EECS6893 Homework 1

Zhuxi Cai zc2270

1. Learn to use PIG.

(1) Run pig in local mode:

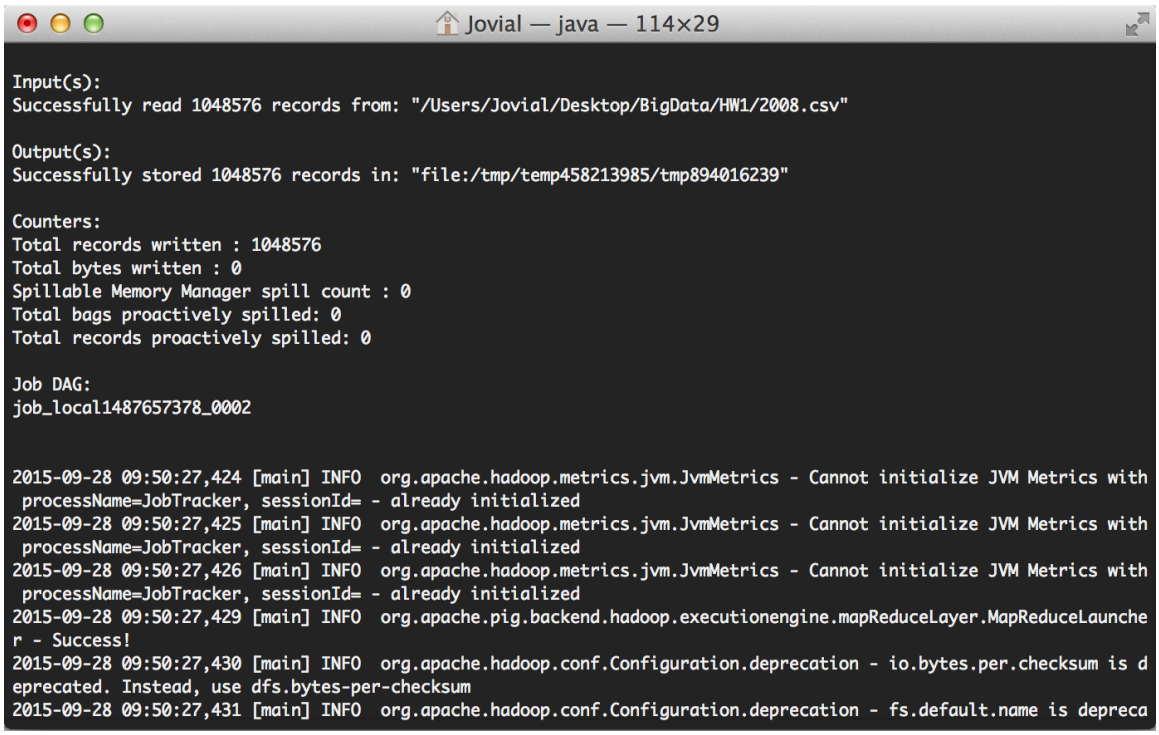
(a) Install pig and read in 2008.csv as 'airline'

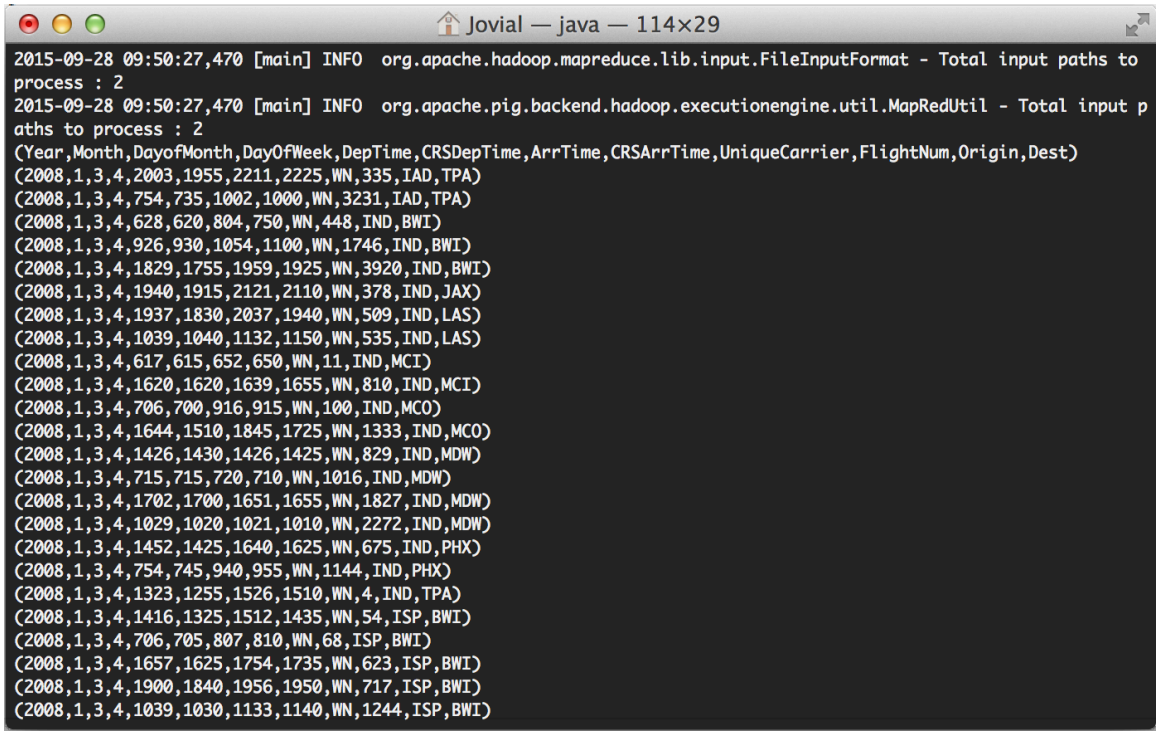
Code:

dyn-160-39-173-25:~ Jovial$ pig -x local

grunt> airline = load '/Users/Jovial/Desktop/BigData/HW1/2008.csv' using PigStorage(',') as (Year,Month,DayofMonth,DayofWeek,DepTime,CRSDepTime, ArrTime,CRSArrTime,UniqueCarr,FlightNum,Origin,Dest);

grunt> dump airline





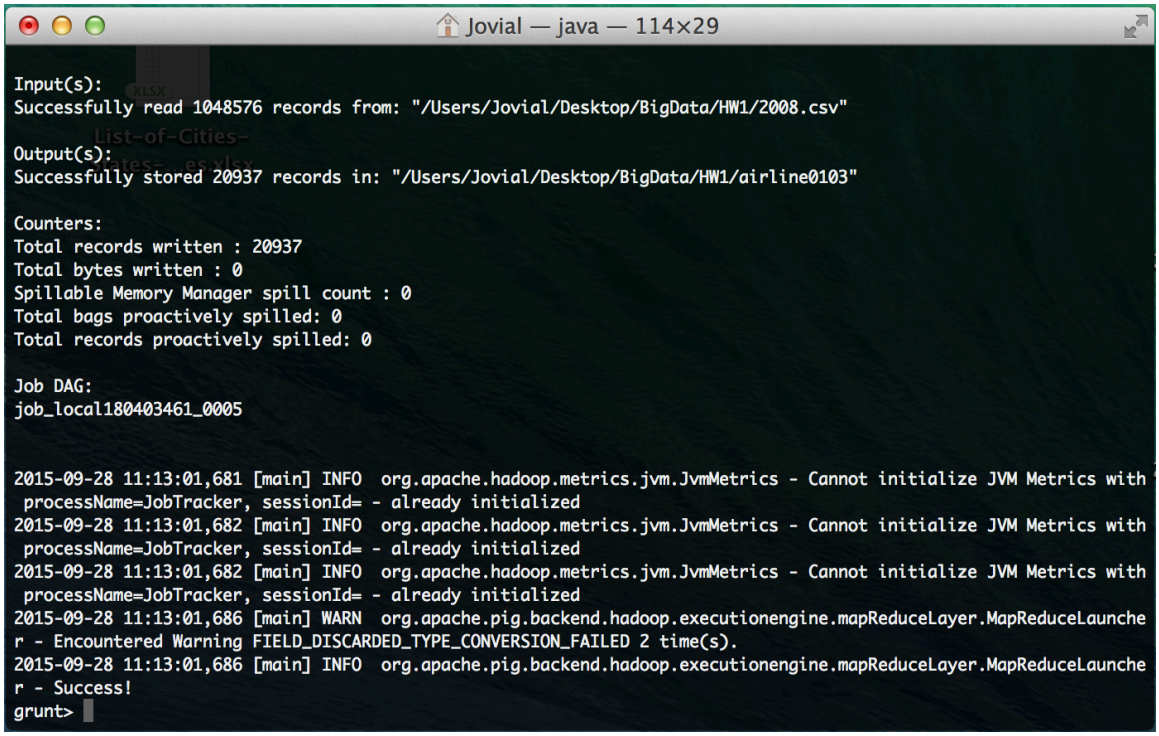
(b) Filter: select the information of airline on 3rd January

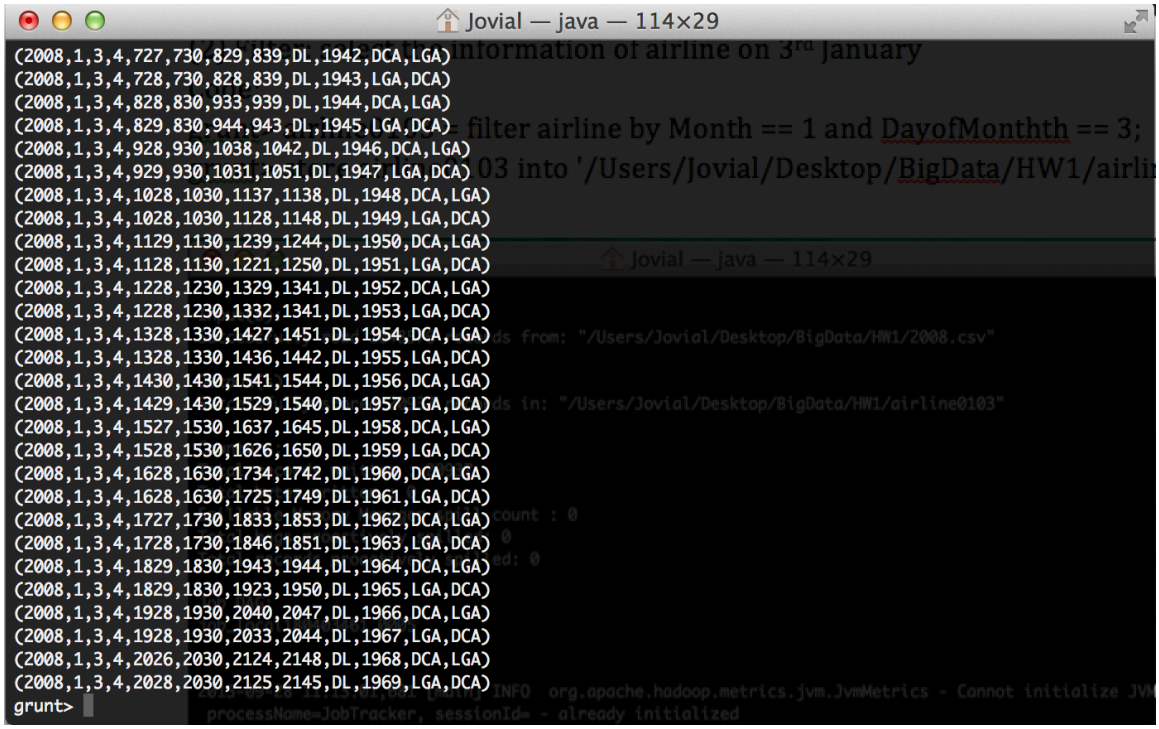
Code:

grunt> airline0103 = filter airline by Month == 1 and DayofMonthth == 3;

grunt> store airline0103 into '/Users/Jovial/Desktop/BigData/HW1/airline0103';

grunt> dump airline0103





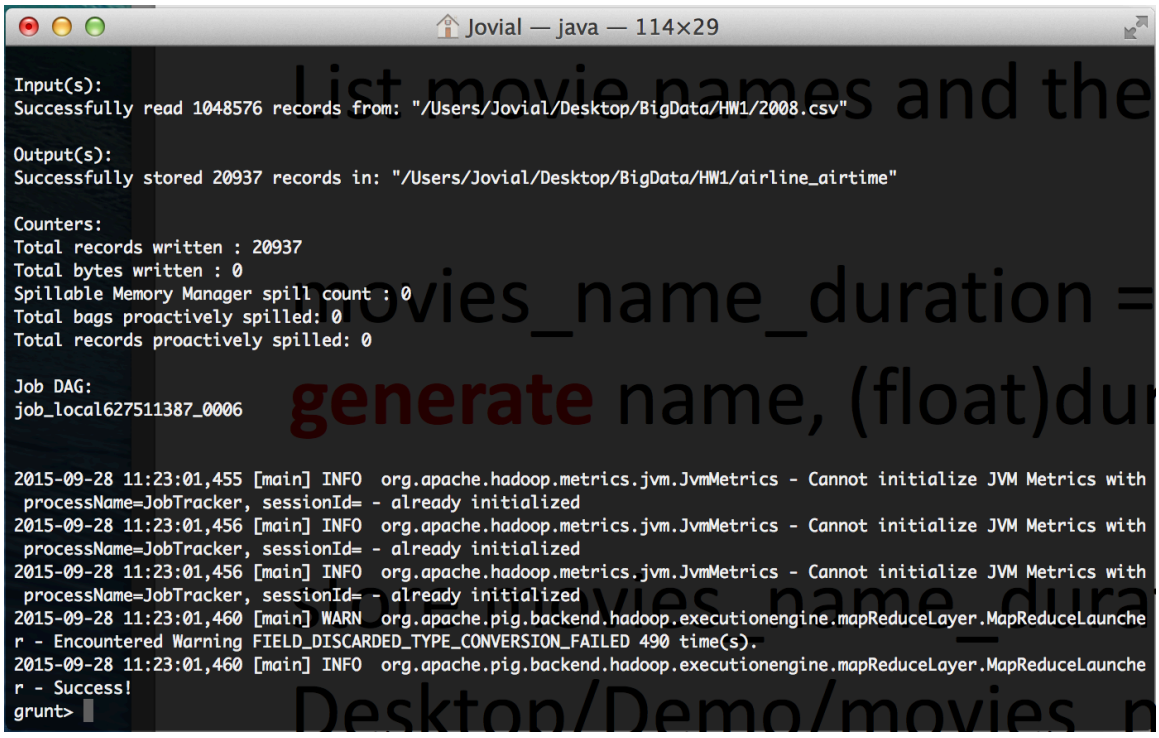
(c) Foreach Generate: list airlines on 3rd January and their airtime (in minutes)

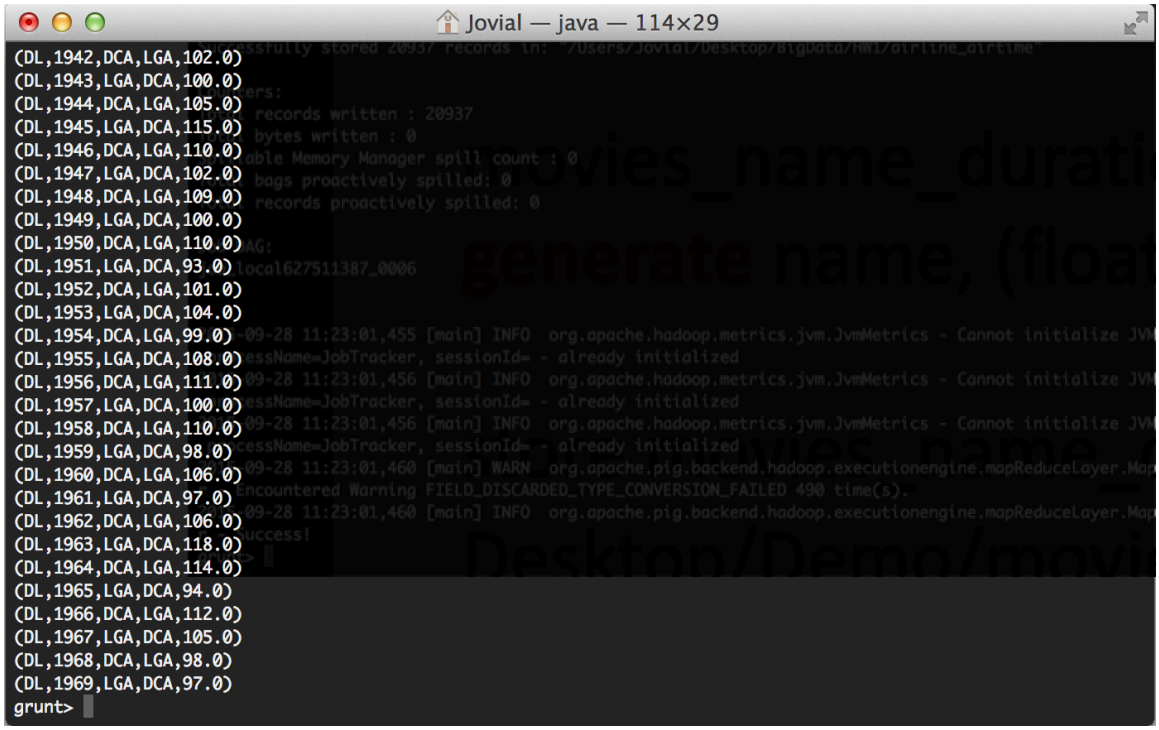
Code:

grunt> airline_airtime = foreach airline0103 generate UniqueCarr,FlightNum,Origin, Dest,(ArrTime-DepTime);

grunt> store airline_airtime into '/Users/Jovial/Desktop/BigData/HW1/airline_air time';

grunt> dump airline_airtime;


Jovial — java — 114×29
Input(s):
Successfully read 1048576 records from: "/Users/Jovial/Desktop/BigData/HW1/2008.csv"
Output(s):
Successfully stored 20937 records in: "/Users/Jovial/Desktop/BigData/HW1/airline_airtime"
Counters:
Total records written : 20937
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_local627511387_0006
2015-09-28 11:23:01,455 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2015-09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2015-09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2015-09-28 11:23:01,460 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 490 time(s).
2015-09-28 11:23:01,460 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>

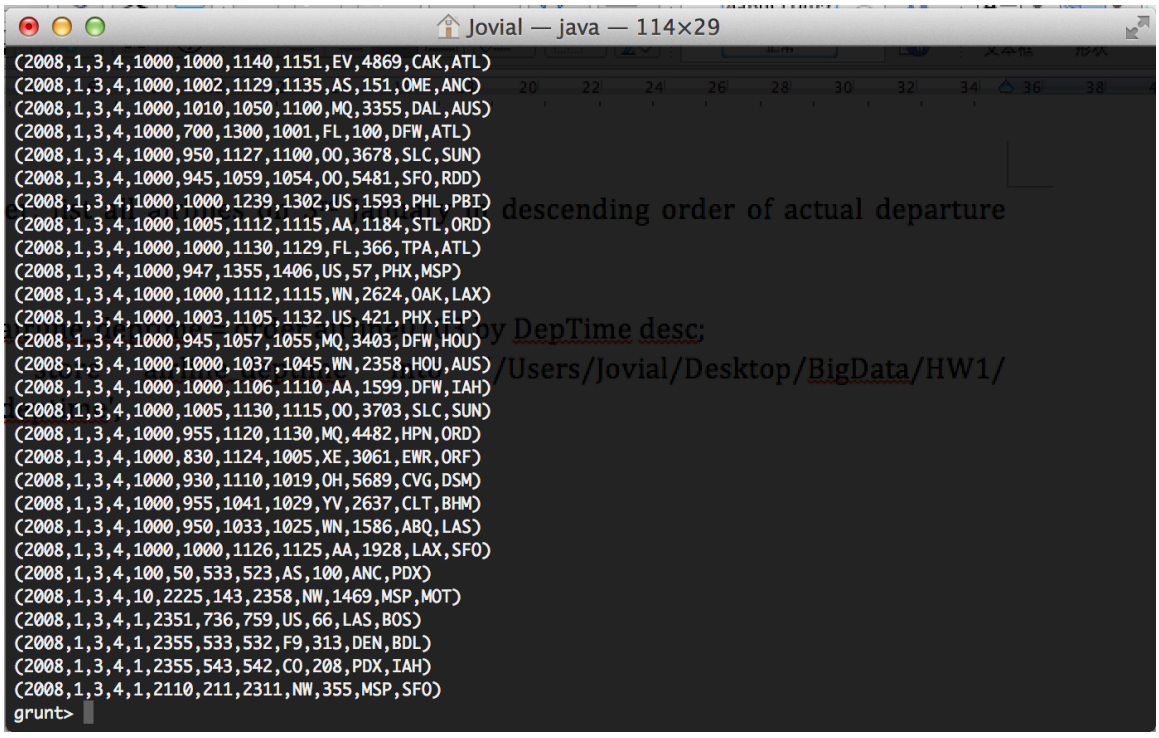

Jovial — java — 114×29
(DL,1942,DCA,LGA,102.0)
(DL,1943,LGA,DCA,100.0)
(DL,1944,DCA,LGA,105.0)
(DL,1945,LGA,DCA,115.0)
(DL,1946,DCA,LGA,110.0)
(DL,1947,LGA,DCA,102.0)
(DL,1948,DCA,LGA,109.0)
(DL,1949,LGA,DCA,100.0)
(DL,1950,DCA,LGA,110.0)
(DL,1951,LGA,DCA,93.0)
(DL,1952,DCA,LGA,101.0)
(DL,1953,LGA,DCA,104.0)
(DL,1954,DCA,LGA,99.0)
(DL,1955,LGA,DCA,108.0)
(DL,1956,DCA,LGA,111.0)
(DL,1957,LGA,DCA,100.0)
(DL,1958,DCA,LGA,110.0)
(DL,1959,LGA,DCA,98.0)
(DL,1960,DCA,LGA,106.0)
(DL,1961,LGA,DCA,97.0)
(DL,1962,DCA,LGA,106.0)
(DL,1963,LGA,DCA,118.0)
(DL,1964,DCA,LGA,114.0)
(DL,1965,LGA,DCA,94.0)
(DL,1966,DCA,LGA,112.0)
(DL,1967,LGA,DCA,105.0)
(DL,1968,DCA,LGA,98.0)
(DL,1969,LGA,DCA,97.0)
grunt>

(d) Order: list all airlines on 3rd January in descending order of actual departure time

Code:

grunt> airline_deptime = order airline0103 by DepTime desc;

grunt> store airline_deptime into '/Users/Jovial/Desktop/BigData/HW1/airline_deptime';



(2) Run pig with HDFS:

(a) Run HDFS first

Code:

dyn-160-39-173-137:~ Jovial$ cd /usr/local/Cellar/hadoop/2.7.0

dyn-160-39-173-137:2.7.0 Jovial$ ./sbin/start-dfs.sh

(b) Upload file to HDFS

Code:

dyn-160-39-173-137:2.7.0 Jovial$ ./bin/hdfs dfs -mkdir ./PigSource

dyn-160-39-173-137:2.7.0 Jovial$ ./bin/hdfs dfs -put /Users/Jovial/Desktop/BigData/HW1/2008.csv ./PigSource

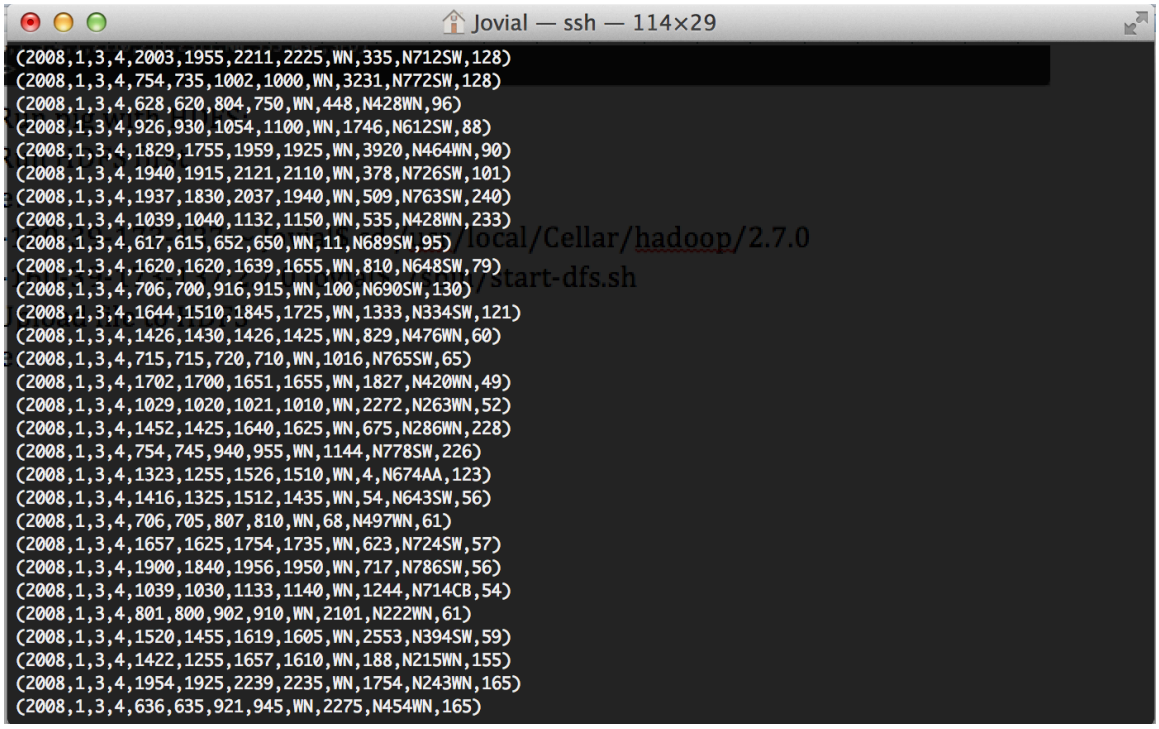
(c) Run pig in grunt with HDFS

Code:

dyn-160-39-173-137:2.7.0 Jovial$ pig

grunt> airline = load './PigSource/2008.csv' using PigStorage(',') as (Year,Month,DayofMonth,DayofWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime, UniqueCarr,FlightNum,Origin,Dest);

grunt> dump airline



(3) Summary of PIG

PIG has the huge advantage in reading and printing huge data set than Excel. In our example, we have a data set of 700 MB large. When I tried to use Excel to open this, it was really slow and my computer nearly crashed. However, I don't need to be afraid of this problem if I use PIG based on Hadoop.

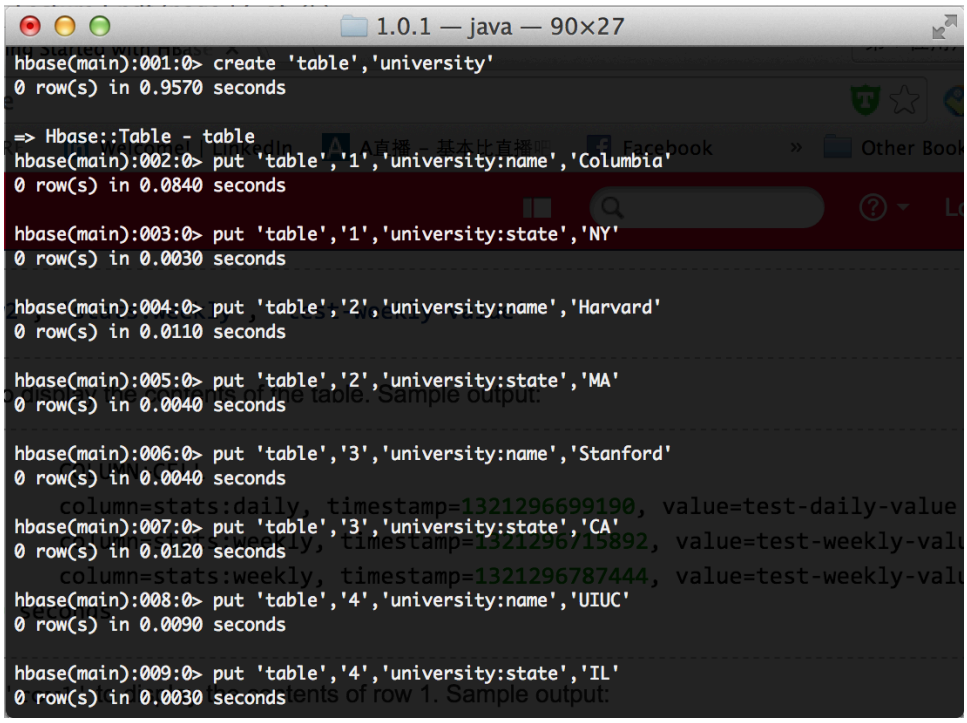
2. Try HBase. Use my own example.

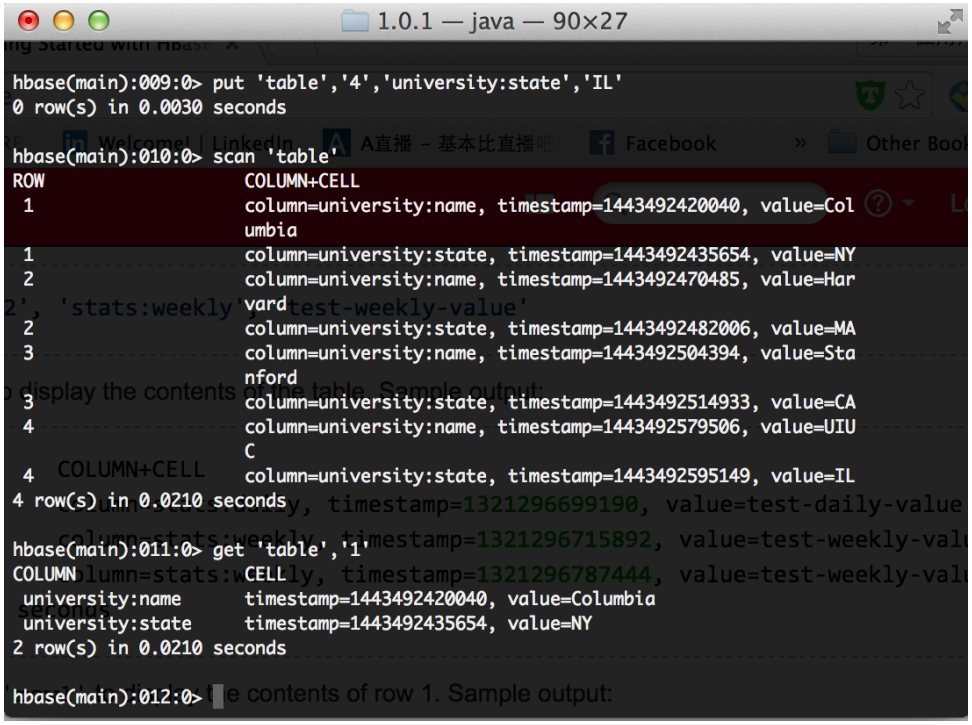
Code:

dyn-160-39-173-137:~ Jovial$ cd /usr/local/Cellar/hbase/1.0.1

dyn-160-39-173-137:1.0.1 Jovial$ ./bin/start-hbase.sh

dyn-160-39-173-137:1.0.1 Jovial$ ./bin/hbase shell





Summary of HBase:

By using HBase, I can create and apply some basic NoSQL technics in a faster and more high-technology way.

3. Try Hive. Use my own example.

(1) start Hadoop, create a file

Code:

Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -mkdir ./test

(2) put your dataset into hdfs

Code:

Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -put /Users/Jovial/Desktop/BigData/HW1/sotc-10.csv ./test

Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -ls test

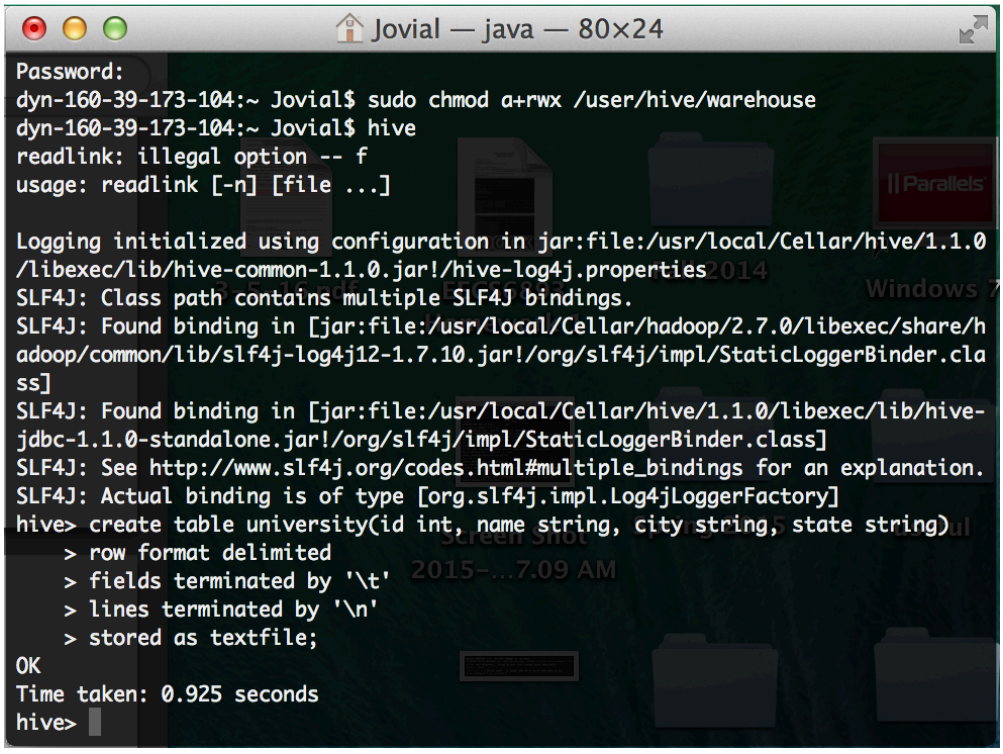
```
Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -ls test
15/09/29 09:07:00 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 Jovial staff   27782556 2015-09-29 09:06 test/sotc-10.csv
Zhuxis-MacBook-Pro:~ Jovial$
```

(3) create the table (SQL) in Hive shell

Code:

dyn-160-39-173-104:~ Jovial$ hive

hive> create table university(id int, name string, city string, state string)
    > row format delimited
    > fields terminated by '\t'
    > lines terminated by '\n'
    > stored as textfile;



(4) store the data into table
Code:
hive> load data inpath '/Users/Jovial/test/University.txt' overwrite into table university;
hive> select * from university;

```
hive> load data inpath '/Users/Jovial/test/University.txt' overwrite into table
university;
Loading data to table default.university
chgrp: changing ownership of 'file:///user/hive/warehouse/university/University.
txt': chown: you are not a member of group wheel
Table default.university stats: [numFiles=1, numRows=0, totalSize=123, rawDataSi
ze=0]
OK
Time taken: 0.149 seconds
hive> select * from university;
OK
1       Harvard         Boston
2       Columbia        NewYork NY
3       Stanford        SanFrancisco    CA
4       Berkeley        Oakland CA
5       Chicago         Chicago
Time taken: 0.054 seconds, Fetched: 5 row(s)
```

Summary of Hive:

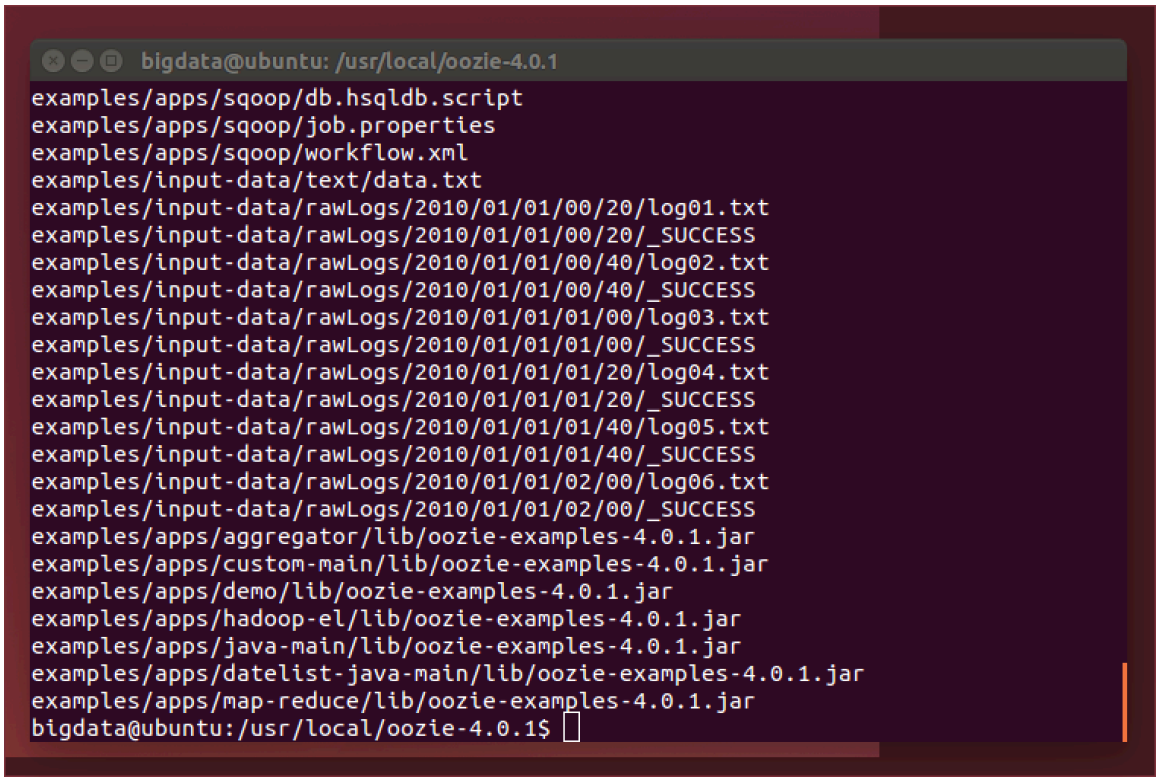
I think it is a very good RDBS based on Hadoop and much quicker than local SQL. However the installation and configure is much more complicated than PIG and HBase. I spend a lot of time configuring Hive, especially in making the directory /user/hive/warehouse and getting the permission to create a new table. At last, I used command 'sudo' to fix this problem. I don't really have better idea.

4. Oozie installation.

(1) start Oozie and check Ooize running status
Code:
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ cd /usr/local/oozie-4.0.1
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozied.sh start
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozie admin -oozie http://localhost:11000/oozie –status

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozie admin -oozie http://localhost
:11000/oozie -status
System mode: NORMAL
bigdata@ubuntu:/usr/local/oozie-4.0.1$
```

(2) untar Oozie example
Code:
bigdata@ubuntu:/usr/local/oozie-4.0.1$ cd /usr/local/oozie-4.0.1
bigdata@ubuntu:/usr/local/oozie-4.0.1$ tar -zxvf oozie-examples.tar.gz



(3) change namenode port number from 8020 to 9000 and change jobtracker port number from 8021 to 8088

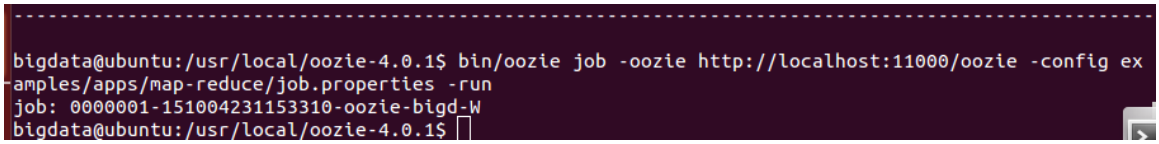
Code:

bigdata@ubuntu:/usr/local/oozie-4.0.1$ cd ./examples

bigdata@ubuntu:/usr/local/oozie-4.0.1/examples$ find ./ -type f -exec sed -i -e 's/8020/9000/g' {} \;

bigdata@ubuntu:/usr/local/oozie-4.0.1/examples$ find ./ -type f -exec sed -i -e 's/8021/8088/g' {} \;

(4) submit a job to Oozie



(5) check the job status

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ bin/oozie job -oozie http://localhost:11000/oozie -info 0
00-151004231153310-oozie-bigd-W
Job ID : 0000000-151004231153310-oozie-bigd-W
------------------------------------------------------------------------------------------------
--------------------------------
Workflow Name : map-reduce-wf
App Path      : hdfs://localhost:9000/user/bigdata/examples/apps/map-reduce
Status        : RUNNING
Run           : 0
User          : bigdata
Group         : -
Created       : 2015-10-05 07:08 GMT
Started       : 2015-10-05 07:08 GMT
Last Modified : 2015-10-05 07:08 GMT
Ended         : -
CoordAction ID: -

Actions
------------------------------------------------------------------------------------------------
--------------------------------
ID                                                                            Status    Ext ID
           Ext Status Err Code
------------------------------------------------------------------------------------------------
--------------------------------
0000000-151004231153310-oozie-bigd-W@:start:                                  OK        -
           OK         -
------------------------------------------------------------------------------------------------
--------------------------------
0000000-151004231153310-oozie-bigd-W@mr-node                                  PREP      -
           -          -
------------------------------------------------------------------------------------------------
--------------------------------

bigdata@ubuntu:/usr/local/oozie-4.0.1$
```

Summary of Oozie:

Good to learn a lot on virtual machine and Linux environment. Now I am more familiar with Linux command lines and Oozie.