PIG WORKOUTS

## =======================================

**APACHE PIG V 0.16.0 Installation steps:**

**=======================================**

1. Goto the below path cd /home/hduser/install/
2. Extract the tarfile,

tar xvzf pig-0.16.0.tar.gz

1. Rename and move the pig folder, sudo mv pig-0.16.0 /usr/local/pig Start the History Server:

mr-jobhistory-daemon.sh start historyserver

**Workouts:**

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

1. Run Pig in different modes in Grunt shell

pig -x local OR

pig -x mapreduce Pig

1. Load, Store and Dump.

raw = LOAD '/home/hduser/pigdata/testdata.txt' USING PigStorage('\t');

DUMP raw;

STORE raw INTO '/home/hduser/pigdata/rawstorage';

1. 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;

1. 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;

1. 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;

1. Sample customer data to demonstrate Load, limit, group by, count, join etc.
   1. *Load Customer records*

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

* 1. *Select only 100 records*

lmt = LIMIT cust 100; dump lmt;

* 1. *Group customer records by profession*

groupbyprofession = GROUP cust BY profession;

* 1. *Count no of customers by profession*

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

* 1. *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);

* 1. *Group transactions by customer*

txnbycust = group txn by custid;

* 1. *Sum total amount spent by each customer*

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

* 1. *Order the customer records beginning from highest spender*

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

* 1. *Select only top 100 customers*

top100buyers = limit custorder 100; least100buyers = limit custorderleast 100;

* 1. *Join the transactions with customer details* top100buyerstxnsjoin = join top100buyers by $0, cust by $0; least100buyerstxnsjoin = join least100buyers by $0, cust by $0;
  2. *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;

* 1. *Order the customer based on the highest spender.* realtop100 = order top100buyerstxnscolumns by $5 desc; realbottom100 = order least100buyerstxnscolumns by $5;
  2. *Dump the final output*

dump realtop100; dump realbottom100;

1. 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;

1. UNION

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

dump union20;