

Exp.No: 4

**Create User Defined Function (UDF) in Apache Pig and execute it in
MapReduce**

AIM:

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

PROCEDURE:

Step-1: 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 command to download Apache Pig in Ubuntu:

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

Step-2: To untar pig-0.16.0.tar.gz file run the following command:

```
tar xvzf pig-0.16.0.tar.gz
```

Step 3: To create a pig folder and move pig-0.16.0 to the pig folder, execute the following command:

```
sudo mv /home/hadoop/pig-0.16.0 /home/hadoop/pig
```

Step 4: 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 settings

```
export PIG_HOME=/home/hadoop/pig
```

```
export PATH=$PATH:$PIG_HOME/bin
```

```
export PIG_CLASSPATH=$PIG_HOME/conf:$HADOOP_INSTALL/etc/hadoop/
```

```
export PIG_CONF_DIR=$PIG_HOME/conf
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdkamd64
```

```
export PIG_CLASSPATH=$PIG_CONF_DIR:$PATH
```

#PIG setting ends

Step 5: Run the following command to make the changes effective in the .bashrc file:

```
source .bashrc
```

Step 6: To start all Hadoop daemons, navigate to the hadoop-3.2.1/sbin folder and run the following commands:

```
./start-dfs.sh
```

```
./start-yarn.sh
```

Step 7: Create a sample text file

```
nano sample.txt
```

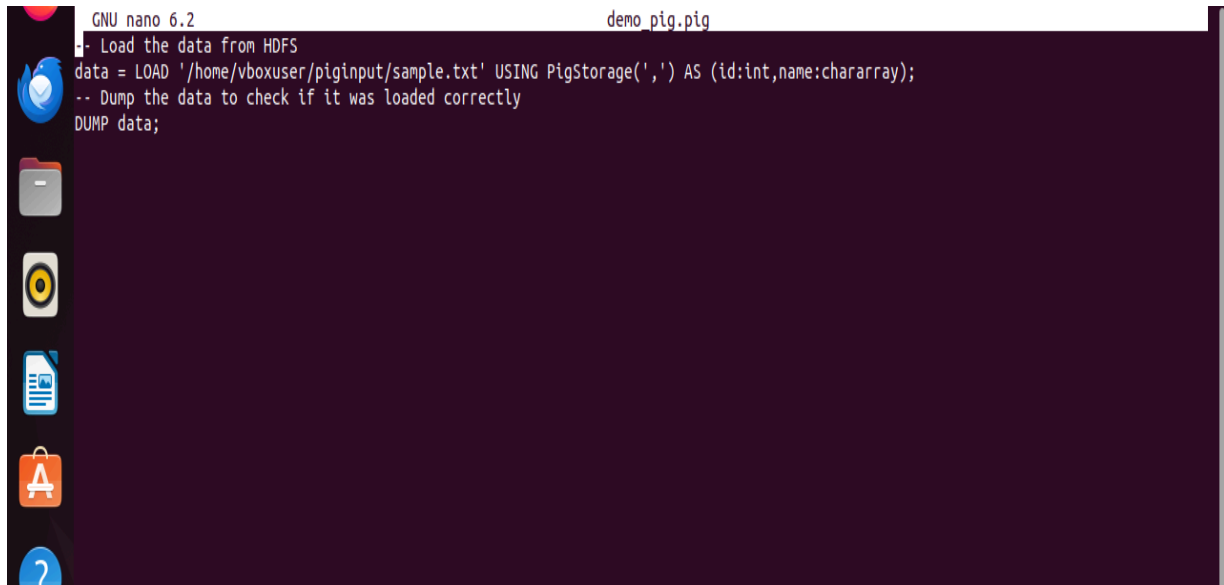


Step 8: Add the text file to the Hadoop environment.

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

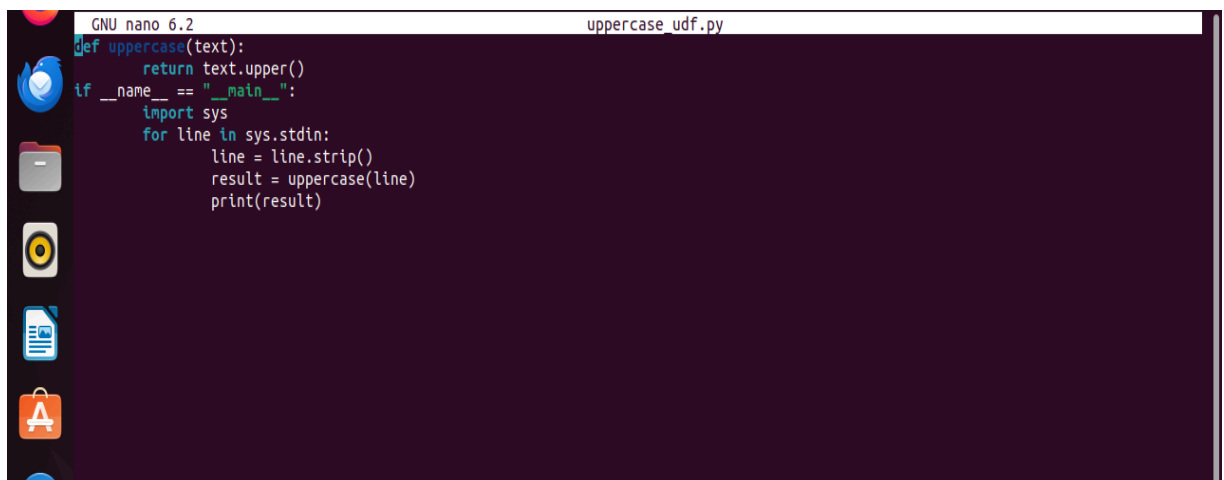
Step 9: Create PIG File

```
nano demo_pig.pig
```



```
GNU nano 6.2                                demo_pig.pig
-- Load the data from HDFS
data = LOAD '/home/vboxuser/piginput/sample.txt' USING PigStorage(',') AS (id:int,name:chararray);
-- Dump the data to check if it was loaded correctly
DUMP data;
```

Step 10: Create udf file and save as uppercase_udf.py



```
GNU nano 6.2                                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)
```

Step 11: Create the udfs folder on hadoop

```
hadoop fs -mkdir /home/hadoop/udfs
```

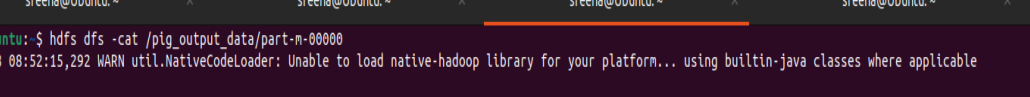
Step 12: Put the uppercase_udf.py in to the above folder

```
hdfs dfs -put uppercase_udf.py /home/hadoop/udfs/
```

Step 13: Create a file named udf_example.pig

```
nano udf_example.pig
```

```
hdfs dfs -cat /home/hadoop/pig_output_data/part-m-00000
```



A terminal window with a dark purple background. The prompt is `sreena@Ubuntu: ~`. The command `hdfs dfs -cat /pig_output_data/part-m-00000` has been executed. The output shows a timestamp `2024-09-18 08:52:15,292 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable` followed by a list of names: `1,JOHN`, `2,JANE`, `3,JOE`, and `4,EMMA`. The prompt is now `sreena@Ubuntu: ~$`. On the left side of the terminal, there are icons for a file manager, a terminal, and a document.

```
sreena@Ubuntu: ~$ hdfs dfs -cat /pig_output_data/part-m-00000
2024-09-18 08:52:15,292 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
1,JOHN
2,JANE
3,JOE
4,EMMA
sreena@Ubuntu: ~$
```

RESULT: Thus the program is executed successfully and output is verified.