PRACTICAL-9

AIM:

To install and run Pig and then write Pig Latin scripts to sort, group, join, project, and filter your data.

THEORY:

Pig:

• Apache Pig is a high-level platform for creating programs that run on Apache Hadoop.

• The language for this platform is called Pig Latin.

• Pig can execute its Hadoop jobs in MapReduce, Apache Tez, or Apache Spark.

• Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for relational database management systems.

• Pig Latin can be extended using user-defined functions (UDFs) which the user can write in Java, Python, JavaScript, Ruby or Groovy and then call directly from the language.

• Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with Hadoop; we can perform all the data manipulation operations in Hadoop using Pig.

• Hadoop Pig is nothing but an abstraction over MapReduce. While it comes to analyze large sets of data, as well as to represent them as data flows, we use Apache Pig. Generally, we use it with Hadoop. By using Pig, we can perform all the data manipulation operations in Hadoop.

• In addition, Pig offers a high-level language to write data analysis programs which we call as Pig Latin. One of the major advantages of this language is, it offers several operators.

• Through them, programmers can develop their own functions for reading, writing, and processing data.

• It has following key properties such as:

• Ease of programming

• Basically, when all the complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, that makes them easy to write, understand, and maintain.

• Optimization opportunities

• It allows users to focus on semantics rather than efficiency, to optimize their execution automatically, in which tasks are encoded permits the system.

• Extensibility

• In order to do special-purpose processing, users can create their own functions.

• Hence, programmers need to write scripts using Pig Latin language to analyze data using Apache Pig.

• However, all these scripts are internally converted to Map and Reduce tasks. It is possible with a component; we call as Pig Engine. That accepts the Pig Latin scripts as input and further convert those scripts into MapReduce jobs.

CODE:

You can start the pig using "pig -x local"

```
admin123@admin123:~$ pig -x local
2021-10-16 23:18:20,023 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2021-10-16 23:18:20,025 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2021-10-16 23:18:20,351 [main] INFO  org.apache.pig.Main - Apache Pig version 0
.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2021-10-16 23:18:20,356 [main] INFO  org.apache.pig.Main - Logging error messag
es to: /home/admin123/pig_1634406500342.log
2021-10-16 23:18:20,509 [main] INFO  org.apache.pig.impl.util.Utils - Default b
ootup file /home/admin123/.pigbootup not found
2021-10-16 23:18:20,772 [main] INFO  org.apache.hadoop.conf.Configuration.depre
cation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.ad
dress
2021-10-16 23:18:20,778 [main] INFO  org.apache.pig.backend.hadoop.executioneng
ine.HExecutionEngine - Connecting to hadoop file system at: file:///
2021-10-16 23:18:21,175 [main] INFO  org.apache.hadoop.conf.Configuration.depre
cation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checks
um
2021-10-16 23:18:21,256 [main] INFO  org.apache.pig.PigServer - Pig Script ID f
or the session: PIG-default-d0a6cf65-8a2c-4384-bf5f-f6801cc9244b
2021-10-16 23:18:21,257 [main] WARN  org.apache.pig.PigServer - ATS is disabled
 since yarn.timeline-service.enabled set to false
grunt> student_details = LOAD './student_details.txt' USING PigStorage(',')
>> as (id:int, firstname:chararray, age:int, city:chararray);
2021-10-16 23:21:21,081 [main] INFO  org.apache.hadoop.conf.Configuration.depre
cation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checks
um
grunt>
```

We will create a file called student_details.txt with input data and load it into pig.

Open

student_details.txt

```
001,Robin,25,London
002,Lucy,23,NY
003,Bobby,26,Paris
004,Max,24,Tokyo
```

```
grunt> student_details = LOAD './student_details.txt' USING PigStorage(',')
>> as (id:int, firstname:chararray, age:int, city:chararray);
2021-10-16 23:27:27,398 [main] INFO  org.apache.hadoop.conf.Configuration.depre
cation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checks
um
```

We can perform different operations using following commands.

```
Order_by_data = ORDER student_details BY age DESC;

Dumb order_by_data;
```

```
grunt> order_by_data = ORDER student_details BY age DESC;
grunt> Dump order_by_data;
```

```
(3,Bobby,26,Paris)
(1,Robin,25,London)
(4,Max,24,Tokyo)
(2,Lucy,23,NY)
grunt>
```

```
Group_data = GROUP student_details by age;

Dump group_data;
```

```
grunt> group_data = GROUP student_details by age;
grunt> Dump group_data;
```

```
(23,{(6,Daisy,23,UK),(2,Lucy,23,NY)})
(24,{(4,Max,24,Tokyo)})
(25,{(5,Carle,25,Yokohama),(1,Robin,25,London)})
(26,{(3,Bobby,26,Paris)})
grunt>
```

```
grunt> group_data = GROUP student_details by age;
grunt> Dump group_data;
```

```
(23,{(6,Daisy,23,UK),(2,Lucy,23,NY)})
(24,{(4,Max,24,Tokyo)})
(25,{(5,Carle,25,Yokohama),(1,Robin,25,London)})
(26,{(3,Bobby,26,Paris)})
grunt>
```

```
Filter_data = FILTER student_details BY age>23;

Dump filter_data;
```

```
(1,Robin,25,London)
(3,Bobby,26,Paris)
(4,Max,24,Tokyo)
(5,Carle,25,Yokohama)
grunt>
```

CONCLUSION:

In this practical, we learnt about pig and used various commands to manipulate data.