# 1. Chuẩn bị file text và file scrip

## 1.1. File scrip

### scrip\_ex1.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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);  DUMP C; |

### scrip\_ex2.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 = FILTER A BY (fqdn MATCHES '.\*google\\.it.\*');  C = GROUP B BY ip\_c;  D = FOREACH C GENERATE group, COUNT(B);  DUMP D; |

### scrip\_ex3.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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, SUM(A.data\_bytes\_s);  DUMP C; |

### scrip\_ex4.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 = FOREACH A GENERATE ip\_c, data\_bytes\_s;  C = GROUP B BY ip\_c;  D = FOREACH C GENERATE group as IP, SUM(B.data\_bytes\_s) as upload;  E = FOREACH (GROUP D ALL) GENERATE TOP(100,1,D) AS top\_users;  F = FOREACH E GENERATE FLATTEN(top\_users);  G = ORDER F BY upload DESC;  Dump G; |

### scrip\_ex5.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 = foreach A generate ip\_c, data\_bytes\_s as upload;  C = group B by ip\_c;  D = foreach C generate group as ip, SUM(B.upload) as sum\_upload;  E = order D by sum\_upload DESC;  F = limit E 100;  G = join F by ip, B by ip\_c;  H = group G by ip;  I = foreach H generate group, MAX(G.upload), (double)100\*MAX(G.upload)/MAX(G.sum\_upload);  dump I; |

### scrip\_ex6.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 ALL;  C = FOREACH B GENERATE group, MIN(A.mss\_s);  DUMP C: |

### scrip\_ex7.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 = FILTER A BY (win\_min\_s == 1460);  dump B; |

### scrip\_ex8.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 ALL;  C = FOREACH B GENERATE group, COUNT(A);  D = GROUP A BY ip\_c;  E = FOREACH D GENERATE group, COUNT(A);  F = CROSS E,C;  G = FOREACH F GENERATE $0, $1, $3, ($1\*100/$3);  DUMP G; |

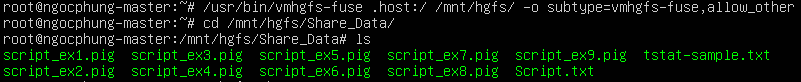
### scrip\_ex9.pig

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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);  DATA = FOREACH A GENERATE port\_c 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;  DUMP = COUNT\_GROUP; |

## 1.2. Copy file vào

Vào quyền root

|  |
| --- |
| # mkdir /mnt/hgfs |
| # /usr/bin/vmhgfs-fuse .host:/ /mnt/hgfs/ -o subtype=vmhgfs-fuse,allow\_other |
| # cd /mnt/hgfs/Share\_Data/ |



A screen shot of a computer program

Description automatically generated

# 2. Exercise 1

## 2.1 Thực hành exercise 1

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex1.pig |



A screenshot of a computer screen

Description automatically generated

## 2.2. Trả lời câu hỏi

**What does a GROUP BY command do? At which phase of MapReduce is GROUP BY performed in this exercise and in general?**

Lệnh GROUP BY được dùng để nhóm các phần tử theo một cột (thuộc tính) nào đó. Cú pháp: GROUP <đối tượng> BY <thuộc tính>.

Trong MapReduce, quá trình nhóm diễn ra ở cả hai giai đoạn: Mapping và Reducing. Trong giai đoạn Mapping, các khóa (ví dụ: ip\_c) được chỉ định để nhóm. Trong giai đoạn Reducing, dữ liệu sẽ được nhóm lại dựa trên các khóa này và có thể thực hiện các thao tác tổng hợp, như tính tổng số kết nối TCP của mỗi IP.

**What does FOREACH command do? At which phase of MapReduce is FOREACH performed in this exercise and in general?**

Lệnh FOREACH duyệt qua các dòng dữ liệu và thực hiện thao tác đã chỉ định trên từng dòng. Nó có thể được áp dụng trong cả giai đoạn Mapping và Reducing. Trong Mapping, nó xử lý các bản ghi trước khi nhóm, trong khi ở Reducing, nó xử lý dữ liệu sau khi đã được nhóm, giúp thực hiện các thao tác tổng hợp hoặc biến đổi bổ sung.

**Explain very briefly how Pig works (i.e. the process of Pig turning a Pig Latin script into runnable MapReduce job(s))**

Pig chuyển tập lệnh Pig Latin thành chuỗi công việc MapReduce qua các bước: dịch tập lệnh thành sơ đồ logic, tối ưu hóa sơ đồ logic thành sơ đồ vật lý, sau đó gửi các công việc MapReduce tới cụm Hadoop để thực hiện các chuyển đổi dữ liệu.

**Explain how you come to a conclusion that your results are correct**

Kết quả có thể được đánh giá chính xác bằng cách so sánh kết quả giữa việc chạy trên local và chế độ MapReduce, hoặc kiểm tra lại đầu ra dự kiến so với kết quả thực tế.

**How many reducers were launched? Why? Can you modify this number?**

Số reducers mặc định thường là 1, nhưng có thể thay đổi bằng lệnh SET default\_parallel <số reducers> hoặc khi nhóm dữ liệu bằng cách sử dụng parallel trong câu lệnh GROUP.

**How many mappers were launched? Why? Can you modify this number?**

Số mappers phụ thuộc vào số lượng file sau quá trình chia nhỏ (split). Số lượng mappers không cố định, có thể giảm số lượng bằng cách cấu hình pig.maxCombinedSplitSize.

# 3. Exercise 2

## 3.1 Thực hành exercise 2

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex2.pig |



A screenshot of a computer program

Description automatically generated

Không có TCP nào có fqdn chứa ‘google.it’

Thử thay đổi nội dung chuỗi dùng để match:

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 = FILTER A BY (fqdn MATCHES '.\*.sitemeter.com');  C = GROUP B BY ip\_c;  D = FOREACH C GENERATE group, COUNT(B);  DUMP D; |



## 3.2. Trả lời câu hỏi

Mặc định, Pig sử dụng **1 reducer** để xử lý các tác vụ, tuy nhiên ta có thể thay đổi số lượng reducers theo yêu cầu. Để tăng số lượng reducers mặc định, ta có thể sử dụng lệnh:

|  |
| --- |
| SET default\_parallel 20; |

Lệnh này sẽ thiết lập 20 reducers cho các tác vụ tiếp theo. Ngoài ra, ta cũng có thể xác định số lượng reducers cụ thể cho từng tiến trình bằng cách thêm từ khóa parallel vào câu lệnh, ví dụ:

|  |
| --- |
| B = GROUP A BY t PARALLEL 18; |

Lệnh này sẽ nhóm dữ liệu và sử dụng 18 reducers cho tiến trình GROUP đó.

# 4. Exercise 3

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex3.pig |



A screenshot of a computer

Description automatically generated

# 5. Exercise 4

## 5.1 Thực hành exercise 4

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex4.pig |



A screenshot of a computer screen

Description automatically generated

## 5.2. Trả lời câu hỏi

**Is this job map-only? Why? Why not?**

Job này không phải là map-only bởi ngoài các lệnh như LOAD và STORE, ta còn sử dụng các lệnh khác như GROUP, FOREACH, và GENERATE. Những lệnh này đòi hỏi reducers để thực hiện xử lý, đặc biệt là lệnh GROUP, vì vậy job này không chỉ bao gồm các thao tác map mà còn có reduce.

**Where did you apply the TOP function?**

Hàm TOP được sử dụng trong câu lệnh:

E = FOREACH (D ALL) GENERATE TOP(100,1,D) AS top\_users;

Hàm này dùng để lấy 100 bản ghi có tổng lượng dữ liệu tải lên cao nhất từ tập hợp các tuple.

**Can you explain how does the TOP function work?**

Hàm TOP nhận ba tham số: số lượng bản ghi muốn lấy (topN), chỉ số của cột dùng để sắp xếp (column), và tập dữ liệu cần chọn lọc (relation hoặc bag). Nó sẽ trả về một tập hợp chứa các bản ghi có giá trị cao nhất trong cột được chỉ định, dựa trên dữ liệu đầu vào.

**TOP function was introduced in PIG v.0.8. How, in your opinion, and based on your understanding of PIG, was the query answered before the TOP command was available? Do you think that it was less efficient than the current implementation?**

Trước khi hàm TOP được giới thiệu, người ta thường sử dụng lệnh ORDER BY để sắp xếp toàn bộ dữ liệu theo cột mong muốn, sau đó dùng LIMIT để giới hạn số lượng kết quả.

Phương pháp này kém hiệu quả hơn bởi vì ORDER BY đòi hỏi phải sắp xếp tất cả dữ liệu, rất tốn tài nguyên và thời gian, đặc biệt với tập dữ liệu lớn. Ngược lại, hàm TOP chỉ cần tìm và lấy ra một số lượng kết quả nhất định mà không phải sắp xếp toàn bộ dữ liệu, giúp tăng hiệu suất.

# 6. Exercise 5

## 6.1 Thực hành exercise 5

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex5.pig |



A screenshot of a computer program

Description automatically generated

## 6.2. Trả lời câu hỏi

**How many jobs were generated?**

Có hai job được tạo ra trong quá trình này. Job đầu tiên là khi ta thực hiện tính tổng lượng byte tải lên cho từng địa chỉ IP. Job thứ hai xảy ra khi ta thực hiện phép JOIN giữa bảng F và bảng B.

**Describe how the join is executed**

Phép JOIN (trong câu lệnh G) kết hợp dữ liệu từ hai bảng F và B dựa trên địa chỉ IP chung. Đầu tiên, các Mapper sẽ tạo ra các cặp khóa-giá trị, trong đó khóa là địa chỉ IP và giá trị là dữ liệu liên quan. Sau đó, hệ thống sẽ nhóm và sắp xếp các cặp này theo địa chỉ IP, và các bản ghi có cùng địa chỉ IP sẽ được gửi đến cùng một reducer. Trong giai đoạn reducer, các bản ghi từ bảng F và B có cùng địa chỉ IP sẽ được kết hợp lại. Sau đó, ta sẽ tính toán kích thước tải lên lớn nhất cho mỗi địa chỉ IP và tỷ lệ phần trăm của nó so với tổng kích thước tải lên.

# 7. Exercise 6

## 7.1 Thực hành exercise 6

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex6.pig |



A screenshot of a computer program

Description automatically generated

## 7.2. Trả lời câu hỏi

**Can you explain why did you find this value?**

Để xác định được giá trị nhỏ nhất của MSS, ta đã gom toàn bộ các dòng dữ liệu lại thành một nhóm duy nhất. Sau đó, ta sử dụng hàm MIN để tìm giá trị nhỏ nhất của trường MSS trong nhóm này.

**What did you learn from this exercise?**

Từ bài tập này, ta nhận thấy rằng trước khi tiến hành các truy vấn trên dữ liệu, việc xem xét dữ liệu trước là rất quan trọng. Ta nên kiểm tra để phát hiện các giá trị ngoại biên (outliers) nếu có, sau đó xử lý những giá trị này trước khi thực hiện các phép tính hoặc truy vấn phức tạp trên dữ liệu.

# 8. Exercise 7

## 8.1. Thực hành exercise 7

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex7.pig |



A screenshot of a computer screen

Description automatically generated

Chuyển số liệu win\_min\_c sang 4380 và kiểm thử

|  |
| --- |
| A = LOAD 'Apache\_Pig/Pig\_Input/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\_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, 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 = FILTER A BY (win\_min\_s == 4380);  dump B; |

A screenshot of a computer screen

Description automatically generated

## 8.2. Trả lời câu hỏi

**How many MR jobs were generated by PIG?**

Tổng cộng có 3 công việc MapReduce đã được thực hiện. Một lần khi ta nhóm tất cả các kết nối, một lần nữa khi ta nhóm các kết nối có win\_min\_c = 4380, và lần cuối cùng là khi ta thực hiện phép JOIN hai tập dữ liệu từ D và G.

**How many reducers were launched per each job? (did you use the PARALLEL keyword?)**

Mỗi công việc chỉ sử dụng một reducer, và ta không sử dụng từ khóa PARALLEL trong quá trình thực hiện.

# 9. Exercise 8

## 9.1. Thực hành exercise 8

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex8.pig |



A screenshot of a computer

Description automatically generated

## 9.1. Trả lời câu hỏi

**How many reducers were launched? Which degree of parallelism did you choose? Why?**

Trong bài tập này, ta sử dụng 2 reducer và không dùng từ khóa PARALLEL. Lý do là vì kích thước dữ liệu khá nhỏ, nên chưa cần thiết phải sử dụng PARALLEL, vì việc sử dụng nó có thể tiêu tốn nhiều tài nguyên hệ thống hơn mà không đem lại cải thiện đáng kể về hiệu suất

**If you run the exercise on a large cluster, using the same dataset, would you use a different value of parallelism?**

Nếu chạy trên một cụm lớn, chắc chắn ta sẽ cần sử dụng ít nhất một mức độ song song (PARALLEL). Tuy nhiên, con số này sẽ phụ thuộc vào sự phân bố dữ liệu, băng thông mạng, và hạ tầng hệ thống để tối ưu hóa hiệu suất.

# 10. Exercise 9

vì không có port 80 trong dữ liệu nên em đã chuyển sang 1280

## 10.1. Thực hành exercise 9

|  |
| --- |
| grunt> exec Apache\_Pig/Pig\_Input/script\_ex2.pig |



A screenshot of a computer program

Description automatically generated

## 10.2. Trả lời câu hỏi

**Using the result of this exercise and the previous one, what can you say about the distribution of server ports?**

Từ bài tập này và bài tập trước, ta có thể thấy rằng sự phân bố các cổng là không đồng đều. Cổng 1280 (cổng này được dùng như một giao thức giao tiếp cụ thể) chiếm tới 0.1% tổng số các cổng được kết nối, trong khi đó vẫn có một số cổng hầu như không được sử dụng, ví dụ như cổng 10153, 22719...