EX : 04(1) **REG.NO:210701290**

CREATE USER DEFINED FUNCTION(UDF)

Aim :

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

Procedure:

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/

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;
```

Run the above file

hadoop@Ubuntu:~/Documents$ pig demo_pig.pig

2024-08-07 12:13:08,791 [main] INFO

org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil

- Total input paths to process : 1

(1,John)

(2,Jane)

(3,Joe)

(4,Emma)

Create udf file an save as uppercase_udf.py

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)
```

Create the udfs folder on hadoop

hadoop@Ubuntu:~/Documents$ hadoop fs -mkdir /home/hadoop/udfs

put the upppercase_udf.py in to the abv folder

hadoop@Ubuntu:~/Documents$ hdfs dfs -put uppercase_udf.py /home/hadoop/udfs/

---

hadoop@Ubuntu:~/Documents$ nano udf_example.pig

copy and paste 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';
```

---

place sample.txt file on hadoop

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

To Run the pig file

```
hadoop@Ubuntu:~/Documents$ pig -f udf_example.pig
```

finally u get

Success!

Job Stats (time in seconds):

JobId Maps Reduces MaxMapTimeMinMapTime AvgMapTime MedianMapTime

MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime

Alias Feature Outputs

job_local1786848041_0001 1 0 n/a n/a n/a n/a 00 0 0

data,uppercased_data MAP_ONLY hdfs:///home/hadoop/pig_output_data,

Input(s):

Successfully read 4 records (42778068 bytes) from: "hdfs:///home/hadoop/sample.txt"

Output(s):

Successfully stored 4 records (42777870 bytes) in: "hdfs:///home/hadoop/pig_output_data"

Counters:

Total records written : 4

Total bytes written : 42777870

Spillable Memory Manager spill count : 0

Total bags proactively spilled: 0

Total records proactively spilled: 0

Job DAG:

job_local1786848041_0001

2024-08-07 13:33:04,631 [main] WARN

org.apache.hadoop.metrics2.impl.MetricsSystemImpl -

JobTracker metrics system already initialized!

2024-08-07 13:33:04,639 [main] WARN

org.apache.hadoop.metrics2.impl.MetricsSystemImpl -

JobTracker metrics system already initialized!

2024-08-07 13:33:04,644 [main] WARN

org.apache.hadoop.metrics2.impl.MetricsSystemImpl -

JobTracker metrics system already initialized!

2024-08-07 13:33:04,667 [main] INFO

org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher -

Success!

Note :

If any error check jython package is installed and check the path specified on the above

steps are give correctly

To check the output file is created

hadoop@Ubuntu:~/Documents$ hdfs dfs -ls /home/hadoop/pig_output_data

Found 2 items

If you need to examine the files in the output folder, use:

To view the output

hadoop@Ubuntu:~/Documents$ hdfs dfs -cat /home/hadoop/pig_output_data/part-m

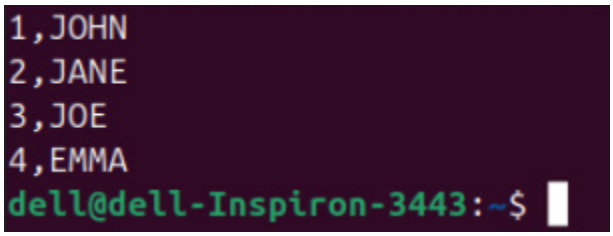
00000

1,JOHN

2,JANE

3,JOE

4,EMMA



Result:

Thus the program is executed successfully