

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:

Pig Download and installation:

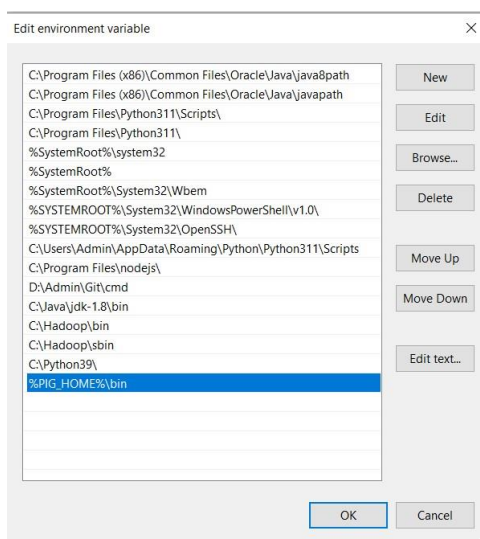
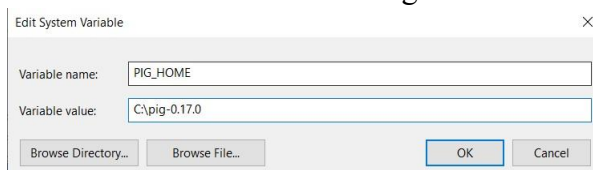
1. Download Pig:

Download Pig from “<https://downloads.apache.org/pig/pig-0.17.0/>”

Index of /pig/pig-0.17.0

Name	Last modified	Size	Description
Parent Directory	-	-	-
README.txt	2017-06-16 18:10	1.4K	
RELEASE_NOTES.txt	2017-06-16 18:10	1.9K	
pig-0.17.0-src.tar.gz	2017-06-16 18:11	15M	
pig-0.17.0-src.tar.gz.asc	2017-06-16 18:11	488	
pig-0.17.0-src.tar.gz.md5	2017-06-16 18:11	56	
pig-0.17.0.tar.gz	2017-06-16 18:10	220K	
pig-0.17.0.tar.gz.asc	2017-06-16 18:11	488	
pig-0.17.0.tar.gz.md5	2017-06-16 18:11	52	

2. Add the environment variable for Pig:



3. Go to C:\pig-0.16.0\bin and open pig (Windows Command Script)

```
set HADOOP_BIN_PATH=%HADOOP_HOME%\libexec
```

4. Open Windows Powershell and type “pig -x local” and check whether pig grunt appears.

Pig is successfully installed.

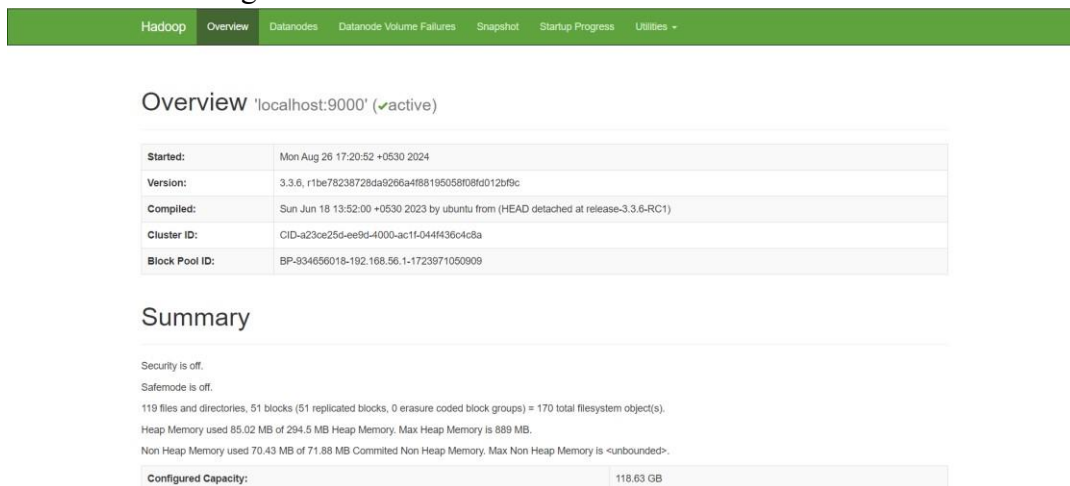
Create UDF:

1. Start Hadoop services:

Open command prompt as an administrator

```
start-dfs.cmd start-  
yarn.cmd
```

2. Open the browser and go to the URL “localhost:9870”



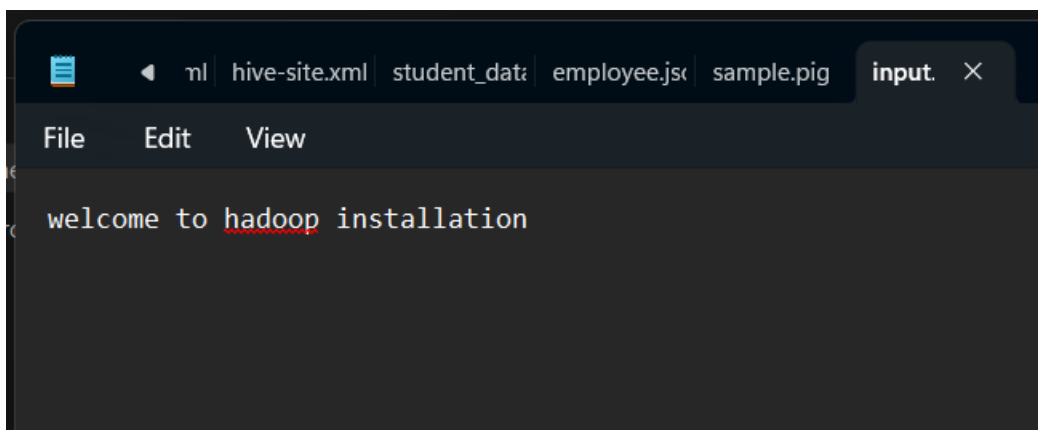
Overview 'localhost:9000' (✓active)	
Started:	Mon Aug 26 17:20:52 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-a23ce25d-ee9d-4000-ac11-044f436c4c8a
Block Pool ID:	BP-934656018-192.168.56.1-1723971050909

Summary

Security is off.
Safemode is off.
119 files and directories, 51 blocks (51 replicated blocks, 0 erasure coded block groups) = 170 total filesystem object(s).
Heap Memory used 85.02 MB of 294.5 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 70.43 MB of 71.88 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	118.63 GB
----------------------	-----------

3. Create a text file “input.txt”:



4. Create a Python file “uppercase_udf.py”:

```
uppercase_udf - Notepad
File Edit Format View Help
def uppercase(text):
    return text.upper()

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

6. Create a Directory in HDFS and copy the Input File to HDFS

```
Hadoop fs -mkdir /piginput
```

```
hadoop fs -put udfs C:\pig\sample.pig /piginput
```

```
C:\hadoop\sbin>Hadoop fs -mkdir /piginput
mkdir: `/piginput': File exists

C:\hadoop\sbin>
```

5. Create pig file "sample.pig":

```
File Edit View

-- Register the Jython standalone JAR using the correct URI format
REGISTER 'file:///C:/jython-standalone-2.7.4.jar';

-- Register the Python UDF script
REGISTER 'C:/pig/my_udf.py' USING jython AS myudfs;

-- Load the input file from HDFS
data = LOAD 'hdfs://localhost:9000/piginput/input.txt' AS (line: chararray);

-- Apply the UDF to convert each line to uppercase
uppercased_data = FOREACH data GENERATE myudfs.to_upper(line);

-- Store the result in HDFS
STORE uppercased_data INTO 'hdfs://localhost:9000/pigOutput/output.txt';
```

6. Execute Pig file:

```
pig -f C:\pig\sample.pig
```

```

C:\hadoop\sbin>pig -f C:\pig\sample.pig
2024-09-14 08:47:02,291 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2024-09-14 08:47:02,296 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2024-09-14 08:47:02,296 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2024-09-14 08:47:02,696 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2024-09-14 08:47:02,697 [main] INFO org.apache.pig.Main - Logging error messages to: C:\hadoop\logs\pig_1726283822682.1.log
2024-09-14 08:47:03,337 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file C:\Users\monid/.pigbootup not found
2024-09-14 08:47:03,426 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2024-09-14 08:47:03,427 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000
2024-09-14 08:47:04,523 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-sample.pig-6562013e-a-b11-405c-a25e-fd4c9a36c3f8
2024-09-14 08:47:04,523 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false

```

7. View the Output `hdfs dfs -ls /pigOutput`

```

C:\hadoop\sbin>hdfs dfs -ls /pigOutput
Found 1 items
drwxr-xr-x - monid supergroup 0 2024-08-27 14:48 /pigOutput/output.txt
C:\hadoop\sbin>

```

`hdfs dfs -cat /pigOutput/output.txt/part-m-00000`

```

C:\hadoop\sbin>hdfs dfs -cat /pigOutput/output.txt/part-m-00000
WELCOME TO HADOOP INSTALLATION
C:\hadoop\sbin>

```

8. Once the map reduce operations are performed successfully, the output will be present in the specified directory.

“/pigOutput/output.data/part-m-00000”

The screenshot displays the Hadoop web interface with a modal window titled "File information - part-m-00000". The modal shows the following details:

- Block information:** Block 0
 - Block ID: 1073741928
 - Block Pool ID: BP-399902486-192.168.228.238-1724038237583
 - Generation Stamp: 1104
 - Size: 31
 - Availability: Moni
- File contents:** WELCOME TO HADOOP INSTALLATION

The background shows the "Browse Directory" view for /pigOutput/output.txt, listing 25 entries. The file list includes a file named "part-m-00000" with a size of 31 bytes.

9. Stop Hadoop Services

```
stop-dfs.cmd stop-  
yarn.cmd
```

Result:

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