Expt-4:

Create UDF (User Defined Functions) in Apache Pig and execute it in MapReduce / HDFS mode

AIM:

To create UDF in Apache Pig and execute it in MapReduce/HDFS mode.

Procedure:

Step 1: Install and Configure Apache Pig

1. **Download Apache Pig**:

   Download the latest version of Pig from the official website: wget https://dlcdn.apache.org/pig/pig-0.16.0/pig-0.16.0.tar.gz

2. **Extract Pig**:

   tar xvzf pig-0.16.0.tar.gz

3. **Move Pig Directory**:

   Move the extracted Pig files to a dedicated folder: sudo mv pig-0.16.0 /usr/local/pig

4. **Set Environment Variables**:

   Edit the .bashrc file to set up Pig environment variables:

   nano ~/.bashrc

   Append the following lines:

   export PIG_HOME=/usr/local/pig export PATH=$PATH:$PIG_HOME/bin export PIG_CLASSPATH=$HADOOP_HOME/conf

   Apply the changes: source ~/.bashrc

5. **Verify Pig Installation**:

Run the following command to verify if Pig has been installed correctly:

```
pig -version
```

Step 2: Create Sample Data for the Pig Job

1. **Create a Sample Data File**: Create a sample text file (sample.txt) with some dummy data:

   ```
   nano sample.txt
   ```

   Add the following content:

   ```
   1,John
   2,Jane
   3,Joe
   4,Emma
   ```

2. **Upload the Data File to HDFS**: Upload the sample file to Hadoop's distributed file system (HDFS):

   ```
   hdfs dfs -mkdir /piginput hdfs
   dfs -put sample.txt /piginput
   ```

Step 3: Write Pig Script for the UDF

1. **Create the Pig Script**:

   Create a new Pig script (demo_pig.pig):

   ```
   nano demo_pig.pig
   ```

   Write the following code in the script to load and display the data:

   ```
   pig
   -- Load data from HDFS
   data = LOAD '/piginput/sample.txt' USING PigStorage(',') AS (id:int,
   name:chararray);
   -- Display the loaded data
   ```

```
DUMP data;
```

Step 4: Write the UDF in Python

1. **Create the Python UDF**:

   Create a Python file (uppercase_udf.py) to convert text to uppercase:

   ```
   nano uppercase_udf.py  def
   uppercase(text):     return
   text.upper()

   if __name__ == "__main__":
      import sys     for line in
   sys.stdin:         line = line.strip()
   print(uppercase(line))
   ```

2. **Upload the Python UDF to HDFS**:

   Upload the UDF to HDFS:

   ```
   hdfs dfs -mkdir /udfs hdfs dfs -put
   uppercase_udf.py /udfs
   ```

Step 5: Update Pig Script to Use UDF

1. **Modify the Pig Script to Include UDF**:

   Edit the demo_pig.pig script to register the UDF and process the data:

   ```
   nano demo_pig.pig
   ```

   Modify the script as follows:

   ```
   pig
   ```

   -- Register the Python UDF script

   ```
   REGISTER '/udfs/uppercase_udf.py' USING jython AS myudf;
   ```

-- Load data from HDFS

```
data = LOAD '/piginput/sample.txt' USING PigStorage(',') AS (id:int,
name:chararray)
```

-- Apply UDF to convert names to uppercase

```
uppercased_data = FOREACH data GENERATE
myudf.uppercase(name);
```

-- Display the transformed data

```
DUMP uppercased_data;
```

Step 6: Run the Pig Script

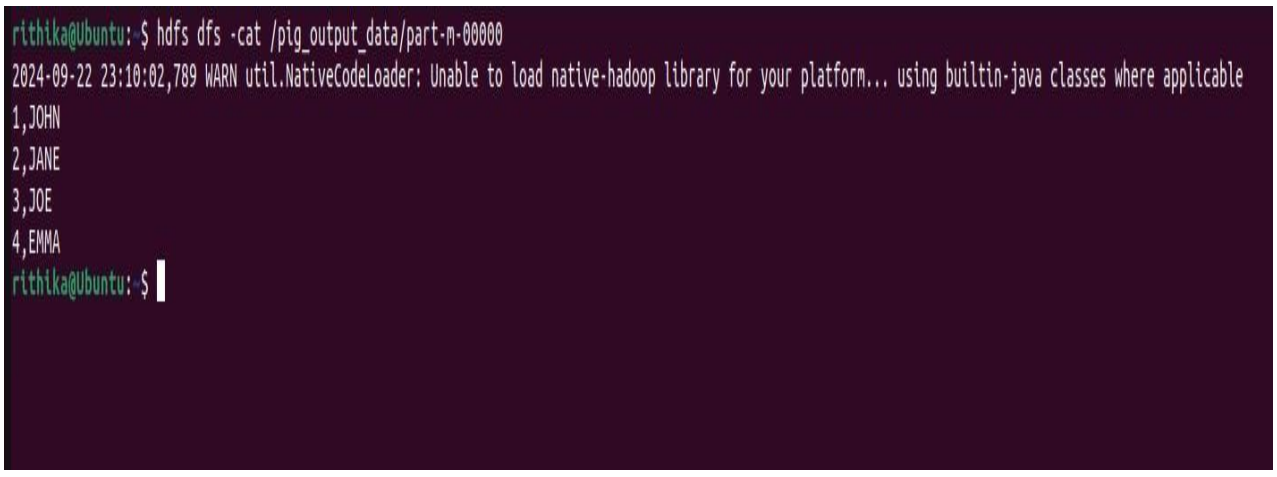
1. **Run the Pig Script**:

   Run the Pig script using the following command: pig

   -x mapreduce demo_pig.pig

2. **View Output** hdfs dfs -cat /pigoutput/part-m-00000

OUTPUT:



RESULT:

Thus, UDF in Apache Pig has been created and executed in MapReduce/HDFS mode successfully.