CS5344 Lab 1

AY2018/2019 Semester 2

The purpose of this lab is to get you started with Spark, and learn how to write, compile, debug and execute a simple Spark program. You will also be tasked to write and submit your own Spark program individually.

1. **Reference programs and documentation** from Spark release are available at

   https://www.tutorialspoint.com/apache_spark/apache_spark_quick_guide.htm

   https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html

2. A VirtualBox image of Ubuntu with Spark deployment has been configured for you. **Appendix A** gives the instructions on how to download and install it.

3. Alternatively, you can learn to install a stand-alone Spark-2.2.1 instance on Ubuntu by yourself and set up the environment by following the instructions in **Appendix B.** For **Mac users,** you can refer to https://medium.com/luckspark/installing-spark-2-3-0-on-macos-high-sierra-276a127b8b85

4. **Appendix C** and **D** are basic guides to help you get started and run your first Spark *WordCount* program in Java and Python respectively.

5. **Task: Write a Spark application (in Java OR Python) to count the number of words that begin with each letter.** In other words, for each letter, count the total number of words that start with that letter. The words need not be unique and you can ignore the letter case and all non-alphabetic characters. Run your program over the input file provided. Upload the following to the Lab1 folder in IVLE.

   (a) Source program file (with documentation within the code)

   (b) Output file

Appendix A. Install VirtualBox and Configure Ubuntu Image

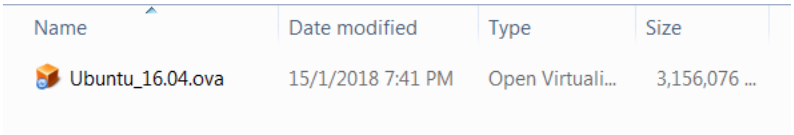
1. Install VirtualBox VM https://www.virtualbox.org/

   VirtualBox supports Windows, Mac OS, Linux, Solaris.

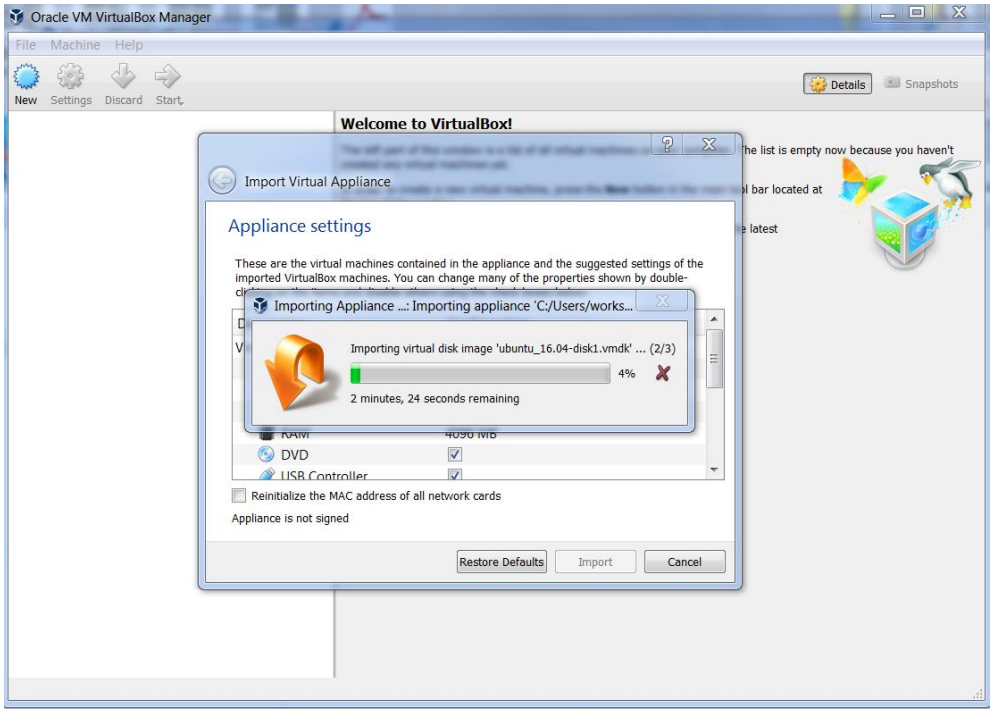
   Download the installation file for your operating system. Double-click on file to install it.

2. Download Ubuntu Image from
   https://www.dropbox.com/s/i8oopo4upl8mm9x/Ubuntu_16.04.zip?dl=0

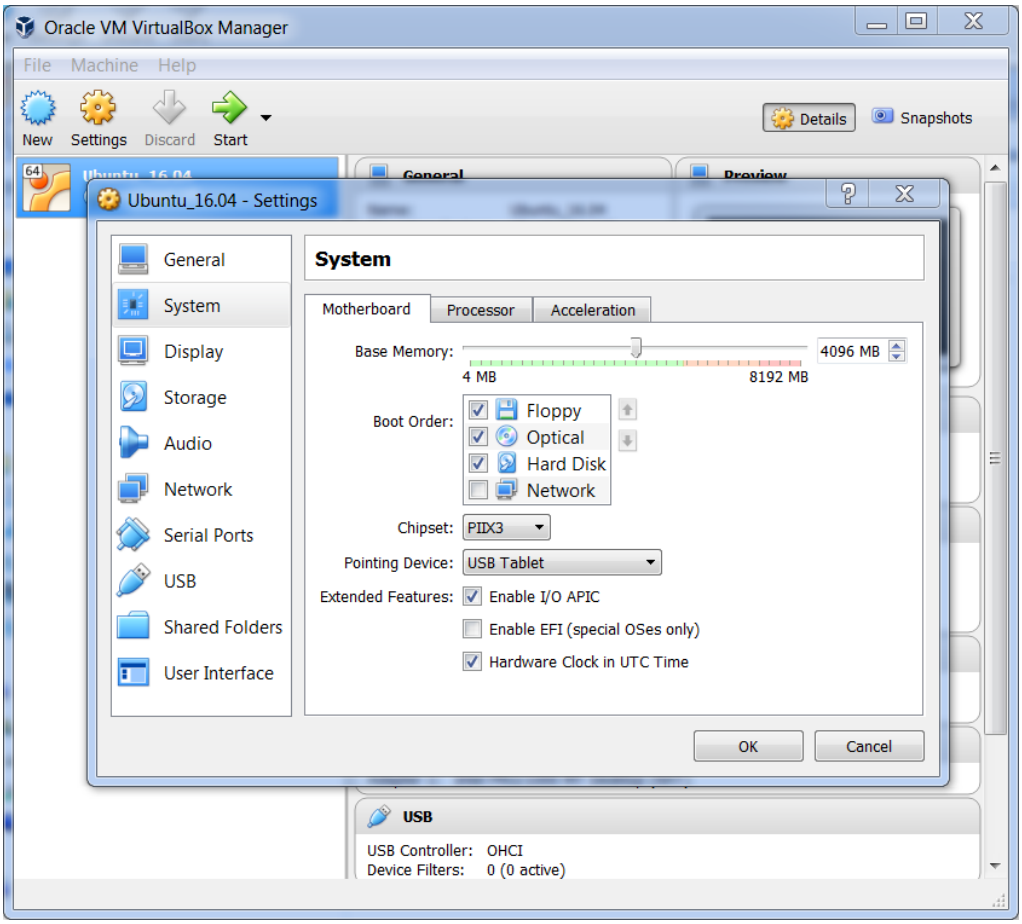
   You will see the following file when you unzip ubuntu_16.04.zip file:



3. Double click Ubuntu_16.04.vbox file, and the image will be loaded into VirtualBox.



   By default, the VM will take up 4G of your physical memory. It is recommended that the VM memory usage does not exceed half of your total memory. To adjust memory usage, click “Settings” button, click “System” tab, and adjust “Base Memory”.



If there is any error while starting the VM related to USB, disable the USB controller from “Settings”, under the “USB” tab.

4. Now you can start the Ubuntu VM. By default, the username is “Spark” and password is “123456” (without quotation marks).

Appendix B. Install Spark-2.2.1 on Ubuntu-16.04 with JDK 8

1. Install JDK 8

Verify Java installation

```
$ java -version
```

If Java is not installed, we install it via the following commands.

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update && sudo apt-get install oracle-java8-installer
```

It may take some time to download the install. When it is done, set the path as follows.

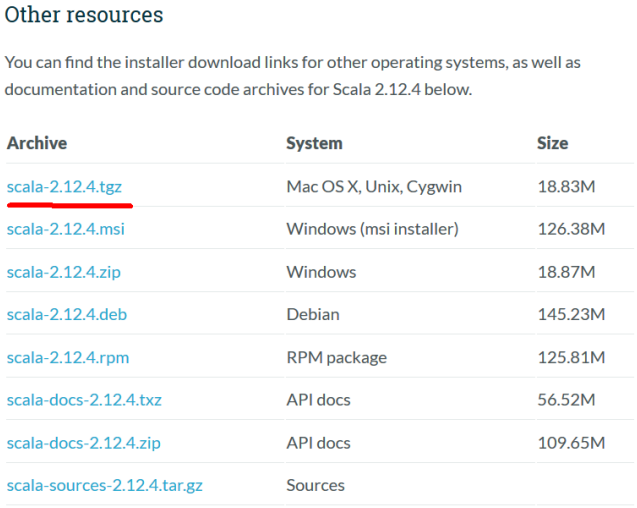
```
$ sudo gedit /etc/environment
```

Append the following line at the end of the file and save it.

JAVA_HOME="/usr/lib/jvm/java-8-oracle"

2. Install Scala

Download Scala in http://www.scala-lang.org/download/



Other resources

You can find the installer download links for other operating systems, as well as documentation and source code archives for Scala 2.12.4 below.

Archive	System	Size
scala-2.12.4.tgz	Mac OS X, Unix, Cygwin	18.83M
scala-2.12.4.msi	Windows (msi installer)	126.38M
scala-2.12.4.zip	Windows	18.87M
scala-2.12.4.deb	Debian	145.23M
scala-2.12.4.rpm	RPM package	125.81M
scala-docs-2.12.4.txz	API docs	56.52M
scala-docs-2.12.4.zip	API docs	109.65M
scala-sources-2.12.4.tar.gz	Sources	

```
$ cd /home/Spark/Downloads
$ tar xvf scala-2.12.4.tgz
```

Use the following commands to move the Scala files to the directory /usr/local/scala

```
$ su -
Password:
# cd /home/Spark/Downloads/
# mv scala-2.12.4 /usr/local/scala
# exit
```

If you have not set the password for root account, use the following command to set it.

```
$ sudo passwd
```

Set the path for Scala.

```
$ sudo gedit /etc/environment
```

Append the following clause to the end of PATH = "/usr/local/sbin:….." in the file, and save it.

:/usr/local/scala/bin

3. Install Maven (to compile java files)

Download Maven from https://maven.apache.org/download.cgi

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binar
the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verif
bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksum
Binary tar.gz archive	apache-maven-3.5.2-bin.tar.gz	apache-maven-3.5.2-bin.tar.gz.md5
Binary zip archive	apache-maven-3.5.2-bin.zip	apache-maven-3.5.2-bin.zip.md5
Source tar.gz archive	apache-maven-3.5.2-src.tar.gz	apache-maven-3.5.2-src.tar.gz.md5
Source zip archive	apache-maven-3.5.2-src.zip	apache-maven-3.5.2-src.zip.md5

Extract Maven files.

```
$ cd /home/Spark/Downloads
$ tar xvf apache-maven-3.5.2-bin.tar.gz
```

Use the following commands to move the Maven files to the directory /usr/local/maven

```
$ su -
Password:
# cd /home/Spark/Downloads/
# mv apache-maven-3.5.2 /usr/local/maven
# exit
```

Set the path for Maven.

```
$ sudo gedit /etc/environment
```

Append the following clause at the end of PATH = "/usr/local/sbin:….." in the file, and save it.

:/usr/local/maven/bin

4. Install Spark

Download Spark from https://spark.apache.org/downloads.html

Extract the file

```
$ cd /home/Spark/Downloads
$ tar xvf spark-2.2.1-bin-hadoop2.7.tgz
```

Move the Spark files to the directory /usr/local/spark

```
$ su -
Password:
# cd /home/Spark/Downloads/
# mv spark-2.2.1-bin-hadoop2.7 /usr/local/spark
# exit
```

Set the path for Spark.

```
$ sudo gedit /etc/environment
```

Append the following clause at the end of PATH = "/usr/local/sbin:….", then save it.

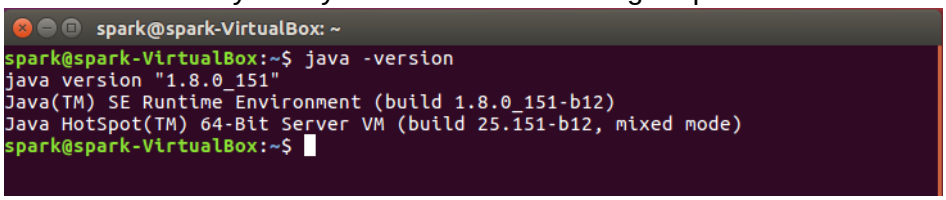
:/usr/local/spark/bin

Now, restart the system to make those changes work!

5. Verify the Software Installations

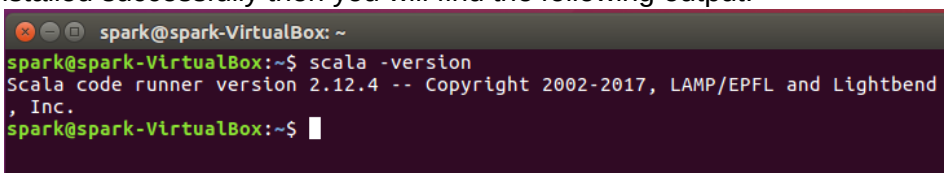
```
$ java -version
```

If Java is installed successfully then you will find the following output.



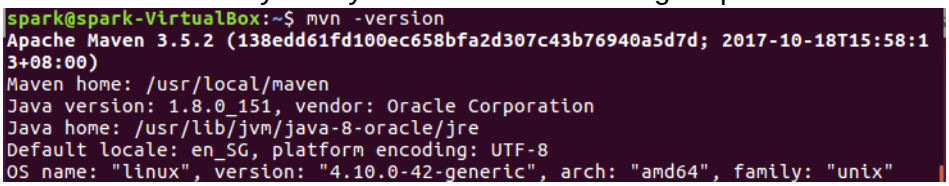

```
$ scala -version
```

If Scala is installed successfully then you will find the following output.



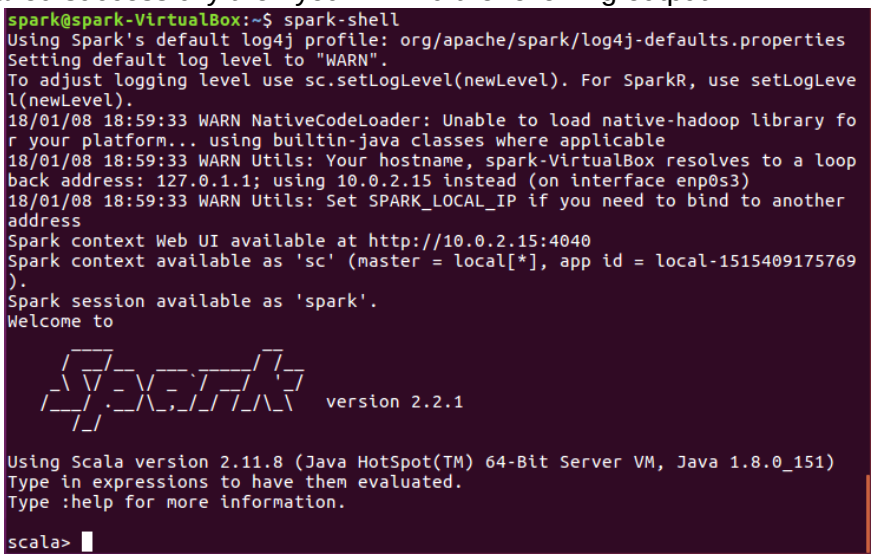

```
$ mvn -version
```

If Maven is installed successfully then you will find the following output.




```
$ spark-shell
```

If spark is installed successfully then you will find the following output.



Appendix C. My First Spark Program (with Java)

1. Download the example files from IVLE (in.txt, pom.xml, SparkWordCount.java).

2. Create a new folder named ”spark-application” with the following files.

```
.
./pom.xml
./in.txt
./src
./src/main
./src/main/java
./src/main/java/com
./src/main/java/com/mycompany
./src/main/java/com/mycompany/app
./src/main/java/com/mycompany/app/SparkWordCount.java
```

3. Execute the following command under “.” directory to generate the .jar file.

```
$ mvn package
```

```
spark@spark-VirtualBox:~/Downloads/spark-java/spark-application$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building wordcount 1.0
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ wordcount
```

...

```
[INFO] Building jar: /home/spark/Downloads/spark-java/spark-application/target/w
ordcount-1.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4.203 s
[INFO] Finished at: 2018-01-09T15:34:16+08:00
[INFO] Final Memory: 33M/79M
[INFO] ------------------------------------------------------------------------
```

There will be a new folder named *target* generated in ”.” directory.

4. Submit to Spark and execute the program using the following command.

```
$ spark-submit --class SparkWordCount --master local target/wordcount-1.0.jar
```

```
spark@spark-VirtualBox:~/Downloads/spark-java/spark-application$ spark-submit --
class SparkWordCount --master local target/wordcount-1.0.jar
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/01/09 15:37:20 INFO SparkContext: Running Spark version 2.2.1
18/01/09 15:37:20 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
18/01/09 15:37:21 WARN Utils: Your hostname, spark-VirtualBox resolves to a loop
back address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
18/01/09 15:37:21 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
18/01/09 15:37:21 INFO SparkContext: Submitted application: wordcount
18/01/09 15:37:21 INFO SecurityManager: Changing view acls to: spark
18/01/09 15:37:21 INFO SecurityManager: Changing modify acls to: spark
18/01/09 15:37:21 INFO SecurityManager: Changing view acls groups to:
18/01/09 15:37:21 INFO SecurityManager: Changing modify acls groups to:
18/01/09 15:37:21 INFO SecurityManager: SecurityManager: authentication disabled
; ui acls disabled; users  with view permissions: Set(spark); groups with view p
ermissions: Set(); users  with modify permissions: Set(spark); groups with modif
y permissions: Set()
```

…

```
18/01/09 15:38:54 INFO MemoryStore: MemoryStore cleared
18/01/09 15:38:54 INFO BlockManager: BlockManager stopped
18/01/09 15:38:54 INFO BlockManagerMaster: BlockManagerMaster stopped
18/01/09 15:38:54 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:
OutputCommitCoordinator stopped!
18/01/09 15:38:54 INFO SparkContext: Successfully stopped SparkContext
18/01/09 15:38:54 INFO ShutdownHookManager: Shutdown hook called
18/01/09 15:38:54 INFO ShutdownHookManager: Deleting directory /tmp/spark-fe3e56
8d-1d5f-4d7a-af40-ff2798cf1b98
spark@spark-VirtualBox:~/Downloads/spark-java/spark-application$
```

You can see a folder named *outfile* generated under "." directory. The result is in the inside file named *part-00000*.

Appendix D. My First Spark Program (with Python)

1. Download the example files from IVLE (in.txt, wordcount.py).
2. Create a new folder named ”spark-application” with the files you downloaded. (in.txt, wordcount.py).
3. To execute the Spark program, using the following command under the folder “…/spark-application/”.

```
$ spark-submit wordcount.py in.txt outfile
```

```
spark@spark-VirtualBox:~/Downloads/spark-python/spark-application$ spark-submit
wordcount.py in.txt outfile
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
19/01/14 12:13:02 WARN Utils: Your hostname, spark-VirtualBox resolves to a loop
back address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
19/01/14 12:13:02 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
19/01/14 12:13:04 INFO SparkContext: Running Spark version 2.2.1
19/01/14 12:13:05 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
19/01/14 12:13:05 INFO SparkContext: Submitted application: wordcount.py
19/01/14 12:13:05 INFO SecurityManager: Changing view acls to: spark
19/01/14 12:13:05 INFO SecurityManager: Changing modify acls to: spark
```

...

```
19/01/14 12:13:12 INFO SparkContext: Successfully stopped SparkContext
19/01/14 12:13:12 INFO ShutdownHookManager: Shutdown hook called
19/01/14 12:13:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-cd9294
59-3369-4305-af19-9427fde05ea3
19/01/14 12:13:12 INFO ShutdownHookManager: Deleting directory /tmp/spark-cd9294
59-3369-4305-af19-9427fde05ea3/pyspark-4dfe6e40-097f-484d-8900-72346cfb353f
spark@spark-VirtualBox:~/Downloads/spark-python/spark-application$
```

You can see a folder named *outfile* generated under ”.” directory. The result is in the inside file named *part-00000*.