INFS3200 / INFS7907 Practice 3

Cloud Computing – Hive

(3% Due Week 13 Practical Sessions)

Learning objectives:

1. Learn the principle of MapReduce
2. Learn how to use Hive
3. Get familiar with basic *nix command lines

Notation in this practical:

- **$HIVE_HOME** - the home directory of your Hive project, or known as Hive home
- **#** - for comment
- ***nix** - either linux or unix operating systems
- **$** - the command line starting position of the console/terminal
- Be aware of **case-sensitivity** when you use *nix commands

Introduction

Hive is a Hadoop-based data warehouse tool that enables easy data summarization, ad-hoc querying and analysis of large volumes of data. Hive defines a simple SQL-like query language, called HQL that enables users familiar with SQL to query the data.[1]

In our lab, we can access the ITEE ***moss*** server via putty (You can find it from the Start menu). When you open it, the screen diplay is like that in Figure 1.

Moss Server addr: remote.labs.eait.uq.edu.au

Then type your student account and password, it should allow you to login. (a student account should start with ***s***, e.g. ***s4000888***).

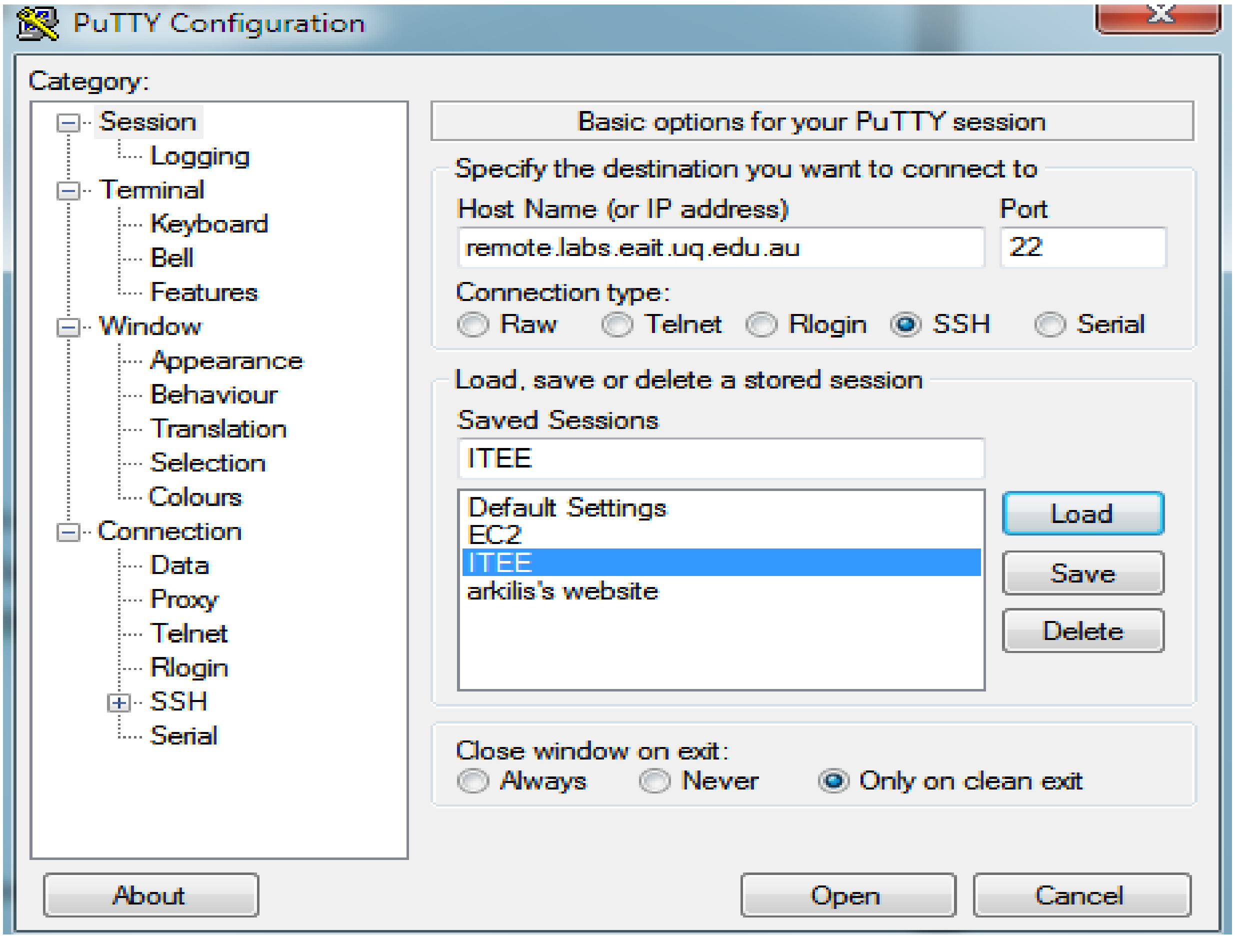
After login:

Note that all the operations are conducted through the command line.

Within the main folder of *hive*, there are many subfolders. You can list them using command "***ls***". Some folders are:

- bin: stores the executable programs of hive
- examples: some official exmaples
- conf: confiugration files

[1] https://cwiki.apache.org/confluence/display/Hive/Home



Specify the destination you want to connect to

Host Name (or IP address): remote.labs.eait.uq.edu.au

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: ITEE

Default Settings
EC2
ITEE
arkilis's website

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 1. Screen display when open moss server via puTTy.

Steps of this Exercise

You need to complete and demonstrate the following steps successfully.

1. Run hive

Hive should be started before any operation command can be used. Firstly, change to the *hive* folder:

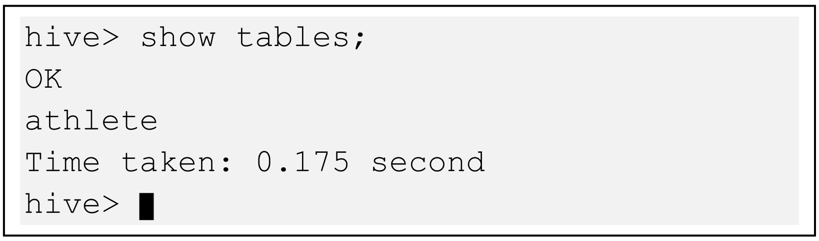
```
cd /
cd opt/local/stow/hive-0.9.0
cd bin
./hive
```

2. Create a table in Hive.

Then, you need to create a table under Hive with the following command line:

```
create table athlete (
  AthleteID int,
  FirstName string,
  LastName string,
  DOB string,
  Gender string,
  Country string)
  row format delimited
  Fields terminated by ','
  Stored as textfile;
```

After the table is created, check whether it is successfully built using the command line as shown in Figure 2.



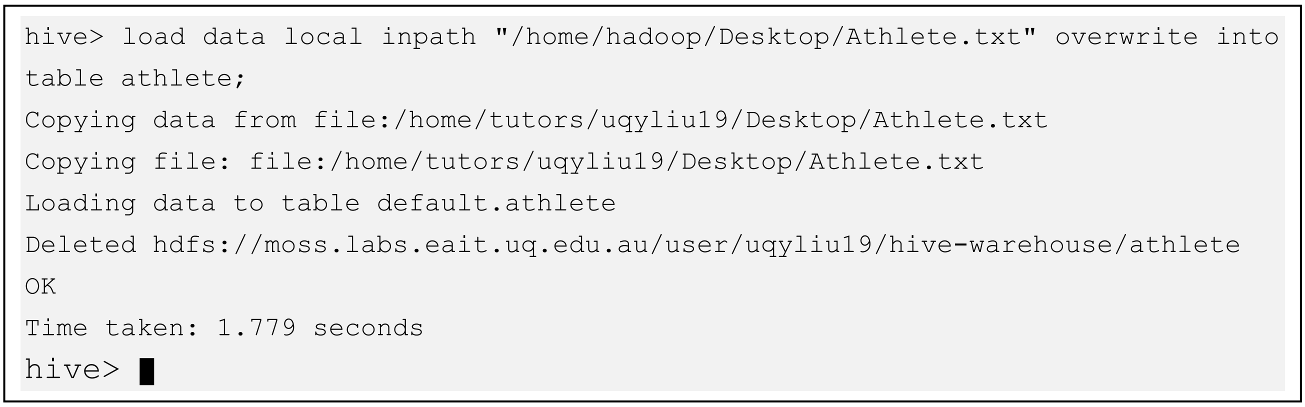
```
hive> show tables;
OK
athlete
Time taken: 0.175 second
hive> █
```

Figure 2. Screen display for checking the table athlete if it is created successfully.

After you have created a table in Hive, you need to import data from an external file named `Athletes.txt`, which is provided in the package of Practical 3. When you download `Athletes.txt`, you can store it in directory `/home/hadoop/Desktop/`. Then use the following (underlined) command to import `Athletes.txt` to Hive:

```
hive> load data local inpath "/home/hadoop/Desktop/Athlete.txt"
overwrite into table athlete;
```

The screen display of the running example is shown in Figure 3.
(Note: you should use your own path storing of `Athletes.txt` to replace the string: `"/home/hadoop/Desktop/Athlete.txt"`).



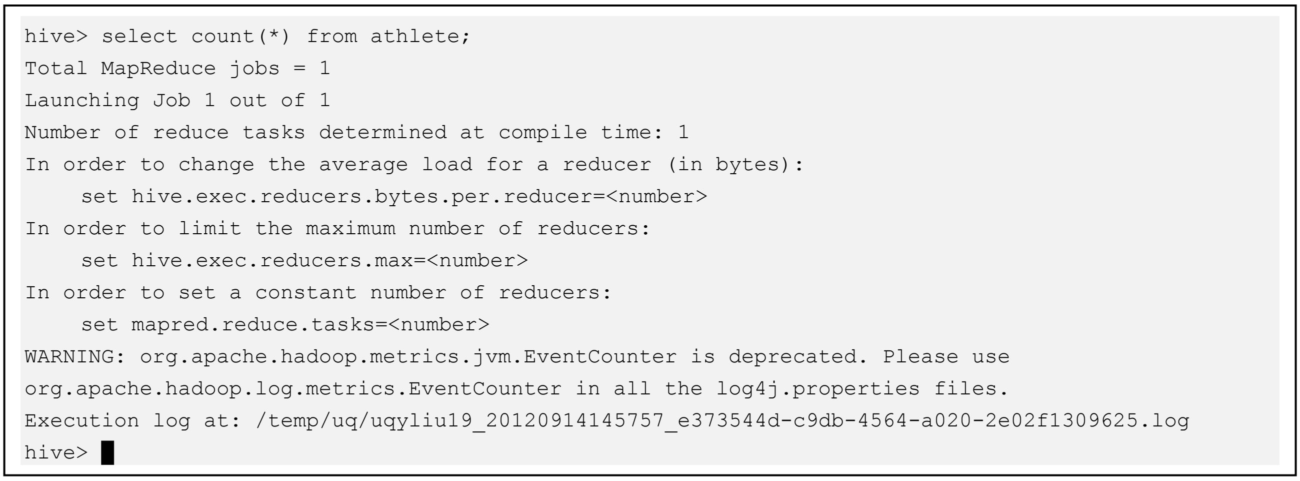
```
hive> load data local inpath "/home/hadoop/Desktop/Athlete.txt" overwrite into
table athlete;
Copying data from file:/home/tutors/uqyliu19/Desktop/Athlete.txt
Copying file: file:/home/tutors/uqyliu19/Desktop/Athlete.txt
Loading data to table default.athlete
Deleted hdfs://moss.labs.eait.uq.edu.au/user/uqyliu19/hive-warehouse/athlete
OK
Time taken: 1.779 seconds
hive> █
```

Figure 3. Screen display of data loading to athlete table in hive.

3. Operation using HiveQL statement: Count the number of rows in table athlete

```
Select count(*) from athlete;
```

Hive will use MapReduce to get the number of the rows.   See Figure 4.



```
hive> select count(*) from athlete;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
    set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
    set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
    set mapred.reduce.tasks=<number>
WARNING: org.apache.hadoop.metrics.jvm.EventCounter is deprecated. Please use
org.apache.hadoop.log.metrics.EventCounter in all the log4j.properties files.
Execution log at: /temp/uq/uqyliu19_20120914145757_e373544d-c9db-4564-a020-2e02f1309625.log
hive> █
```

Figure 4. Screen display of HiveQL statement executed based on MapReduce.

4. Create a new table with the names of all athletes and their ages in 2012.

The following statement is used to create the new table `new_athlete`.

```
create table new_athlete (
  AthleteID int,
  FirstName string,
  LastName string,
  Age int)
  row format delimited
  Fields terminated by '\t’;
```

Also, we need to create a python mapper to help us transform `date-of-birth` to `age` in 2012. You can find the code named `Mapper.py` in the package. Copy the `Mapper.py` to the Hive home folder and then add the `Mapper.py` as the source, see the screen display in Figure 5. Command:

```
add file /home/tutors/uqyliu19/Desktop/Mapper.py;
```

```
hive> add file /home/tutors/uqyliu19/Desktop/Mapper.py;
Added resource: /home/tutors/uqyliu19/Desktop/Mapper.py
hive> █
```

Figure 5. Screen display of adding a python program to hive home folder.

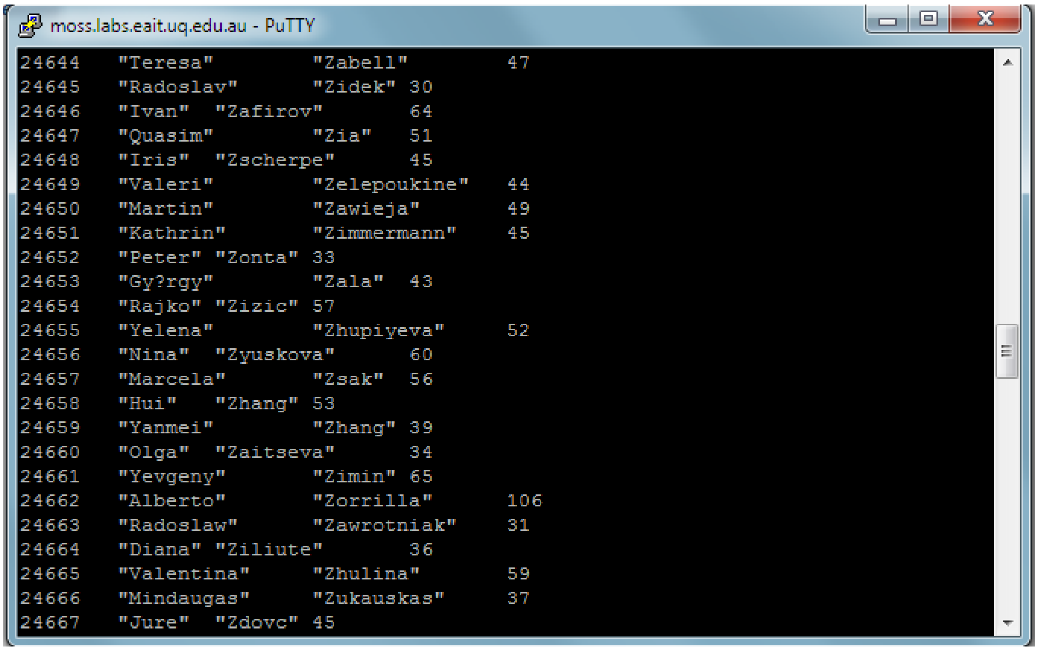
Now, import data from table `athlete` into table `new_athlete`:

```
insert overwrite table new_athlete
select transform (AthleteID, FirstName, LastName, DOB, Gender, Country)
using 'python /home/tutors/uqyliu19/Desktop/Mapper.py'
as (AthleteID, FirstName, LastName, Age)
From athlete;
```

Finally, display the imported data in table `new_athlete`, run command:

```
Select count(*) from new_athlete;
```

The result similar to Figure 6 will be shown.



```
24644   "Teresa"        "Zabell"        47
24645   "Radoslav"      "Zidek" 30
24646   "Ivan"  "Zafirov"       64
24647   "Quasim"        "Zia"   51
24648   "Iris"  "Zscherpe"      45
24649   "Valeri"        "Zelepoukine"   44
24650   "Martin"        "Zawieja"       49
24651   "Kathrin"       "Zimmermann"    45
24652   "Peter" "Zonta" 33
24653   "Gy?rgy"        "Zala"  43
24654   "Rajko" "Zizic" 57
24655   "Yelena"        "Zhupiyeva"     52
24656   "Nina"  "Zyuskova"      60
24657   "Marcela"       "Zsak"  56
24658   "Hui"   "Zhang" 53
24659   "Yanmei"        "Zhang" 39
24660   "Olga"  "Zaitseva"      34
24661   "Yevgeny"       "Zimin" 65
24662   "Alberto"       "Zorrilla"      106
24663   "Radoslaw"      "Zawrotniak"    31
24664   "Diana" "Ziliute"       36
24665   "Valentina"     "Zhulina"       59
24666   "Mindaugas"     "Zukauskas"     37
24667   "Jure"  "Zdovc" 45
```

Figure 6. The result records in table `new_athlete`.