ROLL NO: 210701268

EXP 4: Create UDF in PIG

Aim: To create UDF in PIG using Hadoop.

INSTALLATION OF PIG:

Step-by-step installation of Apache Pig on Hadoop cluster on Ubuntu Pre-requisite:

- Ubuntu 16.04 or higher version running (I have installed Ubuntu on Oracle VM (Virtual Machine) VirtualBox),
- Run Hadoop on ubuntu (I have installed Hadoop 3.2.1 on Ubuntu 16.04). You may refer to my blog "How to install Hadoop installation" click here for Hadoop installation).

Pig installation steps

1) **Step 1:** Login into Ubuntu
2) **Step 2**: Go to https://pig.apache.org/releases.html and copy the path of the latest version of pig that you want to install. Run the following comment to download Apache Pig in Ubuntu:

   $ wget https://dlcdn.apache.org/pig/pig-0.16.0/pig-0.16.0.tar.gz

3) **Step 3**: To untar pig-0.16.0.tar.gz file run the following command:

   $ tar xvzf pig-0.16.0.tar.gz

4) **Step 4:** To create a pig folder and move pig-0.16.0 to the pig folder, execute the following command:

   $ sudo mv /home/hdoop/pig-0.16.0 /home/hdoop/pig

5) **Step 5:** Now open the .bashrc file to edit the path and variables/settings for pig. Run the following command:

   $ sudo nano .bashrc

   Add the below given to .bashrc file at the end and save the file.

   #PIG settingsexport PIG_HOME=/home/hdoop/pigexport PATH=$PATH:$PIG_HOME/binexport PIG_CLASSPATH=$PIG_HOME/conf:$HADOOP_INSTALL/etc/hadoop/export PIG_CONF_DIR=$PIG_HOME/confexport JAVA_HOME=/usr/lib/jvm/java-8openjdkamd64export PIG_CLASSPATH=$PIG_CONF_DIR:$PATH#PIG setting ends

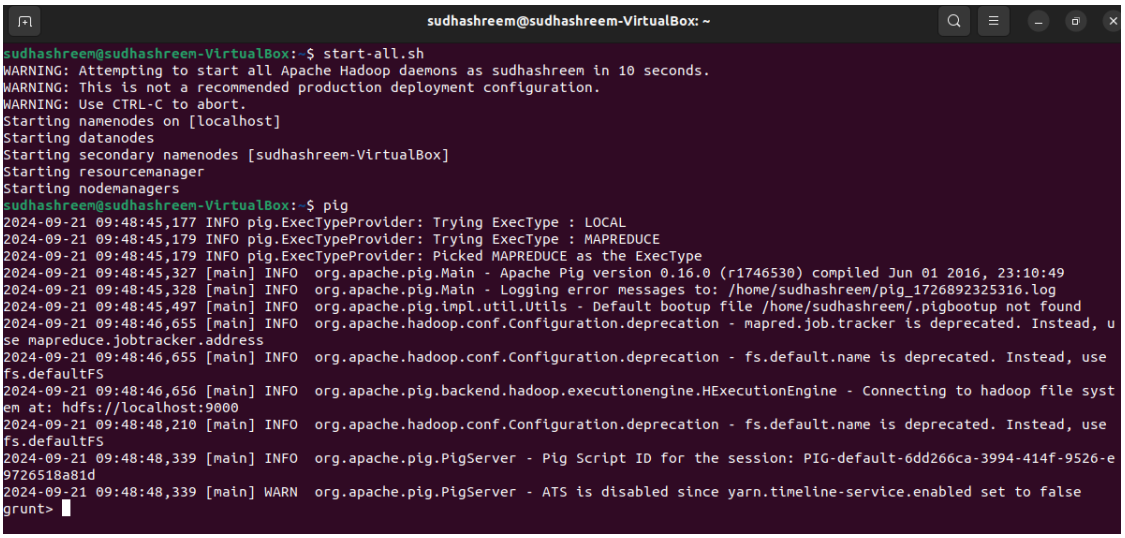
6) **Step 6:** Run the following command to make the changes effective in the .bashrc file:

   $ source .bashrc

7) **Step 7:** To start all Hadoop daemons, navigate to the hadoop-3.2.1/sbin folder and run the following commands:

   $ ./start-dfs.sh$ ./start-yarn$ jps

8) **Step 8:** Now you can launch pig by executing the following command:
   $ pig



9) **Step 9:** Now you are in pig and can perform your desired tasks on pig. You can come out of the pig by the quit command:
   $ quit;

CREATE USER DEFINED FUNCTION(UDF)

Aim : To create User Define Function in Apache Pig and execute it on map reduce.

Procedure:

1) **Create a sample text file**

   hadoop@Ubuntu:~/Documents$ nano sample.txt

   Paste the below content to sample.txt

   1,John

   2,Jane

   3,Joe

4,Emma

hadoop@Ubuntu:~/Documents$ hadoop fs -put sample.txt /home/hadoop/piginput/

---

2) **Create PIG File** hadoop@Ubuntu:~/Documents

   $ nano demo_pig.pig

   paste the below the content to demo_pig.pig

   -- Load the data from HDFS data = LOAD '/home/hadoop/piginput/sample.txt'

   USING PigStorage(',') AS (id:int>

   -- Dump the data to check if it was loaded correctly

   DUMP data;

3) **Run the created file** hadoop@Ubuntu:~/Documents$ pig demo_pig.pig

4) **Create udf file an save as uppercase_udf.py**

nano uppercase_udf.py :

def uppercase(text): return text.upper()

if __name__ == "__main__":

import sys for line in

sys.stdin:

```
        line = line.strip() result
        = uppercase(line)
        print(result)
```

5) Create the udf folder on Hadoop -mkdir/home/Hadoop/udfs and put the uppercase_udf.py inside that folder

6) Create another file udf_example.py :

$ nano udf_example.pig

Put the below content on udf_example.pig

-- Register the Python UDF script

REGISTER 'hdfs:///home/hadoop/udfs/uppercase_udf.py' USING jython AS udf;

-- Load some data data = LOAD 'hdfs:///home/hadoop/sample.txt'

AS (text:chararray);

-- Use the Python UDF uppercased_data = FOREACH data GENERATE

udf.uppercase(text) AS uppercase_text;

-- Store the result

STORE uppercased_data INTO 'hdfs:///home/hadoop/pig_output_data';

7) Place sample.txt file on hadoop hadoop@Ubuntu:~/Documents$

hadoop fs -put sample.txt /home/hadoop/

8) To Run the pig file :

$ hdfs dfs -cat /exp4/output/part-m-00000

$ pig demo_pig.pig

```
sudhashreem@sudhashreem-VirtualBox: ~/DA/exp4
 6 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:48,470 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 7 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:49,478 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 8 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:50,480 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 9 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:50,616 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicati
onStatus=SUCCEEDED. Redirecting to job history server
2024-09-20 19:15:51,647 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 0 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:52,656 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 1 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:53,658 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 2 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:54,660 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 3 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:55,663 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 4 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:56,666 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 5 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:57,668 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 6 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:58,671 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 7 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:15:59,673 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 8 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:16:00,674 [main] INFO  org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried
 9 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2024-09-20 19:16:00,800 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Unable to retr
ieve job to compute warning aggregation.
2024-09-20 19:16:00,804 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2024-09-20 19:16:01,059 [main] INFO  org.apache.pig.Main - Pig script completed in 4 minutes, 6 seconds and 308 milliseconds (246308
 ms)
sudhashreem@sudhashreem-VirtualBox:~/DA/exp4$ hdfs dfs -cat /exp4/output/part-m-00000
1,JOHN
2,JANE
3,JOE
4,EMMA
sudhashreem@sudhashreem-VirtualBox:~/DA/exp4$
```

Result: Thus udf file is successfully created and executed using pig and map reduce.