INCEPTEZ TECHNOLOGIES PIG WORKOUTS

==========
APACHE PIG V 0.15.0 Installation steps:
==========

1) Goto the below path
cd /home/hduser/install/

2) Extract the tarball,

tar xvzf pig-0.15.0.tar.gz

3) Rename and move the pig folder,

sudo mv pig-0.15.0 /usr/local/pig

4) Modify ~/.bashrc profile to add the pig path to access pig binary from anywhere

vi ~/.bashrc
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin
source ~/.bashrc

5) Start pig in local or distributed mode

pig -x local
or
pig

##

Prerequsite:
Data for the complete workout is available in the below path
/home/hduser/pigdata
bag
coursedetails
custs
map
results
student

testdata
tuple
txns

1. Run pig latin script with param.

Create a script as testpig.pig with the below content

vi testpig.pig

```
raw = LOAD '$INPUTDATA' USING PigStorage('\t');
dump raw;
```

Execute the below script from linux command line

```
pig -x local -f testpig.pig -p INPUTDATA=/home/hduser/pigdata/testdata.txt
```

2. Run Pig in different modes in Grunt shell

```
pig -x local
OR
pig -x mapreduce
```

3. Load, Store and Dump.

```
raw = LOAD '/home/hduser/pigdata/testdata.txt' USING PigStorage('\t');
DUMP raw;
STORE raw INTO '/home/hduser/pigdata/rawstorage';
```

4. Handling complex data type in mapreduce mode.

TUPLE Data: Ordered set of separated list of fields
=====
/home/hduser/pigdata/tuple.txt
(3,8,9) (4,5,6)
(1,4,7) (3,7,5)
(2,5,8) (9,5,8)

Script:
=======

```
A = LOAD '/home/hduser/pigdata/tuple.txt' using PigStorage(' ') AS (t1:tuple(t1a:int,
t1b:int,t1c:int),t2:tuple(t2a:int,t2b:int,t2c:int));
describe A;
X = FOREACH A GENERATE t1.t1a,t2.$0;
dump X
```

BAG Data: Collection of tuples
=====

NAMENODE addressvalue: 192.168.1.2
DATANODE addressvalue: 192.168.1.3
DATANODE addressvalue: 192.168.1.4
DATANODE addressvalue: 192.168.1.5
DATANODE addressvalue: 192.168.1.7
EDGENODE addressvalue: 192.168.1.9

```
B = LOAD '/home/hduser/pigdata/bag.txt' USING PigStorage('\t') AS (node: chararray , ip: chararray);
C = filter B by node matches 'DATANODE';
D = foreach C generate ip;
E = foreach D generate TOKENIZE(ip);
dump E;
```

MAP Data: key value pairs
=====

John,27
Aravindh,30
Bala,22
Srini,32

Script:
=======

```
data = load '/home/hduser/pigdata/map.txt' using PigStorage (',') as (name:chararray, age:int);
B = FOREACH data GENERATE TOMAP(name,age) AS m ;
dump B;
```

5. Filtering columns/rows using foreach and filter commands.

```
a = load '/home/hduser/pigdata/bag.txt' as (c1:chararray, c2:chararray);
dump a;
```

Filter column 1 from the file.

```
b = foreach a generate $0;
dump b;
```

Filter all rows with one or more patterns

```
c = filter a by $0 == 'NAMENODE';
c = filter a by $0 == 'NAMENODE' or $0 == 'EDGENODE';
dump c;
```

6. Trending technologies example with datatype as char array, commands such as foreach generate flatten - tokenize, group by, count etc.

```
lines = LOAD '/home/hduser/pigdata/coursedetails.txt' AS (line:chararray);
```

```
words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) as word;
grouped = GROUP words BY word;
wordcount = FOREACH grouped GENERATE group, COUNT(words);
DUMP wordcount;

orderedout = order wordcount by $1 desc;
DUMP orderedout;

Distinct example
#################

distval = distinct words;
dump distval;
```

7. Sample customer data to demonstrate Load, limit, group by, count, join etc.

A. Load Customer records

```
cust = LOAD '/home/hduser/pigdata/custs' using PigStorage(',') AS (
custid:chararray,firstname:chararray, lastname:chararray, age:long, profession:chararray);
```

B. Select only 100 records

```
lmt = LIMIT cust 100;
dump lmt;
```

C. Group customer records by profession

```
groupbyprofession = GROUP cust BY profession;
```

D. Count no of customers by profession

```
countbyprofession = FOREACH groupbyprofession GENERATE group, COUNT (cust);
dump countbyprofession;
```

E. Load transaction records

```
txn = LOAD '/home/hduser/pigdata/txns' using PigStorage(',') AS ( txnid:chararray,
date:chararray,custid:chararray, amount:double, category:chararray, product:chararray,city:chararray,
state:chararray, type:chararray);
```

F. Group transactions by customer

```
txnbycust = group txn by custid;
```

G. Sum total amount spent by each customer

```
spendbycust = foreach txnbycust generate group, SUM(txn.amount);
```

H. Order the customer records beginning from highest spender

```
custorder = order spendbycust by $1 desc;
custorderleast = order spendbycust by $1;

I. Select only top 100 customers
top100buyers = limit custorder 100;
least100buyers = limit custorderleast 100;

J. Join the transactions with customer details
top100buyerstxnsjoin = join top100buyers by $0, cust by $0;
least100buyerstxnsjoin = join least100buyers by $0, cust by $0;

K. Select the required fields from the join for final output
top100buyerstxnscolumns = foreach top100buyerstxnsjoin generate $0, $3, $4, $5, $6, $1;
least100buyerstxnscolumns = foreach least100buyerstxnsjoin generate $0, $3, $4, $5, $6, $1;

L. Order the customer based on the highest spender.
realtop100 = order top100buyerstxnscolumns by $5 desc;
realbottom100 = order least100buyerstxnscolumns by $5;

M. Dump the final output
dump realtop100;
dump realbottom100;

8) STREAMING - use linux command to stream and cut only the first name and last name.

C = STREAM realtop100 THROUGH `cut -f 2-3`;
DUMP C;

9) UNION

top10 = limit realtop100 10;
bottom10 = limit realbottom100 10;
union20 = UNION top10, bottom10;
dump union20;
```

PIG USE CASE 1: Weblog analysis

```
################################
```

Execute the following set of codes to analyze/process weblog data using pig.

```
Create a dir in hdfs
hadoop fs -mkdir /user/hduser/weblogs/

cd /home/hduser/pigdata
hadoop fs -put weblogs_parse.txt /user/hduser/weblogs/

create pig variable with schema
```

```
weblogs = LOAD '/user/hduser/weblogs/web*' USING PigStorage('\t')
      AS (
client_ip:chararray,
full_request_date:chararray,
day:int,
month:chararray,
month_num:int,
year:int,
hour:int,
minute:int,
second:int,
timezone:chararray,
http_verb:chararray,
uri:chararray,
http_status_code:chararray,
bytes_returned:chararray,
referrer:chararray,
user_agent:chararray
);
```

Dump the schema

```
DUMP weblogs;
```

Describe the schema

```
DESCRIBE weblogs;
```

Filter only careers uri and explain on the execution

```
careers_visitors = FILTER weblogs BY uri == '/careers';
EXPLAIN b;
DUMP careers_visitors;
```

Perform foreach iteration to get the ip of the corresponding careers visitors

```
select_fields = FOREACH careers_visitors GENERATE client_ip;
DUMP select_fields;
```

Group the data based on http_verb and count the weblogs

```
groups_fields = GROUP weblogs BY http_verb;
count = FOREACH groups_fields GENERATE group,COUNT(weblogs);
DUMP count;
```

PIG USE CASE 2: Student career analytics

##

This assignment helps you in applying your Pig commands and scripting knowledge from your Live Class.
You need to use the Pig commands to process two different data sets and print the name of all the successful students in an exam.

You have two datasets:
Student: This dataset contains name and roll number of students in a class (/home/hduser/pigdata/student)
Results: This dataset contains roll number and result (Fail or Pass) of students (/home/hduser/pigdata/results)

Problem statement
Write a Pig script to analyse the given datasets and print the student names who have successfully cleared the exam. You should use only PIG commands to achieve the desired result.

Solution approach

Use the following Pig commands in the sequence to join the datasets and find out the result:
1. Load the student data in a variable
2. Load the results data in a variable
3. Join student variable and results variable.
4. Use foreach and generate name and status.
5. filter only pass status.
6. dump the result.

Solution Code (please refer after trying yourself to solve the problem)

```
Student = load '/home/hduser/pigdata/student' as (name:chararray, roll:int);
Results = load '/home/hduser/pigdata/results' as (roll_:int, status:chararray);
PassStudents = join Student by roll, Results by roll_;
PassStudentNames = foreach PassStudents generate name, status;
Passed = filter PassStudentNames by status matches 'pass';
dump Passed;
```