Assignment 6 – 5 marks

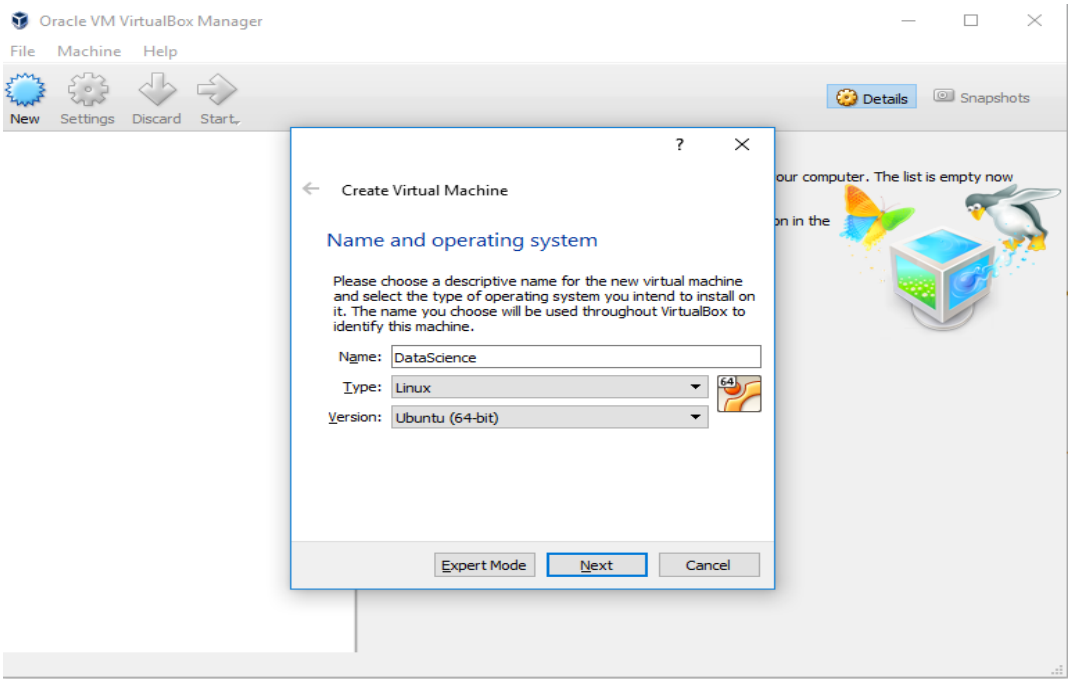
There will be5marks if you implement MapReduced with Hadoop as an underline platform for the word count problem. You can use the instructions in the following page that uses virtual box, Python and word count example:

Step By Step guide for Hadoop installation on Ubuntu 20.04.1 with MapReduce example using Streaming

1. Download VirtualBox from: https://www.virtualbox.org/wiki/Downloads

2. Download Ubuntu 20.04.1 LTS (desktop version **amd64**) from: https://www.ubuntu.com/download/desktop
   Downloaded file : ubuntu-20.04.1-desktop-amd64.iso

3. create a VM with Ubuntu 20.04 image



4. After installing Ubuntu login to the VM and follow instructions given in https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

   Here I am giving step by step details for the installation steps.

5. First, we will update the system's local repository and then install JAVA (default JDK). Run below commands on the terminal.
   sudo apt-get update
   sudo apt install openjdk-8-jdk -y

```
$sudo apt-get update
Hit:1 http://ca.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:3 http://ca.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:4 http://ca.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Fetched 306 kB in 1s (296 kB/s)
Reading package lists... Done
```

6. Now we will install OpenSSH on Ubuntu following commands.
   sudo apt install openssh-server openssh-client -y

6.1 Create new user (Here, put your username on <user> section)

sudo adduser <user>

su - <user>

7. Now we will setup passwordless ssh for Hadoop. First check if you already have passwordless ssh authentication setup; if it is new Ubuntu installation most likely it wouldn't set up. If passwordless ssh authentication is not setup, please follow next step otherwise skip it.

8. run below commands:

ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
ssh localhost

```
mahfuja@mahfuja-VirtualBox:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Your identification has been saved in /home/mahfuja/.ssh/id_rsa
Your public key has been saved in /home/mahfuja/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:glBx1r0TaNNKUX00Ay+Xy+qssh+fhqcdQRVtcXENCKM mahfuja@mahfuja-VirtualBox
The key's randomart image is:
+---[RSA 3072]----+
|    o.o..*+o.+B**|
|   . o  *.+.oo.+*|
|  .    oEo oo.+. |
|   . .  . o. + . |
|    . . S  .. o  |
|       .     o   |
|          ..o    |
|        . .*oo   |
|        .++=*    |
+----[SHA256]-----+
```

```
mahfuja@mahfuja-VirtualBox:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:lr7g1F/IPabP+HOKNHLR6oZNBKjtcSmTKA3Tt8QVE4w.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.8.0-38-generic x86_64)
```

8. Use the mirror link and download the Hadoop package with the wget command:

wget https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

```
mahfuja@mahfuja-VirtualBox:~$ wget https://downloads.apache.org/hadoop/common/h
adoop-3.2.1/hadoop-3.2.1.tar.gz
--2021-01-20 22:55:00--  https://downloads.apache.org/hadoop/common/hadoop-3.2.
1/hadoop-3.2.1.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 2a01:4f8
:10a:201a::2
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 359196911 (343M) [application/x-gzip]
Saving to: ‘hadoop-3.2.1.tar.gz’

hadoop-3.2.1.tar.gz 100%[===================>] 342.56M  5.93MB/s    in 49s

2021-01-20 22:55:49 (6.99 MB/s) - ‘hadoop-3.2.1.tar.gz’ saved [359196911/359196
911]
```

```
mahfuja@mahfuja-VirtualBox:~$ ls
Desktop  Documents  hadoop-3.2.1         Music     Public     tmpdata
dfsdata  Downloads  hadoop-3.2.1.tar.gz  Pictures  Templates  Videos
```

9. Once the download is complete, extract the files to initiate the Hadoop installation
   tar xzf hadoop-3.2.1.tar.gz

10. See the list directories

```
mahfuja@mahfuja-VirtualBox:~$ ls
Desktop  Documents  hadoop-3.2.1         Music     Public     tmpdata
dfsdata  Downloads  hadoop-3.2.1.tar.gz  Pictures  Templates  Videos
```

11. Configure Hadoop Environment Variables
    Edit the .bashrc shell configuration file using a text editor of your choice
    sudo nano .bashrc

    Define the Hadoop environment variables by adding the following content to the end of the file .bashrc file

```
export HADOOP_HOME=/home/<user>/hadoop-3.2.1
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS"-Djava.library.path=$HADOOP_HOME/lib/native"
```

```
export HADOOP_HOME=/home/mahfuja/hadoop-3.2.1
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS"-Djava.library.path=$HADOOP_HOME/lib/native"

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

12. Now will find the Java path, run the following command in your terminal window:
    which javac
    readlink -f /usr/bin/javac

```
mahfuja@mahfuja-VirtualBox:~$ which javac
/usr/bin/javac
mahfuja@mahfuja-VirtualBox:~$ readlink -f /usr/bin/javac
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

The section of the path just before the /bin/javac directory needs to be assigned to the $JAVA_HOME variable on the hadoop-env.sh File

13. Edit hadoop-env.sh File
    Change directory to extracted folder and edit Hadoop-env.sh file for updating java home_path. Use the following commands
    cd hadoop-3.2.1
    nano etc/hadoop/hadoop-env.sh

```
mahfuja@mahfuja-VirtualBox:~$ cd hadoop-3.2.1/
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.1$ ls
bin  include  libexec      NOTICE.txt  sbin
etc  lib      LICENSE.txt  README.txt  share
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.1$ nano etc/hadoop/hadoop-env.sh
```

Uncomment the JAVA_HOME variable and add the following line in hadoop-env.sh file:
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

```
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

14. Now we will update some configuration files for pseudo-distributed operation. First we will edit etc/hadoop/core-site.xml file as below.

Sudo nano etc/hadoop/core-site.xml

```
<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/<user>/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

Here, put your username on <user> section

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/mahfuja/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

15. Edit hdfs-site.xml File
    Create directory for NameNode and DataNode storage:

cd mkdir dfsdata
cd mkdir dfsdata/namenode
cd mkdir dfsdata/datanode

```
mahfuja@mahfuja-VirtualBox:~$ mkdir dfsdata
mahfuja@mahfuja-VirtualBox:~$ mkdir dfsdata/namenode
mahfuja@mahfuja-VirtualBox:~$ mkdir dfsdata/datanode
mahfuja@mahfuja-VirtualBox:~$
```

Sudo nano etc/hadoop/hdfs-site.xml

```
<configuration>
<property>
  <name>dfs.data.dir</name>
  <value>/home/<user>/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/<user>/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Here, put your username on <user> section

```
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
  <name>dfs.data.dir</name>
  <value>/home/mahfuja/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/mahfuja/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

16. Now we will start NameNode and DataNode but before that we will format the HDFS file system.

hdfs namenode -format

```
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.1$ hdfs namenode -format
WARNING: /home/mahfuja/hadoop-3.2.1/logs does not exist. Creating.
2021-01-20 23:25:13,722 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = mahfuja-VirtualBox/127.0.1.1
```

Now, Navigate to the hadoop-3.2.2/sbin directory and execute the following commands to start the NameNode and DataNode:

cd sbin
./start-dfs.sh
./start-yarn.sh

```
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.1$ cd sbin/
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.1/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [mahfuja-VirtualBox]
2021-01-20 23:28:24,870 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
```

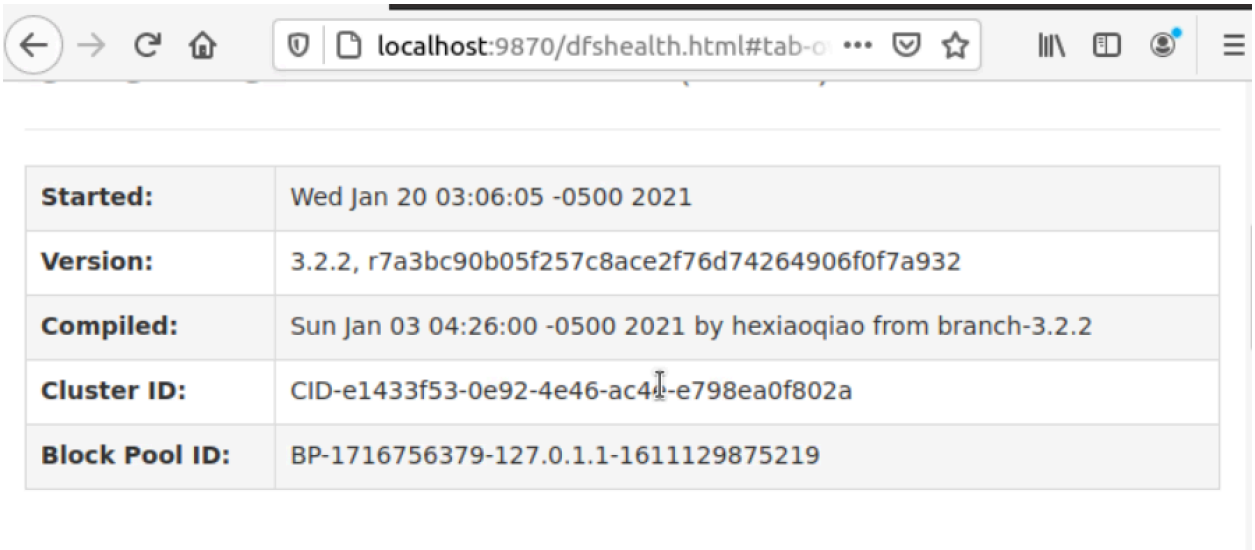
```
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.1/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

Type this simple command to check if all the daemons are active and running as Java processes:

jps

```
mahfuja@mahfuja-VirtualBox:~/hadoop-3.2.2/sbin$ jps
8515 NameNode
8806 SecondaryNameNode
9223 NodeManager
9099 ResourceManager
8636 DataNode
9774 Jps
```

17. Now we can access Web-interface for NameNode at http://localhost:9870/



18. Now let's create some directories in HDFS filesystem.

```
$bin/hdfs dfs -mkdir /user
$bin/hdfs dfs -mkdir /user/demo
$
```

19. Let's download one html page http://hadoop.apache.org and upload into HDFS file system.

wget http://hadoop.apache.org -O hadoop_home_page.html

```
$wget http://hadoop.apache.org/ -O hadoop_home_page.html
--2017-10-14 17:42:01--  http://hadoop.apache.org/
Resolving hadoop.apache.org (hadoop.apache.org)... 140.211.11.105, 195.154.151.36, 2001:bc8:2142:300::
Connecting to hadoop.apache.org (hadoop.apache.org)|140.211.11.105|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38000 (37K) [text/html]
Saving to: 'hadoop_home_page.html'

hadoop_home_page.html                   100%[===================================================================================================>]  37.11K  --.-KB/s    in 0.09s

2017-10-14 17:42:06 (428 KB/s) - 'hadoop_home_page.html' saved [38000/38000]

$
```

```
$bin/hdfs dfs -put  hadoop_home_page.html /user/demo
$ls /user/demo
ls: cannot access '/user/demo': No such file or directory
$
```

Please note that HDFS file system is not same as root file system.

```
$bin/hdfs dfs -ls /user
Found 1 items
drwxr-xr-x   - [redacted] supergroup          0 2017-10-14 17:46 /user/demo
$bin/hdfs dfs -ls /user/demo
Found 1 items
-rw-r--r--   1 [redacted] supergroup      38000 2017-10-14 17:46 /user/demo/hadoop_home_page.html
$ls /user
ls: cannot access '/user': No such file or directory
$ls /user/demo
ls: cannot access '/user/demo': No such file or directory
$
```

Grep example:

20. For this example we are using hadoop-mapreduce-examples-3.2.1.jar file which comes along with Hadoop. In this example we are trying to count the total number of 'https' word occurrences in the given files. First we run the Hadoop job then copy the results from HDFS to the local file system. (you may get 3 occurrences of https)

Command:

hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar grep /user/demo/hadoop_home_page.html -output /user/demo/hadoop_home_page.html_OUTPUT_2

```
$bin/hdfs dfs -ls  /user/demo/
Found 3 items
-rw-r--r--   1 [redacted] supergroup      38000 2017-10-14 17:46 /user/demo/hadoop_home_page.html
drwxr-xr-x   - [redacted] supergroup          0 2017-10-14 18:32 /user/demo/hadoop_home_page.html_OUTPUT_1
drwxr-xr-x   - [redacted] supergroup          0 2017-10-14 18:55 /user/demo/hadoop_home_page.html_OUTPUT_2
$bin/hdfs dfs -get /user/demo/hadoop_home_page.html_OUTPUT_2 hadoop_home_page.html_OUTPUT_2
$
$
$cat hadoop_home_page.html_OUTPUT_2/part-r-00000
2       https
$
$
$cat hadoop_home_page.html |grep -o -w  'https' |wc -w
2
$
```

We can see that there are 2 occurrences of https in the given file and same we can validate using wget command.

Wordcount example:

21. For wordcount example also we are using hadoop-mapreduce-examples-2.7.4.jar file. The wordcount example returns the count of each word in the given documents.

Command:

hadoop jar ../share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount /user/demo/hadoop_home_page.html /user/demo/hadoop_home_page.html_OUTPUT_1

Another three commands are in the following screen shot.

```
$bin/hdfs dfs -ls  /user/demo/
Found 2 items
-rw-r--r--   1 [redacted] supergroup      38000 2017-10-14 17:46 /user/demo/hadoop_home_page.html
drwxr-xr-x   [redacted] supergroup          0 2017-10-14 18:32 /user/demo/hadoop_home_page.html_OUTPUT_1
$bin/hdfs dfs -get /user/demo/hadoop_home_page.html_OUTPUT_1 hadoop_home_page.html_OUTPUT_1
$head -n 50  hadoop_home_page.html_OUTPUT_1/part-r-00000
"       3
"-//W3C//DTD    1
"Swiss  1
"http://www.w3.org/TR/html4/loose.dtd"> 1
"release        2
&copy;  1
&gt;    1
   1
'.google-analytics.com/ga.js';  1
```

Wordcount using Hadoop streaming (python)

22. Here is mapper and reducer program for wordcount.

```
$ls
bin  hadoop_home_page.html           hadoop_home_page.html_OUTPUT_2      include  libexec      logs        README.txt  share  wordcount_map.py
etc  hadoop_home_page.html_OUTPUT_1  hadoop_home_page.html_OUTPUT_COUNT  lib      LICENSE.txt  NOTICE.txt  sbin        src    wordcount_red.py
$cat wordcount_map.py
#!/usr/bin/env python
import sys
for line in sys.stdin:
    line = line.strip()
    words = line.split()
    for word in words:
        print '%s\t%s' % (word, 1)

$
$
$cat wordcount_red.py
#!/usr/bin/env python
import sys
tmp_word = None
total_count = 0
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t')
    count = int(count)
    if tmp_word == word:
        total_count += count
    else:
        print '%s\t%s' % (tmp_word, total_count)
        total_count = count
        tmp_word = word
print '%s\t%s' % (tmp_word, total_count)

$
```

23. We run the program as below and the copy the result to local file system.

Command :

hadoop jar ../share/hadoop/tools/lib/hadoop-streaming-3.2.1.jar -mapper ./wordcount_map.py -reducer ./wordcount_red.py -input /user/demo/hadoop_home_page.html -output /user/demo/hadoop_home_page.html_OUTPUT_COUNT

```
$bin/hdfs dfs -ls  /user/demo/
Found 4 items
-rw-r--r--   1 [redacted] supergroup      38000 2017-10-14 17:46 /user/demo/hadoop_home_page.html
drwxr-xr-x   - [redacted] supergroup          0 2017-10-14 18:32 /user/demo/hadoop_home_page.html_OUTPUT_1
drwxr-xr-x   - [redacted] supergroup          0 2017-10-14 18:55 /user/demo/hadoop_home_page.html_OUTPUT_2
drwxr-xr-x   - [redacted] supergroup          0 2017-10-14 19:17 /user/demo/hadoop_home_page.html_OUTPUT_COUNT
$bin/hdfs dfs -get /user/demo/hadoop_home_page.html_OUTPUT_COUNT hadoop_home_page.html_OUTPUT_COUNT
$head hadoop_home_page.html_OUTPUT_COUNT/part-00000
None    0
"       3
"-//W3C//DTD    1
"Swiss  1
"http://www.w3.org/TR/html4/loose.dtd"> 1
"release        2
&copy;  1
&gt;    1
   1
'.google-analytics.com/ga.js';  1
```

Please note that if you power off the virtual machine and if you are not sure how to start the namenode without formatting it you need to do the assignment all over
again in order to demo. But instead of power off the machine, if you save the state of the machine although it is temporary solution it should work and you don’t need to do everything all over.