wordcount program number of mappers used is 2 and number of reducers used is 1.

Q14. How much time do mappers and reducers spend for the above tasks, separately?
Ans. The time spent by map tasks is 17999 ms and the time spent by reduce tasks is 5782 ms.

Task 5:
I have taken a sample data in sample.txt which represents the scores gained by each student in their tests. Each line in sample .text shows a student ID in first column and scores in the remaining 13 columns. The Map-Reduce calculates the sum of all the test scores and gives the output as Student Id-sum of all test scores as key-value pair.

The following steps can be followed to compile and execute the ProcessUnits.java program:

- Create a directory input in root directory of Hadoop container using mkdir input.
- In this directory create a file sample.txt.
- Create ProcessUnits.java in the root directory of Hadoop container.
- Create directory to store compiled java classes.
- Download **Hadoop-core-1.2.1.jar from** http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core/1.2.1 in the root directory.
- Use the following commands to compile the ProcessUnits.java program:

```
javac -classpath hadoop-core-1.2.1.jar -d units         ProcessUnits.java
jar -cvf units.jar -C units/ .
```

- Use the following command to create an input directory in HDFS:
  ```
  hadoop fs -mkdir -p input_dir
  ```
- Use the following command to put input files to HDFS:
  ```
  hdfs dfs -put ./input/* input_dir
  ```
- Use the following command to execute the application:
  ```
  hadoop jar units.jar hadoop.ProcessUnits input_dir output_dir
  ```
- Use the following command to see the output in Part-00000 file:
  ```
  hadoop fs -cat output_dir/part-00000
  ```

References:

https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm