IST769 Homework Submission Template

Basic Information

Your Name: Mark Roberts  
Your SUID: 598273961  
Your Email: mrober04@syr.edu  
Date Due: August 10, 2021   
Homework #: 5

Your Answers:

1. Upload all the documents in **datasets/text** into a folder called **text** in HDFS. What HDFS command must you run to verify the files are there after they are uploaded? Your answer should consist of the command you typed to complete the task.

|  |
| --- |
| CODE |
| hdfs dfs -mkdir text  cd datasets/text  hdfs dfs -put \* text  hdfs dfs -ls text  ls  hdfs dfs -ls |
| SCREENSHOT/OUTPUT |
|  |

1. In this part you will upload the **clickstream** dataset to HDFS. Specifically, create a **clickstream** folder in HDFS, then create a **logs** and **iplookup** folder inside the clickstream folder. Upload all of the **\*.log** files from the **datasets/clickstream** local folder into **clickstream/logs** in HDFS. Upload the **ip\_lookup.csv** file from the same folder into **clickstream/iplookup** on HDFS. Verify the files are there. Your answer should consist of the commands you typed to complete the task.

|  |
| --- |
| CODE |
| [cloudera@quickstart ~]$ hdfs dfs -mkdir -p clickstream/logs  [cloudera@quickstart ~]$ hdfs dfs -mkdir -p clickstream/iplookup  [cloudera@quickstart ~]$ hdfs dfs -put datasets/clickstream/\*.log clickstream/logs  [cloudera@quickstart ~]$ hdfs dfs -put datasets/clickstream/\*.csv clickstream/iplookup  [cloudera@quickstart ~]$ hdfs dfs -ls clickstream/logs |
| SCREENSHOT/OUTPUT |
|  |

1. Use the MapReduce examples:  
   export MREX=/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar   
   to perform a wordcount on the 2016 State of the Union address, saving the output to the HDFS folder **sotu2016**. Write down the commands to complete the task. How many times does the word **are** appear in the 2016 State of the Union address? Describe a process which could be done to make the wordcount more useful?

|  |
| --- |
| CODE |
| [cloudera@quickstart ~]$ export MREX=/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar  [cloudera@quickstart ~]$ yarn jar $MREX  [cloudera@quickstart ~]$ yarn jar $MREX wordcount text/2016-state-of-the-union.txt sotu2016/sotu-wordcount  [cloudera@quickstart ~]$ hdfs dfs -ls sotu2016/sotu-wordcount  [cloudera@quickstart ~]$ hdfs dfs -cat sotu2016/sotu-wordcount/part-r-00000 |
| SCREENSHOT/OUTPUT |
| 21/08/09 18:40:36 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032  21/08/09 18:40:37 INFO input.FileInputFormat: Total input paths to process : 1  21/08/09 18:40:37 INFO mapreduce.JobSubmitter: number of splits:1  21/08/09 18:40:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1628525630386\_0001  21/08/09 18:40:38 INFO impl.YarnClientImpl: Submitted application application\_1628525630386\_0001  21/08/09 18:40:38 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application\_1628525630386\_0001/  21/08/09 18:40:38 INFO mapreduce.Job: Running job: job\_1628525630386\_0001  21/08/09 18:40:46 INFO mapreduce.Job: Job job\_1628525630386\_0001 running in uber mode : false  21/08/09 18:40:46 INFO mapreduce.Job: map 0% reduce 0%  21/08/09 18:40:52 INFO mapreduce.Job: map 100% reduce 0%  21/08/09 18:40:59 INFO mapreduce.Job: map 100% reduce 100%  21/08/09 18:41:00 INFO mapreduce.Job: Job job\_1628525630386\_0001 completed successfully  21/08/09 18:41:00 INFO mapreduce.Job: Counters: 49  File System Counters  FILE: Number of bytes read=27301  FILE: Number of bytes written=282261  FILE: Number of read operations=0  FILE: Number of large read operations=0  FILE: Number of write operations=0  HDFS: Number of bytes read=35874  HDFS: Number of bytes written=19316  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)=4104  Total time spent by all reduces in occupied slots (ms)=3974  Total time spent by all map tasks (ms)=4104  Total time spent by all reduce tasks (ms)=3974  Total vcore-seconds taken by all map tasks=4104  Total vcore-seconds taken by all reduce tasks=3974  Total megabyte-seconds taken by all map tasks=4202496  Total megabyte-seconds taken by all reduce tasks=4069376  Map-Reduce Framework  Map input records=173  Map output records=6152  Map output bytes=60075  Map output materialized bytes=27301  Input split bytes=143  Combine input records=6152  Combine output records=2018  Reduce input groups=2018  Reduce shuffle bytes=27301  Reduce input records=2018  Reduce output records=2018  Spilled Records=4036  Shuffled Maps =1  Failed Shuffles=0  Merged Map outputs=1  GC time elapsed (ms)=62  CPU time spent (ms)=2110  Physical memory (bytes) snapshot=478142464  Virtual memory (bytes) snapshot=2736619520  Total committed heap usage (bytes)=369098752  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=35731  File Output Format Counters  Bytes Written=19316 |

1. Type the following command to import the **fudgemart\_v3** database into the local **mysql** instance on the Hadoop client:  
   mysql -u root -p < ~/datasets/fudgemart/mysql.sql   
   The password is **cloudera**. Write down the commands you used to complete these tasks:  
   Use the **sqoop** utility to verify there are tables in the database by listing them from the **fudgemart\_v3** database. Next write a sqoop command to import Fudgemart products in the ‘Clothing’ department into a HDFS folder **/user/cloudera/fudgemart-clothing**

|  |
| --- |
| CODE |
| [cloudera@quickstart ~]$ mysql -u root -p < ~/datasets/fudgemart/mysql.sql  Enter password:  [cloudera@quickstart ~]$ sqoop list-databases --connect jdbc:mysql://cloudera --username=root --password=cloudera  [cloudera@quickstart ~]$ hdfs dfs -mkdir /user/cloudera/fudgemart-clothing  [cloudera@quickstart ~]$ sqoop eval --connect jdbc:mysql://cloudera/fudgemart\_v3 --username=root --password=cloudera --query="SELECT \* FROM fudgemart\_products WHERE product\_department = 'Clothing'" > clothes.txt  [cloudera@quickstart ~]$ cat clothes.txt  [cloudera@quickstart ~]$ hdfs dfs -put clothes.txt /user/cloudera/fudgemart-clothing  [cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/fudgemart-clothing  Found 1 items  -rw-r--r-- 1 cloudera cloudera 3295 2021-08-09 19:33 /user/cloudera/fudgemart-clothing/clothes.txt  [cloudera@quickstart ~]$ rm clothes.txt  rm: remove regular file `clothes.txt'? y |
| SCREENSHOT/OUTPUT |
|  |

1. Let’s import HDFS data into MySQL. Record each command you type as your solution:
2. Load **datasets/tweets/tweets.psv** into the HDFS folder **tweets**
3. Login to MySQL: mysql -u root -p The password is **cloudera**. Create a database **twitter**
4. Create a table called **tweets** inside the database **twitter** the table should have columns for id, timestamp, date time, username, and tweet\_text.
5. Export the data from HDFS into the MySQL table.  
   **TIPS:** If your SQOOP job fails it is likely due to the table constraints such as selecting a data type too small for the imported data. Try to insert a row in the table using a sample from the HDFS data. This will help you to ensure your chosen data types will work.

|  |
| --- |
| CODE |
| [cloudera@quickstart ~]$ hdfs dfs -mkdir tweets  [cloudera@quickstart ~]$ hdfs dfs -put datasets/tweets/tweets.psv tweets  [cloudera@quickstart ~]$ hdfs dfs -ls tweets  [cloudera@quickstart ~]$ mysql -u root -p  mysql> CREATE DATABASE twitter;  mysql> USE twitter;  mysql> CREATE TABLE tweets (id varchar(50), timestamp varchar(50), date\_time varchar (50), user\_name varchar(50), tweet\_text varchar(300));  mysql> SELECT \* FROM tweets;  Empty set (0.01 sec)  mysql> \q  Bye  [cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://cloudera/twitter --username=root --password=cloudera -  -table=tweets --export-dir=tweets --input-fields-terminated-by="|" |
| SCREENSHOT/OUTPUT |
| Graphical user interface, text  Description automatically generated with medium confidence |