1. Apache Pig Getting started

Posted on February 1, 2016

Input Data

scores.data in folder:**/Users/arulk/projects** representing marks of 4 students in 3 subjects:

```

Science, 80, 75, 89, 90
Maths,   90, 87, 78, 92
English, 78, 88, 65, 99

```

Calculate the max mark for each subject.

300+ Java Interview FAQs

300+ Java FAQs
16+ Java Key Areas Q&As
150+ Java Architect FAQs
80+ Java Code Quality Q&As
150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs
Tutorials - Big Data
TUT - Starting Big Data
TUT - Starting Spark & Scala

TOP

Step 1: Download Apache Pig from http://apache.mirror.digitalpacific.com.au/pig/ and extract the tar file.

```

tar -zxvf  ./downloads/pig-0.15.0.tar.gz

```

Step 2: Set up PIG_HOME and add $PIG_HOME/bin to the path via **.profile** or **.bashrc**.

```

export JAVA_HOME=/Library/Java/JavaVirtualMachine
export M3_HOME=~/tools/apache-maven-3.3.9
export HADOOP_HOME=~/hadoop-eco/hadoop-2.7.1
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/

export HBASE_HOME=~/hbase-1.0.3

export PIG_HOME=~/pig-0.15.0

export PATH=$PATH:$M3_HOME/bin:$JAVA_HOME/bin:$HA
export HADOOP_INSTALL=$HADOOP_HOME

```

```

$source .profile # or .bashrc

```

Step 3: Using the '-x local' options starts pig in the local mode whereas executing the pig command without any options starts in Pig in the cluster mode. When in local mode, pig can access files on the local file system. In cluster mode, pig can access files on HDFS.

```

$pig -x local
```

Step 4: Read the file "/Users/arulk/projects/scores.data".

```
grunt>  student_marks = LOAD '/Users/arulk/project
```

Note: If data type is not provided, the default data type is "**byte array**"

The "student_marks" is known as a "**relation**", and NOT a variable. Pig is a **data flow language**. A Pig relation is a bag of **tuples**. A Pig relation is similar to a table in a relational database, where the tuples in the bag correspond to the rows in a table.

A **tuple** is just like a row in a table. It is comma separated list of fields.

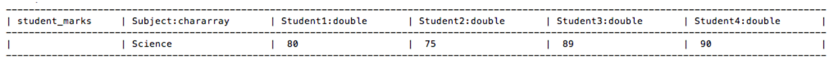
```
(Science, 80, 75, 89, 90)
```

A **bag** is an unordered collection of tuples.

```
{(80.0),(75.0),(89.0),(90.0)}
```

Three handy commands to check the **structure (aka schema)**, hoe the data flow was derived and the **actual data** are:

```
grunt> describe student_marks
grunt> illustrate student_marks
grunt> dump student_marks

```



```
------------------------------------------------------------------------------------------------------------------------------
| student_marks     | Subject:chararray     | Student1:double     | Student2:double     | Student3:double     | Student4:double     |
------------------------------------------------------------------------------------------------------------------------------
|                   | Science               |  80                 |  75                 |  89                 |  90                 |
------------------------------------------------------------------------------------------------------------------------------
```

illustrate student_marks

```

grunt> dump student_marks

(Science, 80, 75, 89, 90)
(Maths,  90, 87, 78, 92)
(English, 78, 88, 65, 99)


```

Step 5: Bag student mark columns.

```

grunt> bagged = foreach student_marks generate Sub

```

```

grunt> dump bagged

(Science,{(80.0),(75.0),(89.0),(90.0)})
(Maths,{(90.0),(87.0),(78.0),(92.0)})
(English,{(78.0),(88.0),(65.0),(99.0)})


```

Step 6: Flattening the Bag.

```

grunt> pivoted_1 = foreach bagged generate Subject

```

```

grunt> dump pivoted_1;

(Science,80.0)
(Science,75.0)
(Science,89.0)
```

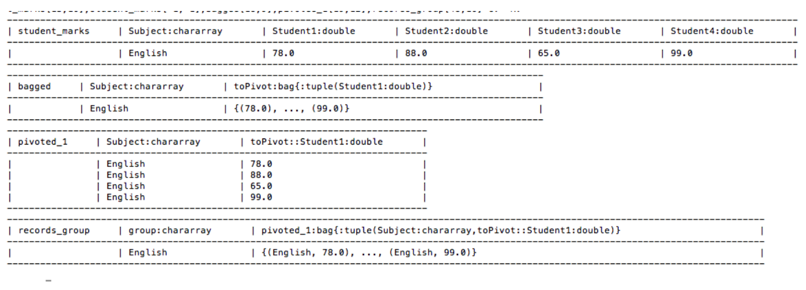
```
(Science,90.0)
(Maths,90.0)
(Maths,87.0)
(Maths,78.0)
(Maths,92.0)
(English,78.0)
(English,88.0)
(English,65.0)
(English,99.0)


```

Step 7: Group it by Subject.

```

grunt> records_group = GROUP pivoted_1 by Subject;

```

```

grunt> dump records_group;

(Maths,{(Maths,92.0),(Maths,78.0),(Maths,87.0),(Ma
(English,{(English,99.0),(English,65.0),(English,8
(Science,{(Science,90.0),(Science,89.0),(Science,7


```

```

grunt> illustrate records_group

```



```
--------------------------------------------------------------------------------------------------------------------------
| student_marks     | Subject:chararray       | Student1:double      | Student2:double      | Student3:double      | Student4:double      |
--------------------------------------------------------------------------------------------------------------------------
|                   | English                 | 78.0                 | 88.0                 | 65.0                 | 99.0                 |
--------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
| bagged     | Subject:chararray       | toPivot:bag{:tuple(Student1:double)}            |
-----------------------------------------------------------------------------------------
|            | English                 | {(78.0), ..., (99.0)}                           |
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------
| pivoted_1     | Subject:chararray       | toPivot::Student1:double       |
-----------------------------------------------------------------------
|               | English                 | 78.0                           |
|               | English                 | 88.0                           |
|               | English                 | 65.0                           |
|               | English                 | 99.0                           |
-----------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
| records_group     | group:chararray        | pivoted_1:bag{:tuple(Subject:chararray,toPivot::Student1:double)}            |
---------------------------------------------------------------------------------------------------------------------
|                   | English                | {(English, 78.0), ..., (English, 99.0)}                                      |
---------------------------------------------------------------------------------------------------------------------
```

illustrate records_group

Step 8: Final result

```

grunt> result = FOREACH records_group GENERATE grou


grunt> dump result

```

```

(Maths,92.0)
(English,99.0)
(Science,90.0)

```

```

grunt> illustrate result

```

‹ 2. HBase Shell commands

2. Apache Pig: Regex (Regular expressions) ›

Disclaimer

Top