Hadoop Pig 練習

Pig 提供 Script 語言 Pig Latin，用來撰寫 MapReduce 程式。自動將腳本轉換成在 Hadoop 執行的 MapReduce Java 程式。使用者不懂 Java 也能撰寫 MapReduce。一般透過 Pig 腳本程式轉換，會比直接用 Java 撰寫 MapReduce 效能降低 25%。

0. 安裝完成 Hadoop，執行 Hadoop
1. 下載 Pig

   cd ~

   cd Downloads

   sudo wget http://ftp.mirror.tw/pub/apache/pig/pig-0.17.0/pig-0.17.0.tar.gz
2. 解壓縮

   tar -zxf pig-0.17.0.tar.gz

   sudo mv pig-0.17.0 /opt/pig
3. 新增環境變數

   #將以下資料加到 profile 最下面並儲存，sudo gedit ~/.bashrc

   export PATH=$PATH:/opt/pig/bin

   export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/pig/lib

   export PIG_HOME=/opt/pig

   export PIG_CONF_DIR=/opt/pig/conf

   export PIG_CLASSPATH=/opt/hadoop/etc/Hadoop

4. Pig 要在 MapReduce 執行，要將相關的 lib 複製到 Hadoop 的 lib 底下

   cp /opt/pig/lib/antlr-runtime-3.4.jar /opt/hadoop/share/hadoop/mapreduce

   cp /opt/pig/lib/automaton-1.11-8.jar /opt/hadoop/share/hadoop/mapreduce

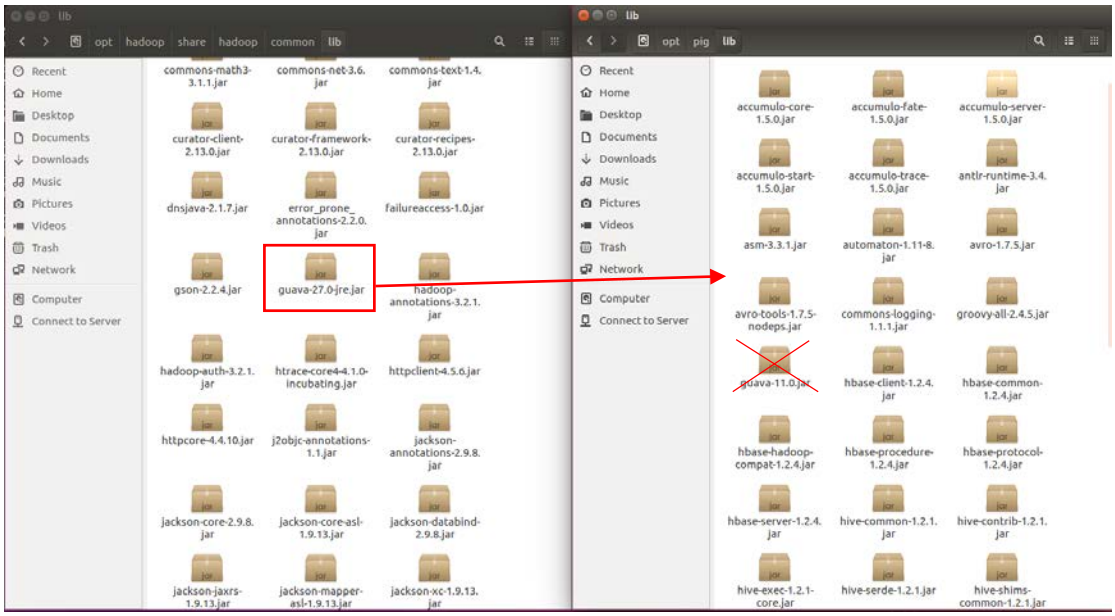
   cp /opt/pig/lib/jline-2.11.jar /opt/hadoop/share/hadoop/mapreduce

   cp /opt/pig/lib/joda-time-2.9.3.jar /opt/hadoop/share/hadoop/mapreduce

5. pig 與 hadoop 同步 guava 版本，刪除 pig 的 guava 並將 hadoop 的 guava 複製一份丟至 pig 的 lib 具體位置見下圖。

   cp /opt/hadoop/share/hadoop/common/lib/guava-27.0-jre.jar /opt/pig/lib/.

   rm /opt/pig/lib/guava-11.0.jar



6. 執行環境變數

   source /etc/profile

   source ./.bashrc

#pig 範例，Aggregation (Local Mode):

1. 下載 excite-small.log

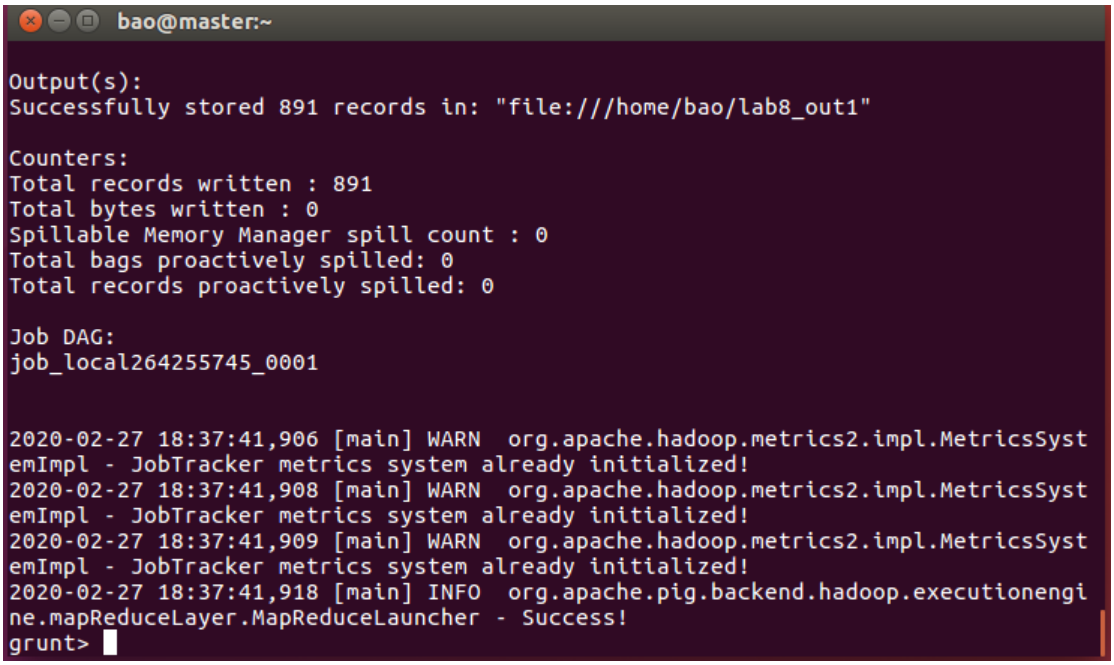
   wget http://www.hadoop.tw/excite-small.log

2. 在本機執行 pig

   pig -x local

3. Aggregation

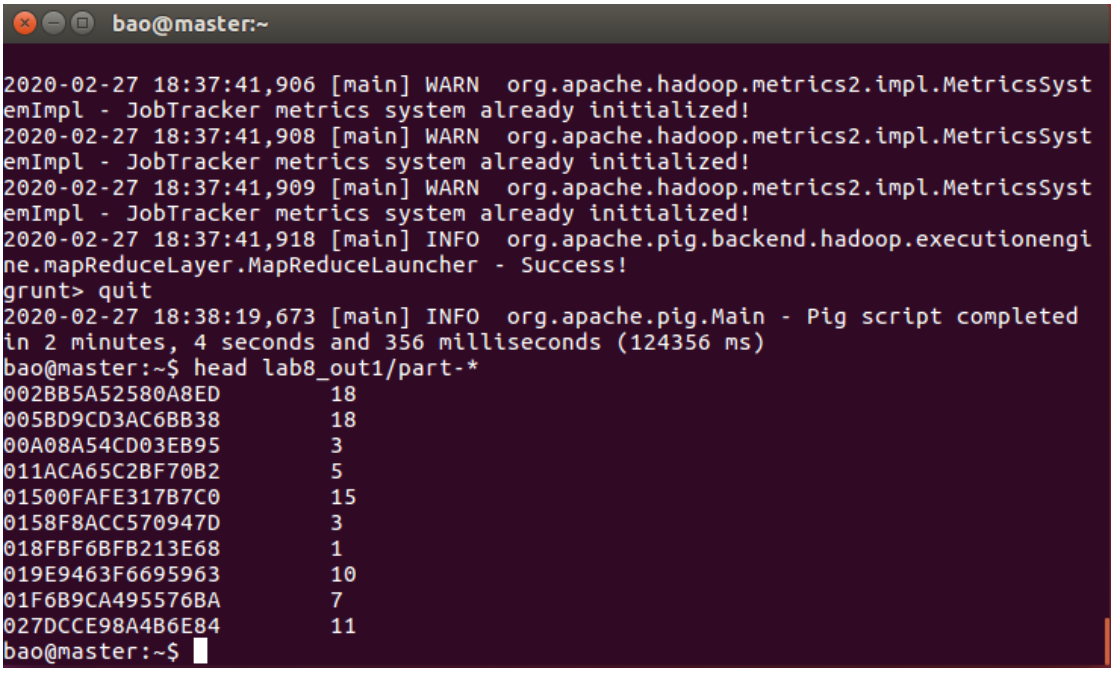
```
log = LOAD 'excite-small.log' AS (user, timestame, query);
grpd = GROUP log BY user;
cntd = FOREACH grpd GENERATE group, COUNT(log);
STORE cntd INTO 'lab8_out1';
```



quit

4. 利用 head 來查看結果(head 預設查看前面 10 筆資料)

head lab8_out1/part-*



#Filter (Local Mode)

1. 在本機執行 pig

pig -x local

2. Filter

log = LOAD 'excite-small.log' AS (user, timestame, query);

grpd = GROUP log BY user;

cntd = FOREACH grpd GENERATE group, COUNT(log) AS cnt;

fltrd = FILTER cntd BY cnt >50;

STORE fltrd INTO 'lab8_out2';

quit

3. 一樣利用 head 來查看結果(head 預設查看前面 10 筆資料)

head lab8_out2/part-*

```
bao@master:~
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1871386005_0001


2020-02-27 18:41:43,626 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSyst
emImpl - JobTracker metrics system already initialized!
2020-02-27 18:41:43,628 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSyst
emImpl - JobTracker metrics system already initialized!
2020-02-27 18:41:43,630 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSyst
emImpl - JobTracker metrics system already initialized!
2020-02-27 18:41:43,638 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
grunt> quit
2020-02-27 18:41:53,304 [main] INFO  org.apache.pig.Main - Pig script completed
in 20 seconds and 369 milliseconds (20369 ms)
bao@master:~$ head lab8_out2/part-*
0B294E3062F036C3        61
128315306CE647F6        78
7D286B5592D83BBE        59
bao@master:~$
```

#Sorting (Local Mode)

1. 在本機執行 pig

pig -x local

2. Sorting

log = LOAD 'excite-small.log' AS (user, timestame, query);

grpd = GROUP log BY user;

cntd = FOREACH grpd GENERATE group, COUNT(log) AS cnt;

fltrd = FILTER cntd BY cnt > 50;

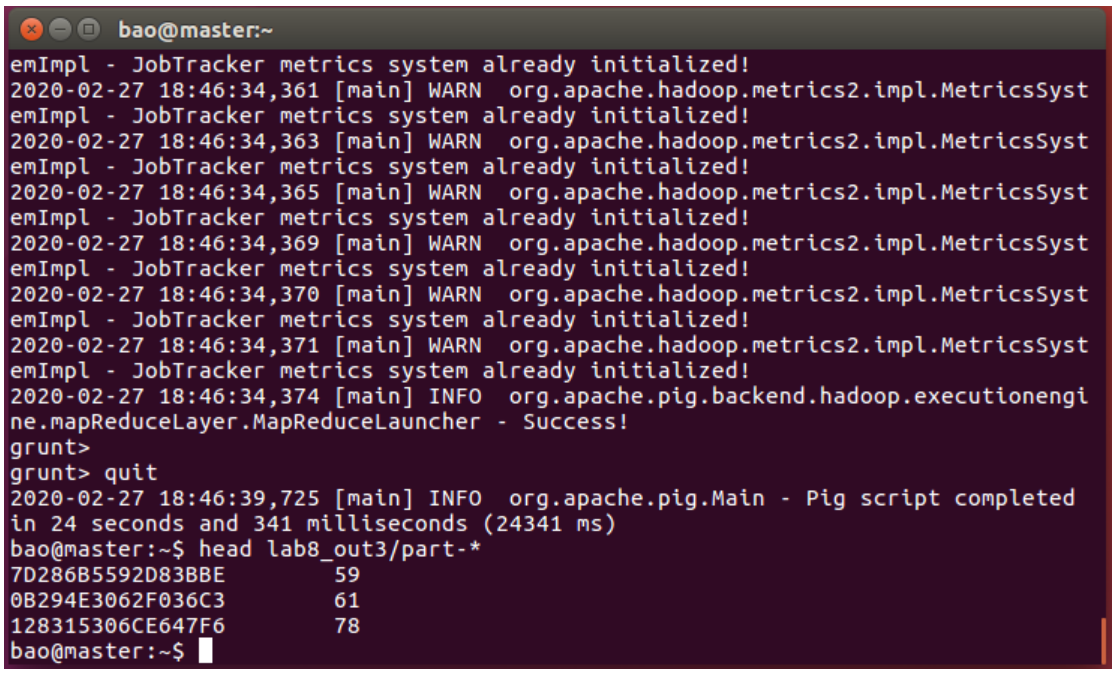
srtd = ORDER fltrd BY cnt ;

STORE srtd INTO 'lab8_out3' ;

quit

3. 一樣利用 head 來查看結果(head 預設查看前面 10 筆資料)

head lab8_out3/part-*



hadoop+pig

1. 首先先重開機,然後開啟 hadoop(為了讓 profile 檔生效),再啟動 hadoop

   source /opt/hadoop/etc/hadoop/hadoop-env.sh

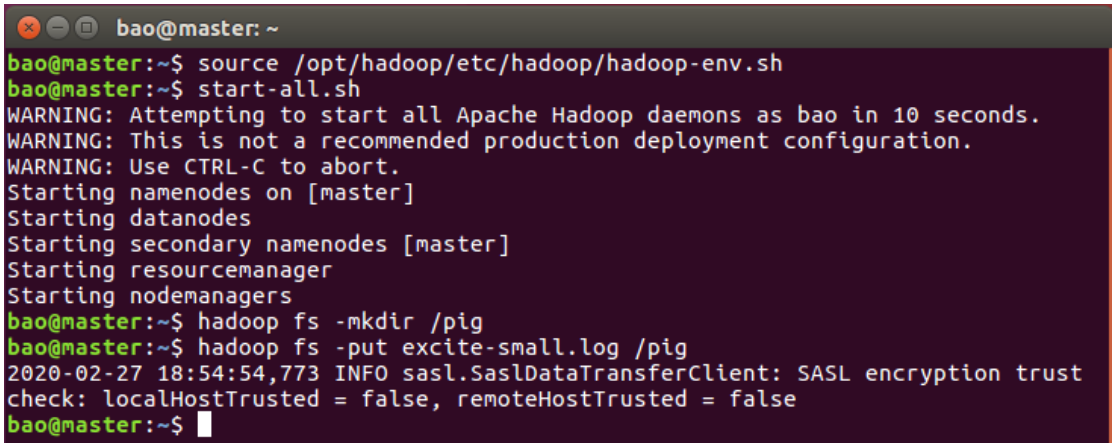
   start-all.sh

2. 在 hdfs 上建一個資料夾為 pig

   hadoop fs -mkdir /pig

3. 將檔案傳到 hdfs 上

   hadoop fs -put excite-small.log /pig



4. 執行 pig

pig

log = LOAD '/pig/excite-small.log' AS (user, timestamp, query);

grpd = GROUP log BY user;

cntd = FOREACH grpd GENERATE group, COUNT(log) AS cnt;

fltrd = FILTER cntd BY cnt > 50;

STORE fltrd INTO 'lab8_out';　　(這步驟會有點久)

quit

5. 查看結果:

hadoop fs -cat /user/XXXX/lab8_out/*

Note:XXXX 為學號

```
bao@master: ~
2020-02-27 19:01:57,453 [main] INFO  org.apache.hadoop.ipc.Client - Retrying con
nect to server: master/192.168.84.139:10020. Already tried 8 time(s); retry poli
cy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISEC
ONDS)
2020-02-27 19:01:58,455 [main] INFO  org.apache.hadoop.ipc.Client - Retrying con
nect to server: master/192.168.84.139:10020. Already tried 9 time(s); retry poli
cy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISEC
ONDS)
2020-02-27 19:01:58,556 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Unable to retrieve job to compute warning
aggregation.
2020-02-27 19:01:58,557 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
grunt>
grunt> quit
2020-02-27 19:02:03,041 [main] INFO  org.apache.pig.Main - Pig script completed
in 3 minutes, 54 seconds and 754 milliseconds (234754 ms)
bao@master:~$ hadoop fs -cat /user/bao/lab8_out/*
2020-02-27 19:02:16,432 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localHostTrusted = false, remoteHostTrusted = false
0B294E3062F036C3        61
128315306CE647F6        78
7D286B5592D83BBE        59
bao@master:~$
```

Batch 練習

The first column is name, the secord column is age, and the final one is location who lives.

1. Please Tell me after 20 years, who's age will be greater than 100.
2. Please (randomly) tell me any 3 records who live in Taipei.
3. Please give me a new table(relation) with additional column that describe whether this person is young or old.

Rule : if someones's age is less than 50, mark him as 'young', otherwise mark him as 'old'

The expected result is as followings:

Mar　　18　Taipei　　young

Jacky 72　Pingtung　old

下載資料檔案

wget http://www.cc.ntut.edu.tw/~jykuo/course/pig01_source

#編輯 pig_0101.pig

gedit pig0101.pig

寫入

```
a = LOAD 'pig01_source' AS (name:chararray, age:int, location:chararray);
b = FILTER a BY (age + 20 > 100);
DUMP b;
```

輸出為

(Shawn,89,Tauchung)

(Kimi,98,Taipei)

執行 pig0101.pig

```
pig -x local
run pig0101.pig
```

#編輯 pig_0102.pig

gedit pig0102.pig

寫入

```
a = LOAD 'pig01_source' AS (name:chararray, age:int, location:chararray);
b = FILTER a BY LOWER(location) == 'taipei';
c = LIMIT b 3;
DUMP c;
```

輸出為

(Sam,32,Taipei)

(Tom,50,Taipei)

(Mary,18,Taipei)

#編輯 pig_0103.pig

gedit pig0103.pig

寫入

```
a = LOAD 'pig01_source' AS (name:chararray, age:int, location:chararray);
b = FOREACH a GENERATE *, (age < 50? 'young' : 'old') AS description;
DUMP b;
```

輸出為

(Mary,18,Taipei,young)

(John,42,Kaohsiung,young)

(Tom,50,Taipei,old)

(Rora,12,Pingtung,young)

(Jacky,72,Pingtung,old)

(Phoenix,28,Taichung,young)
(Shawn,89,Tauchung,old)
(Sam,32,Taipei,young)
(Ivy,57,Taipei,old)
(Kimi,98,Taipei,old)

執行 pig0103.pig
pig -x local
run pig0103.pig