Đỗ Hoàng Thịnh – 20133122

PIG LABORATORY

**0. Set up**

**0.1. Mở HDFS**

- Mở CMD chế độ admin

chạy lệnh

%HADOOP\_HOME%\sbin\start-dfs.cmd

%HADOOP\_HOME%\sbin\start-yarn.cmd

A screenshot of a computer screen

Description automatically generated with medium confidence

**0.2. Pig 2 chế độ**

- Local

A screenshot of a computer

Description automatically generated

- Mapreduce

A screenshot of a computer

Description automatically generated

**0.3. Tắt Safemode**

A screenshot of a computer

Description automatically generated with medium confidence

**0.4. Chuẩn bị File dữ liệu**

- Tstat

A screen shot of a computer

Description automatically generated with medium confidence

- 100linee

A screenshot of a computer

Description automatically generated

**1. Exercise 1**

**1.1. Code**

A = LOAD 'C:/PigData/tstat-sample.txt' using PigStorage(' ') AS (ip\_c:chararray, port\_c:int,

packets\_c:int,

rst\_c:int,

ack\_c:int,

purack\_c:int,

unique\_bytes\_c:long,

data\_pkts\_c:int,

data\_bytes\_c:long,

rexmit\_pkts\_c:int,

rexmit\_bytes\_c:long,

out\_seq\_pkts\_c:int,

syn\_c:int,

fin\_c:int,

ws\_c:int,

ts\_c:int,

window\_scale\_c:int,

sack\_req\_c:int,

sack\_sent\_c:int,

mss\_c:int,

max\_seg\_size\_c:int,

min\_seg\_size\_c:int,

win\_max\_c:int,

win\_min\_c:int,

win\_zero\_c:int,

cwin\_max\_c:long,

cwin\_min\_c:long,

initial\_cwin\_c:long,

average\_rtt\_c:double,

rtt\_min\_c:double,

rtt\_max\_c:double,

stdev\_rtt\_c:double,

rtt\_count\_c:int,

ttl\_min\_c:int,

ttl\_max\_c:int,

rtx\_RTO\_c:int,

rtx\_FR\_c:int,

reordering\_c:int,

net\_dup\_c:int,

unknown\_segments\_c:int,

flow\_control\_c:int,

unnece\_rtx\_rto\_c:int,

unnece\_rtx\_fr\_c:int,

different\_syn\_seqno\_c:int,

ip\_s:chararray,

port\_s:int,

packets\_s:int,

rst\_s:int,

ack\_s:int,

purack\_s:int,

unique\_bytes\_s:long,

data\_pkts\_s:int,

data\_bytes\_s:long,

rexmit\_pkts\_s:int,

rexmit\_bytes\_s:long,

out\_seq\_pkts\_s:int,

syn\_s:int,

fin\_s:int,

ws\_s:int,

ts\_s:int,

window\_scale\_s:int,

sack\_req\_s:int,

sack\_sent\_s:int,

mss\_s:int,

max\_seg\_size\_s:int,

min\_seg\_size\_s:int,

win\_max\_s:int,

win\_min\_s:int,

win\_zero\_s:int,

cwin\_max\_s:long,

cwin\_min\_s:long,

initial\_cwin\_s:long,

average\_rtt\_s:double,

rtt\_min\_s:double,

rtt\_max\_s:double,

stdev\_rtt\_s:double,

rtt\_count\_s:int,

ttl\_min\_s:int,

ttl\_max\_s:int,

rtx\_RTO\_s:int,

rtx\_FR\_s:int,

reordering\_s:int,

net\_dup\_s:int,

unknown\_segments\_s:int,

flow\_control\_s:int,

unnece\_rtx\_rto\_s:int,

unnece\_rtx\_fr\_s:int,

different\_syn\_seqno\_s:int,

completion\_time:double,

first\_time:double,

last\_time:double,

C\_first\_payload:double,

S\_first\_payload:double,

C\_last\_payload:double,

S\_last\_payload:double,

C\_first\_ack:double,

S\_first\_ack:double,

first\_time\_abs:double,

c\_internal:int,

s\_internal:int,

connection\_type:int,

p2p\_type:int,

p2p\_subtype:int,

ed2k\_data:int,

ed2k\_signaling:int,

ed2k\_c2s:int,

ed2k\_c2c:int,

ed2k\_chat:int,

http\_type:int,

ssl\_client\_hello:chararray,

ssl\_server\_hello:chararray,

dropbox\_id:bytearray,

fqdn:chararray);

B = GROUP A BY ip\_c;

C = FOREACH B GENERATE group, COUNT(A);

STORE C INTO '/PigData/output/ex1';

**1.2.Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex1.pig”

A screenshot of a computer

Description automatically generated

**1.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated with medium confidence

**2. Exercise 2**

Mục đích của bài tập này là đếm tổng số kết nối TCP có **google.it** trong FDQN (trường 110 trong tập dữ liệu **tstat-sample.txt**)

**2.1. Code**

RAW\_DATA = LOAD 'C:/PigData/tstat-sample.txt' using PigStorage(' ') AS (ip\_c:chararray,

port\_c:int,

packets\_c:int,

rst\_c:int,

ack\_c:int,

purack\_c:int,

unique\_bytes\_c:long,

data\_pkts\_c:int,

data\_bytes\_c:long,

rexmit\_pkts\_c:int,

rexmit\_bytes\_c:long,

out\_seq\_pkts\_c:int,

syn\_c:int,

fin\_c:int,

ws\_c:int,

ts\_c:int,

window\_scale\_c:int,

sack\_req\_c:int,

sack\_sent\_c:int,

mss\_c:int,

max\_seg\_size\_c:int,

min\_seg\_size\_c:int,

win\_max\_c:int,

win\_min\_c:int,

win\_zero\_c:int,

cwin\_max\_c:long,

cwin\_min\_c:long,

initial\_cwin\_c:long,

average\_rtt\_c:double,

rtt\_min\_c:double,

rtt\_max\_c:double,

stdev\_rtt\_c:double,

rtt\_count\_c:int,

ttl\_min\_c:int,

ttl\_max\_c:int,

rtx\_RTO\_c:int,

rtx\_FR\_c:int,

reordering\_c:int,

net\_dup\_c:int,

unknown\_segments\_c:int,

flow\_control\_c:int,

unnece\_rtx\_rto\_c:int,

unnece\_rtx\_fr\_c:int,

different\_syn\_seqno\_c:int,

ip\_s:chararray,

port\_s:int,

packets\_s:int,

rst\_s:int,

ack\_s:int,

purack\_s:int,

unique\_bytes\_s:long,

data\_pkts\_s:int,

data\_bytes\_s:long,

rexmit\_pkts\_s:int,

rexmit\_bytes\_s:long,

out\_seq\_pkts\_s:int,

syn\_s:int,

fin\_s:int,

ws\_s:int,

ts\_s:int,

window\_scale\_s:int,

sack\_req\_s:int,

sack\_sent\_s:int,

mss\_s:int,

max\_seg\_size\_s:int,

min\_seg\_size\_s:int,

win\_max\_s:int,

win\_min\_s:int,

win\_zero\_s:int,

cwin\_max\_s:long,

cwin\_min\_s:long,

initial\_cwin\_s:long,

average\_rtt\_s:double,

rtt\_min\_s:double,

rtt\_max\_s:double,

stdev\_rtt\_s:double,

rtt\_count\_s:int,

ttl\_min\_s:int,

ttl\_max\_s:int,

rtx\_RTO\_s:int,

rtx\_FR\_s:int,

reordering\_s:int,

net\_dup\_s:int,

unknown\_segments\_s:int,

flow\_control\_s:int,

unnece\_rtx\_rto\_s:int,

unnece\_rtx\_fr\_s:int,

different\_syn\_seqno\_s:int,

completion\_time:double,

first\_time:double,

last\_time:double,

C\_first\_payload:double,

S\_first\_payload:double,

C\_last\_payload:double,

S\_last\_payload:double,

C\_first\_ack:double,

S\_first\_ack:double,

first\_time\_abs:double,

c\_internal:int,

s\_internal:int,

connection\_type:int,

p2p\_type:int,

p2p\_subtype:int,

ed2k\_data:int,

ed2k\_signaling:int,

ed2k\_c2s:int,

ed2k\_c2c:int,

ed2k\_chat:int,

http\_type:int,

ssl\_client\_hello:chararray,

ssl\_server\_hello:chararray,

dropbox\_id:bytearray,

fqdn:chararray);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE ip\_c, $110;

DATA\_FILTERED = FILTER DATA BY $1 MATCHES '.\*google.com.\*';

--dump DATA\_FILTERED;

--describe DATA\_FILTERED;

-- Count #connections

COUNT\_GROUP = GROUP DATA\_FILTERED ALL;

COUNT = FOREACH COUNT\_GROUP GENERATE COUNT\_STAR(DATA\_FILTERED);

-- Store the output

STORE COUNT INTO '/PigData/output/ex2';

**2.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex2.pig”

A screen shot of a computer

Description automatically generated with medium confidence

**2.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**3.Exercise 3**

Mục đích của bài này là tính tổng số byte được tải lên, tải xuống và tổng số byte (lên và xuống) của mỗi IP.

**3.1. Code**

RAW\_DATA = LOAD 'C:/PigData/100linee.txt'

AS (ts:long, sport, dport, sip, dip,

l3proto, l4proto, flags,

phypkt, netpkt, overhead,

phybyte, netbyte:long);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE sip, dip, netbyte;

DATA\_UP = GROUP DATA BY sip;

FLOW\_UP = FOREACH DATA\_UP GENERATE group as id\_up, SUM(DATA.netbyte) as upload;

DATA\_DOWN = GROUP DATA BY dip;

FLOW\_DOWN = FOREACH DATA\_DOWN GENERATE group as id\_down, SUM(DATA.netbyte) as download;

FLOW\_JOIN = JOIN FLOW\_UP by id\_up FULL, FLOW\_DOWN by id\_down;

SUMMARY = FOREACH FLOW\_JOIN GENERATE

(id\_up is null?id\_down:id\_up) AS IP,

(upload is null?0:upload) as upload,

(download is null?0:download) as download,

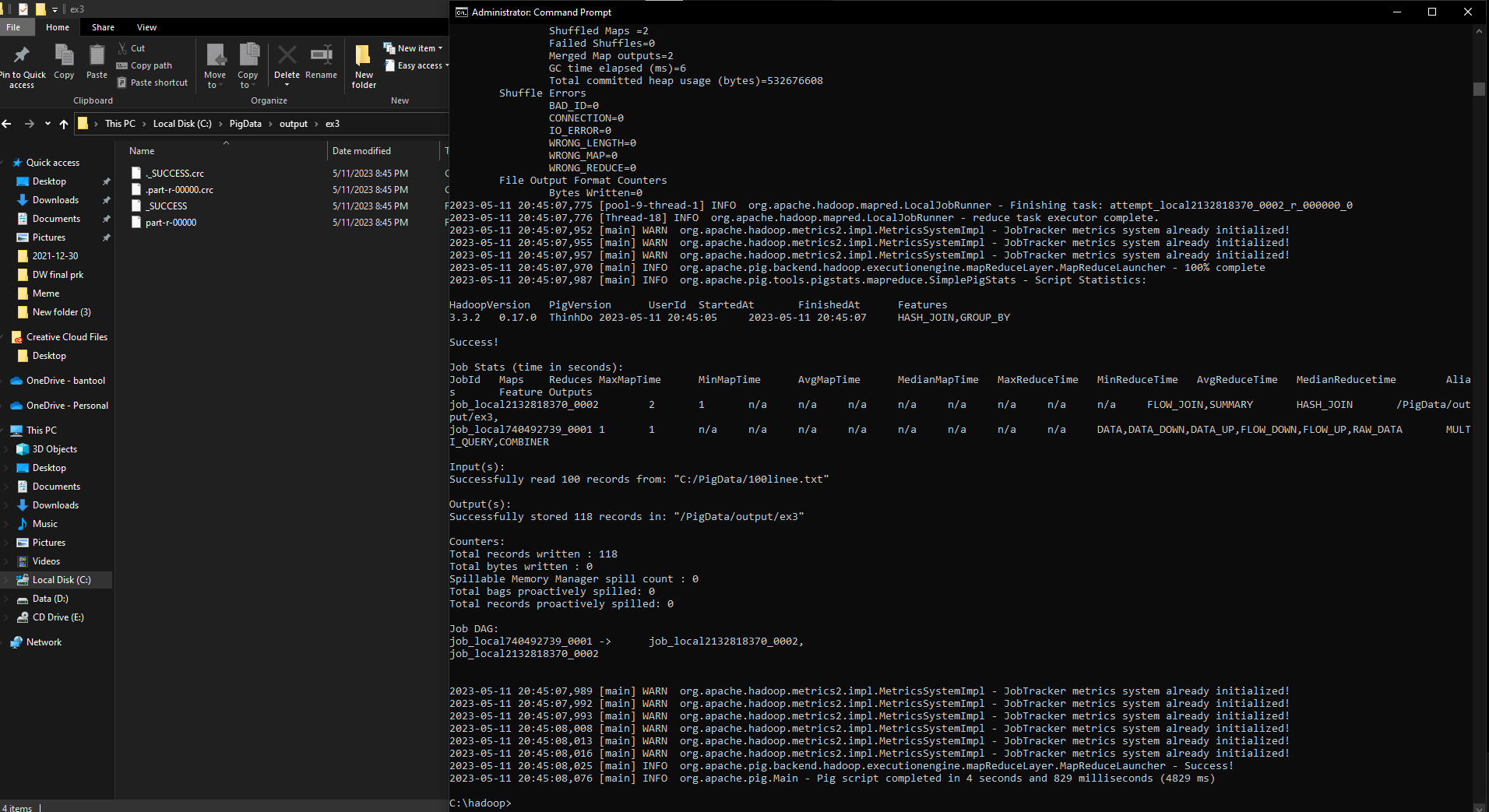
((upload is null?0:upload)+(download is null?0:download)) as total;

-- Store the output

STORE SUMMARY INTO '/PigData/output/ex3';

**3.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex3.pig”



**3.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**4.Exercise 4**

Mục đích của bài này là tìm top 100 địa chỉ IP có lượt tải lên cao nhất

**4.1. Code**

RAW\_DATA = LOAD 'C:/PigData/100linee.txt'

AS (ts:long, sport, dport, sip, dip,

l3proto, l4proto, flags,

phypkt, netpkt, overhead,

phybyte, netbyte:long);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE sip, netbyte;

DATA\_UP = GROUP DATA BY sip;

FLOW\_UP = FOREACH DATA\_UP GENERATE group as IP, SUM(DATA.netbyte) as upload;

SUMMARY\_SORTED = ORDER FLOW\_UP BY upload DESC;

SUMMARY\_TOP100 = LIMIT SUMMARY\_SORTED 100;

-- Store the output

STORE SUMMARY\_TOP100 INTO '/PigData/output/ex4';

**4.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex4.pig”

A screenshot of a computer screen

Description automatically generated with medium confidence

**4.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**5.Exercise 5**

**5.1. Code**

RAW\_DATA = LOAD 'C:/PigData/100linee.txt'

AS (ts:long, sport, dport, sip, dip,

l3proto, l4proto, flags,

phypkt, netpkt, overhead,

phybyte, netbyte:long);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE sip, netbyte as upload;

DATA\_UP = GROUP DATA BY sip;

FLOW\_UP = FOREACH DATA\_UP GENERATE group as ip, SUM(DATA.upload) as sum\_upload;

FLOW\_UP\_SORTED = ORDER FLOW\_UP BY sum\_upload DESC;

FLOW\_UP\_TOP100 = LIMIT FLOW\_UP\_SORTED 100;

FLOW\_JOIN = JOIN FLOW\_UP\_TOP100 BY ip, DATA BY sip;

FLOW\_JOIN\_GROUP = GROUP FLOW\_JOIN BY ip;

RESULT = FOREACH FLOW\_JOIN\_GROUP GENERATE group, MAX(FLOW\_JOIN.upload), (double)100 \* MAX(FLOW\_JOIN.upload) / MAX(FLOW\_JOIN.sum\_upload);

-- Store the output

STORE RESULT INTO '/PigData/output/ex5';

**5.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex5.pig”

A screenshot of a computer screen

Description automatically generated with medium confidence

**5.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**6.Exercise 6**

**6.1. Code**

-- Load raw data generated by tcpdump

RAW\_DATA = LOAD 'C:/PigData/tstat-sample.txt' using PigStorage(' ') AS (ip\_c:chararray,

port\_c:int,

packets\_c:int,

rst\_c:int,

ack\_c:int,

purack\_c:int,

unique\_bytes\_c:long,

data\_pkts\_c:int,

data\_bytes\_c:long,

rexmit\_pkts\_c:int,

rexmit\_bytes\_c:long,

out\_seq\_pkts\_c:int,

syn\_c:int,

fin\_c:int,

ws\_c:int,

ts\_c:int,

window\_scale\_c:int,

sack\_req\_c:int,

sack\_sent\_c:int,

mss\_c:int,

max\_seg\_size\_c:int,

min\_seg\_size\_c:int,

win\_max\_c:int,

win\_min\_c:int,

win\_zero\_c:int,

cwin\_max\_c:long,

cwin\_min\_c:long,

initial\_cwin\_c:long,

average\_rtt\_c:double,

rtt\_min\_c:double,

rtt\_max\_c:double,

stdev\_rtt\_c:double,

rtt\_count\_c:int,

ttl\_min\_c:int,

ttl\_max\_c:int,

rtx\_RTO\_c:int,

rtx\_FR\_c:int,

reordering\_c:int,

net\_dup\_c:int,

unknown\_segments\_c:int,

flow\_control\_c:int,

unnece\_rtx\_rto\_c:int,

unnece\_rtx\_fr\_c:int,

different\_syn\_seqno\_c:int,

ip\_s:chararray,

port\_s:int,

packets\_s:int,

rst\_s:int,

ack\_s:int,

purack\_s:int,

unique\_bytes\_s:long,

data\_pkts\_s:int,

data\_bytes\_s:long,

rexmit\_pkts\_s:int,

rexmit\_bytes\_s:long,

out\_seq\_pkts\_s:int,

syn\_s:int,

fin\_s:int,

ws\_s:int,

ts\_s:int,

window\_scale\_s:int,

sack\_req\_s:int,

sack\_sent\_s:int,

mss\_s:int,

max\_seg\_size\_s:int,

min\_seg\_size\_s:int,

win\_max\_s:int,

win\_min\_s:int,

win\_zero\_s:int,

cwin\_max\_s:long,

cwin\_min\_s:long,

initial\_cwin\_s:long,

average\_rtt\_s:double,

rtt\_min\_s:double,

rtt\_max\_s:double,

stdev\_rtt\_s:double,

rtt\_count\_s:int,

ttl\_min\_s:int,

ttl\_max\_s:int,

rtx\_RTO\_s:int,

rtx\_FR\_s:int,

reordering\_s:int,

net\_dup\_s:int,

unknown\_segments\_s:int,

flow\_control\_s:int,

unnece\_rtx\_rto\_s:int,

unnece\_rtx\_fr\_s:int,

different\_syn\_seqno\_s:int,

completion\_time:double,

first\_time:double,

last\_time:double,

C\_first\_payload:double,

S\_first\_payload:double,

C\_last\_payload:double,

S\_last\_payload:double,

C\_first\_ack:double,

S\_first\_ack:double,

first\_time\_abs:double,

c\_internal:int,

s\_internal:int,

connection\_type:int,

p2p\_type:int,

p2p\_subtype:int,

ed2k\_data:int,

ed2k\_signaling:int,

ed2k\_c2s:int,

ed2k\_c2c:int,

ed2k\_chat:int,

http\_type:int,

ssl\_client\_hello:chararray,

ssl\_server\_hello:chararray,

dropbox\_id:bytearray,

fqdn:chararray);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE mss\_c;

DATA\_SORTED = ORDER DATA BY mss\_c;

DATA\_MIN = LIMIT DATA\_SORTED 1;

-- Store the output

STORE DATA\_MIN INTO '/PigData/output/ex6';

**6.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex6.pig”

A screen shot of a computer

Description automatically generated with medium confidence

**6.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**7.Exercise 7**

**7.1. Code**

-- Load raw data generated by tcpdump

RAW\_DATA = LOAD 'C:/PigData/tstat-sample.txt' using PigStorage(' ') AS (ip\_c:chararray,

port\_c:int,

packets\_c:int,

rst\_c:int,

ack\_c:int,

purack\_c:int,

unique\_bytes\_c:long,

data\_pkts\_c:int,

data\_bytes\_c:long,

rexmit\_pkts\_c:int,

rexmit\_bytes\_c:long,

out\_seq\_pkts\_c:int,

syn\_c:int,

fin\_c:int,

ws\_c:int,

ts\_c:int,

window\_scale\_c:int,

sack\_req\_c:int,

sack\_sent\_c:int,

mss\_c:int,

max\_seg\_size\_c:int,

min\_seg\_size\_c:int,

win\_max\_c:int,

win\_min\_c:int,

win\_zero\_c:int,

cwin\_max\_c:long,

cwin\_min\_c:long,

initial\_cwin\_c:long,

average\_rtt\_c:double,

rtt\_min\_c:double,

rtt\_max\_c:double,

stdev\_rtt\_c:double,

rtt\_count\_c:int,

ttl\_min\_c:int,

ttl\_max\_c:int,

rtx\_RTO\_c:int,

rtx\_FR\_c:int,

reordering\_c:int,

net\_dup\_c:int,

unknown\_segments\_c:int,

flow\_control\_c:int,

unnece\_rtx\_rto\_c:int,

unnece\_rtx\_fr\_c:int,

different\_syn\_seqno\_c:int,

ip\_s:chararray,

port\_s:int,

packets\_s:int,

rst\_s:int,

ack\_s:int,

purack\_s:int,

unique\_bytes\_s:long,

data\_pkts\_s:int,

data\_bytes\_s:long,

rexmit\_pkts\_s:int,

rexmit\_bytes\_s:long,

out\_seq\_pkts\_s:int,

syn\_s:int,

fin\_s:int,

ws\_s:int,

ts\_s:int,

window\_scale\_s:int,

sack\_req\_s:int,

sack\_sent\_s:int,

mss\_s:int,

max\_seg\_size\_s:int,

min\_seg\_size\_s:int,

win\_max\_s:int,

win\_min\_s:int,

win\_zero\_s:int,

cwin\_max\_s:long,

cwin\_min\_s:long,

initial\_cwin\_s:long,

average\_rtt\_s:double,

rtt\_min\_s:double,

rtt\_max\_s:double,

stdev\_rtt\_s:double,

rtt\_count\_s:int,

ttl\_min\_s:int,

ttl\_max\_s:int,

rtx\_RTO\_s:int,

rtx\_FR\_s:int,

reordering\_s:int,

net\_dup\_s:int,

unknown\_segments\_s:int,

flow\_control\_s:int,

unnece\_rtx\_rto\_s:int,

unnece\_rtx\_fr\_s:int,

different\_syn\_seqno\_s:int,

completion\_time:double,

first\_time:double,

last\_time:double,

C\_first\_payload:double,

S\_first\_payload:double,

C\_last\_payload:double,

S\_last\_payload:double,

C\_first\_ack:double,

S\_first\_ack:double,

first\_time\_abs:double,

c\_internal:int,

s\_internal:int,

connection\_type:int,

p2p\_type:int,

p2p\_subtype:int,

ed2k\_data:int,

ed2k\_signaling:int,

ed2k\_c2s:int,

ed2k\_c2c:int,

ed2k\_chat:int,

http\_type:int,

ssl\_client\_hello:chararray,

ssl\_server\_hello:chararray,

dropbox\_id:bytearray,

fqdn:chararray);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE win\_min\_c;

DATA\_GROUP = GROUP DATA ALL;

COUNT\_ALL = FOREACH DATA\_GROUP GENERATE 'same' AS key, (double)COUNT\_STAR(DATA) AS val;

DATA\_FILTERED = FILTER DATA BY win\_min\_c == 14600;

DATA\_FILTERED\_GROUP = GROUP DATA\_FILTERED ALL;

COUNT\_FILTERED = FOREACH DATA\_FILTERED\_GROUP GENERATE 'same' AS key, (double)COUNT\_STAR(DATA\_FILTERED) AS val;

COUNT\_JOIN = JOIN COUNT\_ALL BY key, COUNT\_FILTERED BY key;

PERCENTAGE = FOREACH COUNT\_JOIN GENERATE 100 \* COUNT\_FILTERED.val / COUNT\_ALL.val;

-- Store the output

STORE PERCENTAGE INTO '/PigData/output/ex7';

**7.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex7.pig”

A screenshot of a computer screen

Description automatically generated with medium confidence

**7.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**8.Exercise 8**

**8.1. Code**

-- Load raw data generated by tcpdump

RAW\_DATA = LOAD 'C:/PigData/tstat-sample.txt' using PigStorage(' ') AS (ip\_c:chararray,

port\_c:int,

packets\_c:int,

rst\_c:int,

ack\_c:int,

purack\_c:int,

unique\_bytes\_c:long,

data\_pkts\_c:int,

data\_bytes\_c:long,

rexmit\_pkts\_c:int,

rexmit\_bytes\_c:long,

out\_seq\_pkts\_c:int,

syn\_c:int,

fin\_c:int,

ws\_c:int,

ts\_c:int,

window\_scale\_c:int,

sack\_req\_c:int,

sack\_sent\_c:int,

mss\_c:int,

max\_seg\_size\_c:int,

min\_seg\_size\_c:int,

win\_max\_c:int,

win\_min\_c:int,

win\_zero\_c:int,

cwin\_max\_c:long,

cwin\_min\_c:long,

initial\_cwin\_c:long,

average\_rtt\_c:double,

rtt\_min\_c:double,

rtt\_max\_c:double,

stdev\_rtt\_c:double,

rtt\_count\_c:int,

ttl\_min\_c:int,

ttl\_max\_c:int,

rtx\_RTO\_c:int,

rtx\_FR\_c:int,

reordering\_c:int,

net\_dup\_c:int,

unknown\_segments\_c:int,

flow\_control\_c:int,

unnece\_rtx\_rto\_c:int,

unnece\_rtx\_fr\_c:int,

different\_syn\_seqno\_c:int,

ip\_s:chararray,

port\_s:int,

packets\_s:int,

rst\_s:int,

ack\_s:int,

purack\_s:int,

unique\_bytes\_s:long,

data\_pkts\_s:int,

data\_bytes\_s:long,

rexmit\_pkts\_s:int,

rexmit\_bytes\_s:long,

out\_seq\_pkts\_s:int,

syn\_s:int,

fin\_s:int,

ws\_s:int,

ts\_s:int,

window\_scale\_s:int,

sack\_req\_s:int,

sack\_sent\_s:int,

mss\_s:int,

max\_seg\_size\_s:int,

min\_seg\_size\_s:int,

win\_max\_s:int,

win\_min\_s:int,

win\_zero\_s:int,

cwin\_max\_s:long,

cwin\_min\_s:long,

initial\_cwin\_s:long,

average\_rtt\_s:double,

rtt\_min\_s:double,

rtt\_max\_s:double,

stdev\_rtt\_s:double,

rtt\_count\_s:int,

ttl\_min\_s:int,

ttl\_max\_s:int,

rtx\_RTO\_s:int,

rtx\_FR\_s:int,

reordering\_s:int,

net\_dup\_s:int,

unknown\_segments\_s:int,

flow\_control\_s:int,

unnece\_rtx\_rto\_s:int,

unnece\_rtx\_fr\_s:int,

different\_syn\_seqno\_s:int,

completion\_time:double,

first\_time:double,

last\_time:double,

C\_first\_payload:double,

S\_first\_payload:double,

C\_last\_payload:double,

S\_last\_payload:double,

C\_first\_ack:double,

S\_first\_ack:double,

first\_time\_abs:double,

c\_internal:int,

s\_internal:int,

connection\_type:int,

p2p\_type:int,

p2p\_subtype:int,

ed2k\_data:int,

ed2k\_signaling:int,

ed2k\_c2s:int,

ed2k\_c2c:int,

ed2k\_chat:int,

http\_type:int,

ssl\_client\_hello:chararray,

ssl\_server\_hello:chararray,

dropbox\_id:bytearray,

fqdn:chararray);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE port\_s as port, data\_bytes\_s as bytes;

DATA\_GROUP\_ALL = GROUP DATA ALL;

SUM\_ALL = FOREACH DATA\_GROUP\_ALL GENERATE SUM(DATA.bytes) as total;

DATA\_GROUP = GROUP DATA BY port;

SUM\_GROUP = FOREACH DATA\_GROUP GENERATE group as port, (double)100 \* SUM(DATA.bytes) / SUM\_ALL.total;

-- Store the output

STORE SUM\_GROUP INTO '/PigData/output/ex8';

**8.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex8.pig”

A screenshot of a computer

Description automatically generated

**8.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated

**9.Exercise 9**

**9.1. Code**

-- Load raw data generated by tcpdump

RAW\_DATA = LOAD 'C:/PigData/tstat-sample.txt' using PigStorage(' ') AS (ip\_c:chararray,

port\_c:int,

packets\_c:int,

rst\_c:int,

ack\_c:int,

purack\_c:int,

unique\_bytes\_c:long,

data\_pkts\_c:int,

data\_bytes\_c:long,

rexmit\_pkts\_c:int,

rexmit\_bytes\_c:long,

out\_seq\_pkts\_c:int,

syn\_c:int,

fin\_c:int,

ws\_c:int,

ts\_c:int,

window\_scale\_c:int,

sack\_req\_c:int,

sack\_sent\_c:int,

mss\_c:int,

max\_seg\_size\_c:int,

min\_seg\_size\_c:int,

win\_max\_c:int,

win\_min\_c:int,

win\_zero\_c:int,

cwin\_max\_c:long,

cwin\_min\_c:long,

initial\_cwin\_c:long,

average\_rtt\_c:double,

rtt\_min\_c:double,

rtt\_max\_c:double,

stdev\_rtt\_c:double,

rtt\_count\_c:int,

ttl\_min\_c:int,

ttl\_max\_c:int,

rtx\_RTO\_c:int,

rtx\_FR\_c:int,

reordering\_c:int,

net\_dup\_c:int,

unknown\_segments\_c:int,

flow\_control\_c:int,

unnece\_rtx\_rto\_c:int,

unnece\_rtx\_fr\_c:int,

different\_syn\_seqno\_c:int,

ip\_s:chararray,

port\_s:int,

packets\_s:int,

rst\_s:int,

ack\_s:int,

purack\_s:int,

unique\_bytes\_s:long,

data\_pkts\_s:int,

data\_bytes\_s:long,

rexmit\_pkts\_s:int,

rexmit\_bytes\_s:long,

out\_seq\_pkts\_s:int,

syn\_s:int,

fin\_s:int,

ws\_s:int,

ts\_s:int,

window\_scale\_s:int,

sack\_req\_s:int,

sack\_sent\_s:int,

mss\_s:int,

max\_seg\_size\_s:int,

min\_seg\_size\_s:int,

win\_max\_s:int,

win\_min\_s:int,

win\_zero\_s:int,

cwin\_max\_s:long,

cwin\_min\_s:long,

initial\_cwin\_s:long,

average\_rtt\_s:double,

rtt\_min\_s:double,

rtt\_max\_s:double,

stdev\_rtt\_s:double,

rtt\_count\_s:int,

ttl\_min\_s:int,

ttl\_max\_s:int,

rtx\_RTO\_s:int,

rtx\_FR\_s:int,

reordering\_s:int,

net\_dup\_s:int,

unknown\_segments\_s:int,

flow\_control\_s:int,

unnece\_rtx\_rto\_s:int,

unnece\_rtx\_fr\_s:int,

different\_syn\_seqno\_s:int,

completion\_time:double,

first\_time:double,

last\_time:double,

C\_first\_payload:double,

S\_first\_payload:double,

C\_last\_payload:double,

S\_last\_payload:double,

C\_first\_ack:double,

S\_first\_ack:double,

first\_time\_abs:double,

c\_internal:int,

s\_internal:int,

connection\_type:int,

p2p\_type:int,

p2p\_subtype:int,

ed2k\_data:int,

ed2k\_signaling:int,

ed2k\_c2s:int,

ed2k\_c2c:int,

ed2k\_chat:int,

http\_type:int,

ssl\_client\_hello:chararray,

ssl\_server\_hello:chararray,

dropbox\_id:bytearray,

fqdn:chararray);

-- Prepare the data such that input time stamp can be used accordingly to the queries

DATA = FOREACH RAW\_DATA GENERATE port\_s as port;

DATA\_GROUP\_ALL = GROUP DATA ALL;

COUNT\_ALL = FOREACH DATA\_GROUP\_ALL GENERATE COUNT\_STAR(DATA) as total;

DATA\_FILTERED = FILTER DATA BY port == 80;

DATA\_GROUP = GROUP DATA\_FILTERED ALL;

COUNT\_GROUP = FOREACH DATA\_GROUP GENERATE (double)100 \* COUNT\_STAR(DATA\_FILTERED) / COUNT\_ALL.total;

-- Store the output

STORE COUNT\_GROUP INTO '/PigData/output/ex9';

**9.2. Chạy file lưu đoạn code trên**

Pig -x Local “C:\PigData\PigCode\ex9.pig”

A screenshot of a computer screen

Description automatically generated with medium confidence

**9.3. Kiểm tra kết quả chạy**

A screenshot of a computer

Description automatically generated