CSE4334/5334 Data Mining

Spring 2015, Prof. Chengkai Li

Department of Computer Science and Engineering

University of Texas at Arlington

Programming Assignment 3

In this assignment, you will write MapReduce programs in Python. You will use Apache Hadoop, an open-source implementation of Google's proprietary MapReduce system. Since we don't have a cluster to use for this course, you will set up a single-node Hadoop environment on your own personal computer. The programs you write will work in a real cluster. It is just that you won't be able to observe performance advantage against solving the problems in a centralized system. In fact, you will observe worse execution efficiency, since the overhead of Hadoop environment cannot pay off in a single-node setup.

Setting up your own Hadoop environment can be non-trivial, even if it has only one node. To avoid the hassle, we will use a virtual machine made ready by Hortonworks. The virtualization software we will use is VMware. The Hadoop virtual machine we will use is Hortonworks Sandbox. There are other virtualization software and Hadoop virtual machines. Our following discussion is based on the setup of VMware Player + Hortonworks Sandbox.

You need at least 15GB free space on your hard drive.

1 Setting up the Environment

1.1 Enable BIOS Support for Virtualization

Here is an example of how to do it: http://bit.ly/1ncbAqk

This probably is only necessary if you have a PC. It appears to be irrelevant to Mac, but I couldn't verify. If you encounter troubles using an Mac, this page might have some useful information for you: http://bit.ly/1BZoaKe.

1.2 Download and Install VMware Player 7 from http://vmw.re/1J8x5At

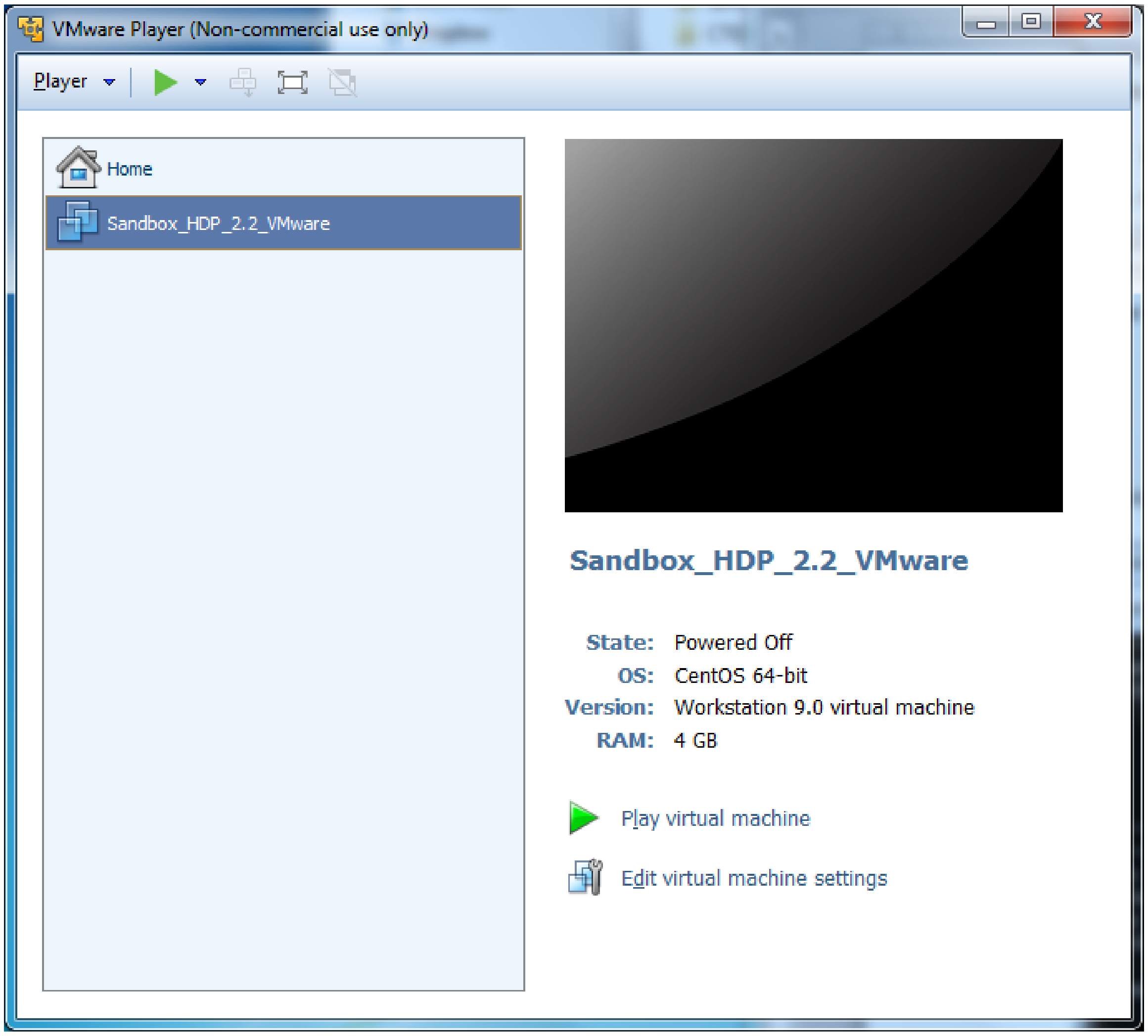
VMware Player 7 is what I used and tested for this assignment. Other versions may work as well. For instance, if you have an Mac, it seems that you need to use VMware Fusion.

1.3 Download Hortonworks Sandbox HDP 2.2 from http://bit.ly/19xTc4B

According to this page, this virtual machine should work on 32-bit and 64-bit OS (Windows XP, Windows 7, Windows 8 and Mac OSX).

1.4 Install Hortonworks Sandbox HDP 2.2 in VMware

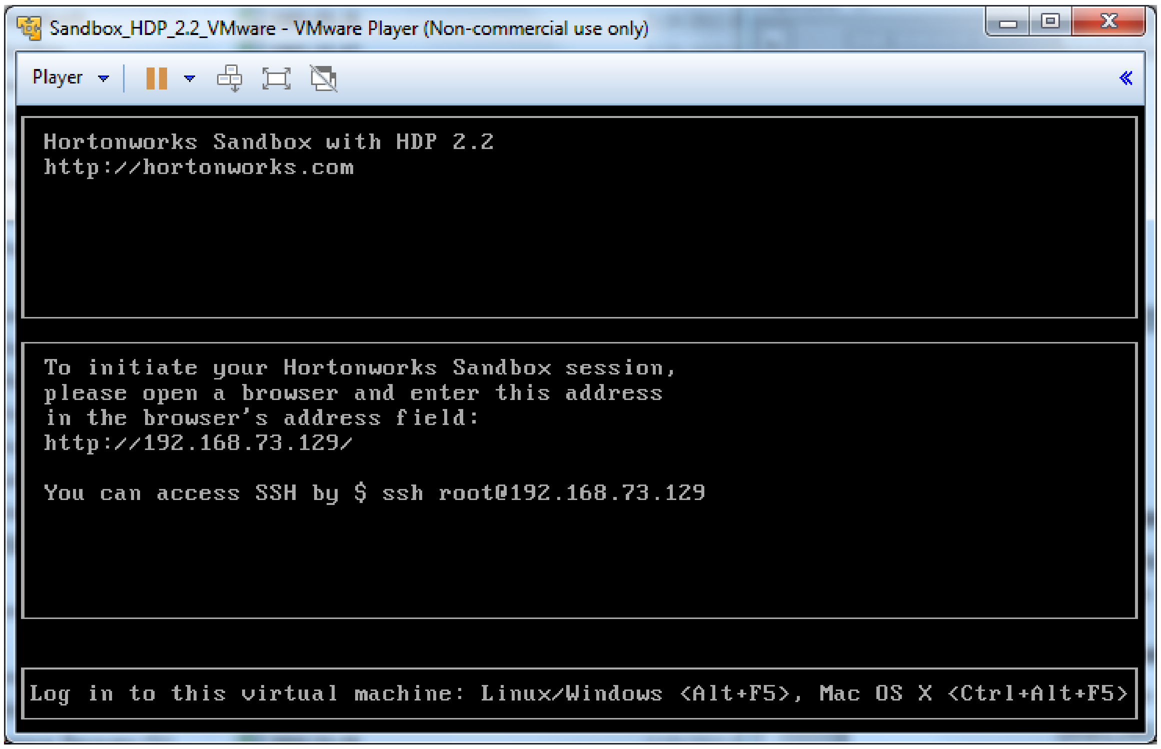
From the above page, you can see that, for VMware, they only have "Install Guide" for Mac, at http://bit.ly/19tJfpK. But the steps for Windows are pretty similar. You don't really need a guide. Just "Open" the file Sandbox_HDP_2.2_VMware.ova in VMware. After a while, you should be able to see the following screenshot:


VMware Player (Non-commercial use only)
Player
Home
Sandbox_HDP_2.2_VMware
Play virtual machine
Edit virtual machine settings

Sandbox_HDP_2.2_VMware

State: Powered Off
OS: CentOS 64-bit
Version: Workstation 9.0 virtual machine
RAM: 4 GB

The virtual machine with Hadoop environment is ready. Just click the Play (Power on) button or click "Play virtual machine" to turn on the guest OS. After a while, you should see the following screenshot:


Sandbox_HDP_2.2_VMware - VMware Player (Non-commercial use only)
Player

```
Hortonworks Sandbox with HDP 2.2
http://hortonworks.com

To initiate your Hortonworks Sandbox session,
please open a browser and enter this address
in the browser's address field:
http://192.168.73.129/

You can access SSH by $ ssh root@192.168.73.129

Log in to this virtual machine: Linux/Windows <Alt+F5>, Mac OS X <Ctrl+Alt+F5>
```

2 Data Files

In this assignment, we will use a set of 30 files that contain the transcripts of all presidential debates in history. Each file is for one debate. The files' suffix is .csv. But they are not really comma separated. The fields in the files are separated by "^". More about this later when you need to take care of it.

The 30 files are provided to you in a ZIP file that can be downloaded from the Programming Assignment 3 (P3) entry in Blackboard.

3 Test Example

To verify the environment is working, we will run the classic Word Count example.

3.1 Transfer data files into the local file system of the guest OS

Use SSH and SFTP tools to upload all 30 files of presidential debate transcripts into the guest OS with the Hadoop environment. Login information as follows:

IP address: 192.168.73.129

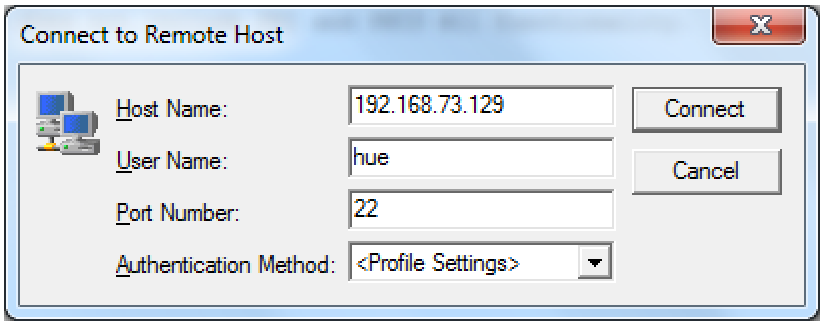
Username: hue

Password: hadoop

We now explain the detailed steps of doing it in Windows, using SSH Secure Shell Client and SSH Secure File Transfer which can be downloaded from http://www.uta.edu/oit/cs/unix/ssh/Secure-Shell-Client.php. There are similar tools on Mac and Unix/Linux. And you can also use command line.

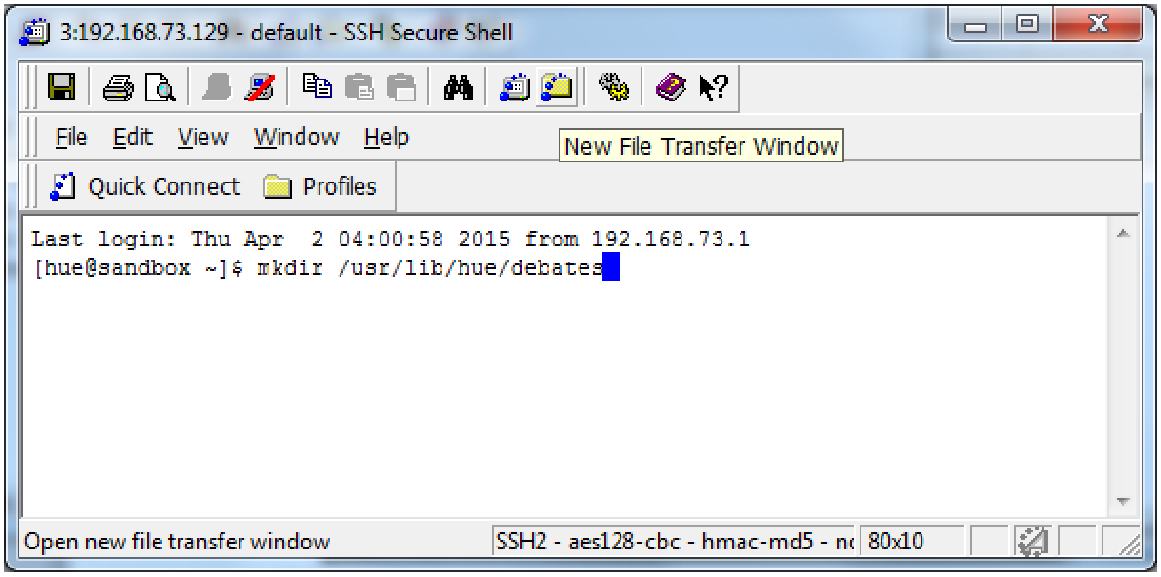
(a) Login to the guest OS using SSH Secure Shell Client. Below is the screenshot of my SSH Secure Shell Client that is about to the make the connection.

Below is the screenshot of my SSH Secure File Transfer that is about to connect to the guest OS.



Host Name: 192.168.73.129
User Name: hue
Port Number: 22
Authentication Method: <Profile Settings>

After I press "connect" and enter "hadoop" as the password, I get the following screenshot:


3:192.168.73.129 - default - SSH Secure Shell
File
Edit
View
Window
Help
New File Transfer Window
Quick Connect
Profiles
Open new file transfer window
SSH2 - aes128-cbc - hmac-md5 - n
80x10

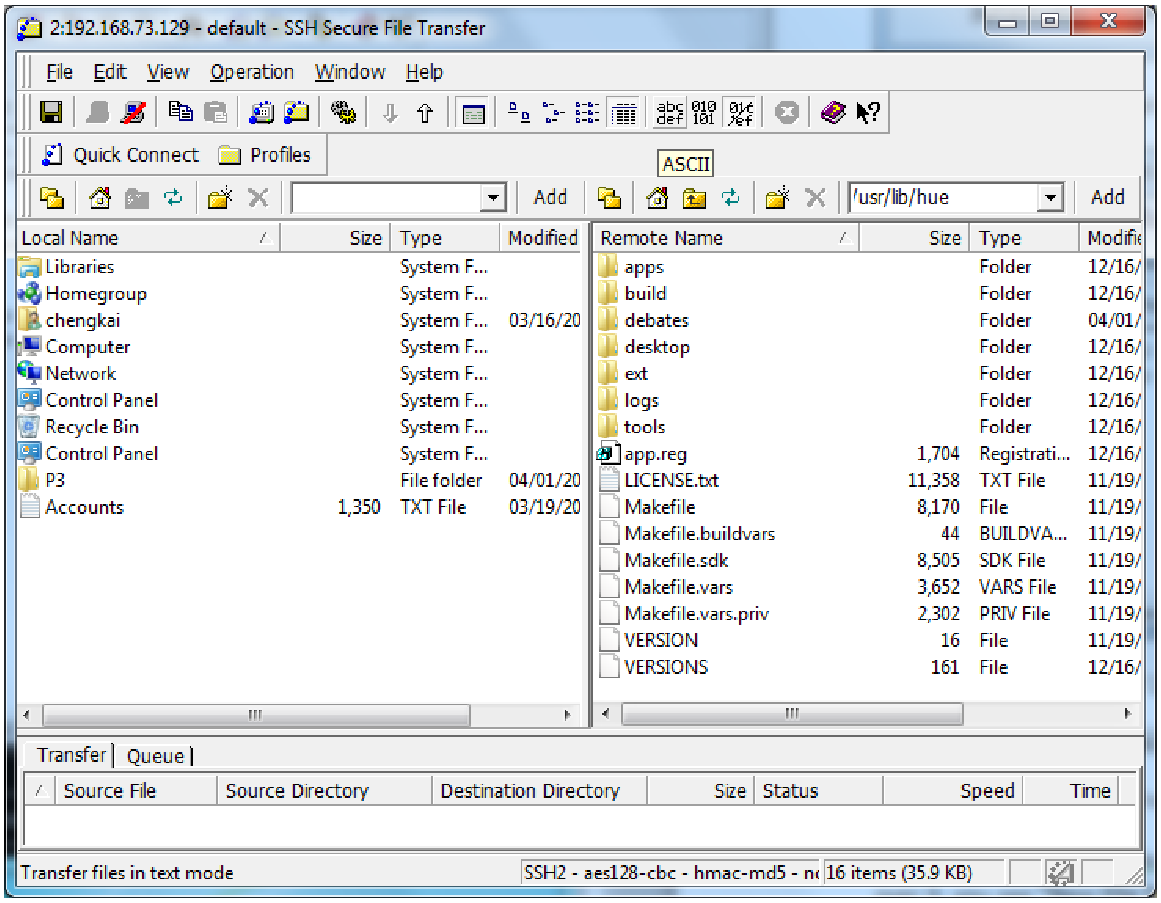
```
Last login: Thu Apr  2 04:00:58 2015 from 192.168.73.1
[hue@sandbox ~]$ mkdir /usr/lib/hue/debates
```

(b) Create a folder `/usr/lib/hue/debates` by typing the following command:

```
mkdir /usr/lib/hue/debates
```

(c) Open SSH Secure File Transfer.

Notice the "SSH Secure File Transfer" icon in the above screenshot? (When you move your mouse over it, you see "New File Transfer Window". Click it, you will get the following window:


2:192.168.73.129 - default - SSH Secure File Transfer
File Edit View Operation Window Help
Quick Connect
Profiles
ASCII
Add
/usr/lib/hue
Add
Transfer
Queue

Local Name	Size	Type	Modified
Libraries		System F...	
Homegroup		System F...	
chengkai		System F...	03/16/20
Computer		System F...	
Network		System F...	
Control Panel		System F...	
Recycle Bin		System F...	
Control Panel		System F...	
P3		File folder	04/01/20
Accounts	1,350	TXT File	03/19/20

Remote Name	Size	Type	Modifie
apps		Folder	12/16/
build		Folder	12/16/
debates		Folder	04/01/
desktop		Folder	12/16/
ext		Folder	12/16/
logs		Folder	12/16/
tools		Folder	12/16/
app.reg	1,704	Registrati...	12/16/
LICENSE.txt	11,358	TXT File	11/19/
Makefile	8,170	File	11/19/
Makefile.buildvars	44	BUILDVA...	11/19/
Makefile.sdk	8,505	SDK File	11/19/
Makefile.vars	3,652	VARS File	11/19/
Makefile.vars.priv	2,302	PRIV File	11/19/
VERSION	16	File	11/19/
VERSIONS	161	File	12/16/

Source File	Source Directory	Destination Directory	Size	Status	Speed	Time

Transfer files in text mode | SSH2 - aes128-cbc - hmac-md5 - n | 16 items (35.9 KB)

Now you can figure out how to transfer files into the guest OS. From now on, I assume the 30 files are uploaded to folder `/usr/lib/hue/debates`.

Make sure to use ASCII mode to transfer all data files and source codes in this assignment. (Click the icon labelled as "ASCII" in the above screenshot, before you transfer the files.)

3.2 Create HDFS data folder and prepare data files

Go back to the SSH Secure Shell Client. Type the following commands:

`hdfs dfs -mkdir debates` (Create a folder named "debates" in Hadoop Distributed File System (HDFS) for input data files)

`hdfs dfs -put /usr/lib/hue/debates/*.csv debates` (Copy files from the local file system to the created HDFS folder)

3.3 Create HDFS folders and prepare data files

The guest OS already has a word count example implemented, in a JAR file `hadoop-mapreduce-examples.jar`.

Type the following command:

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar wordcount debates wordcount_output
```

The command will run Word Count on the 30 presidential debate files in HDFS folder "`debates`". The output will be stored in HDFS folder "`wordcount_output`". During its execution, you will see various messages. When it is done, you will see "map 100% reduce 100%" and some statistics about its execution.

Note: If you want to run the above command again, you need to first delete `wordcount_output` from HDFS, by the following command:

```
hdfs dfs –rm -r -f wordcount_output
```

Now, let's see what output files were produced in the output folder:

```
hdfs dfs –ls wordcount_output
```

You should see a file named `part-r-00000` in the folder. Let's see its content:

```
hdfs dfs –cat wordcount_output/part-r-00000
```

You will see some special characters. That's due to noises introduced by our data processing steps in preparing the 30 files. You don't need to worry about it.

4 Hadoop Programs in Python

Hadoop was implemented for Java programs. However, it also has a way to support programs in Python, through its streaming API. In this step, we will do Word count, using Python.

(4.1) Download the Python files mapper.py and reducer.py from P3's entry in Blackboard. Create a folder /usr/lib/hue/wordcount and transfer the 2 Python files into /usr/lib/hue/wordcount. The steps are similar to the ones explained in Section 3.

(4.2) Execute the following commands:

```
chmod +x /usr/lib/hue/wordcount/*.py
```

This makes sure you can run mapper.py by just command `mapper.py` (without the longer form `python mapper.py`). This is enabled by the shebang in the first line of `mapper.py` : `#!/usr/bin/env python`. The same for reducer.py.

Transferring the files in ASCII format from your host OS' file system into the guest OS' file system was critical. If you don't use ASCII, Unix line ending of the files might be destroyed, which will make the shebang not working properly.

(4.3) One nice thing about using Python for Hadoop is that we can even test it without Hadoop environment. Run the following command. It uses unix pipes to connect mapper.py and reducer.py with standard Linux commands `cat` and `sort`.

```
cat ../debates/*.csv | ./mapper.py | sort -k1,2 | ./reducer.py
```

What do you see in the output? Word count results!

If this doesn't work, it is likely due to shebang not working properly. You can use the following commands instead:

```
cat ../debates/*.csv | Python ./mapper.py | sort -k1,2 | Python ./reducer.py
```

(4.4) Now run the Python programs in Hadoop.

Execute the following command to remove the current HDFS folder wordcount_output:

```
hdfs dfs -rm -r -f wordcount_output
```

Then run the Python Hadoop program:

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar -files mapper.py,reducer.py -input debates -output wordcount_output -mapper mapper.py -reducer reducer.py
```

Remember to use the following commands to find out the output file name and check its content:

```
hdfs dfs –ls wordcount_output
```

`hdfs dfs –cat wordcount_output/part-00000` (Note that the file name is different from the one produced by the Java wordcount program.)