Part 3

For this part of the assignment, you will run wordcount on a single-node Hadoop instance. I am going to provide detailed instructions to help you get Hadoop running. The instructions are following Hadoop: The Definitive Guide instructions presented in Appendix A: Installing Apache Hadoop.

You can download 2.6.4 from here. You can copy-paste these commands (right-click in PuTTy to paste, but please watch out for error messages and run commands one by one)

Install ant to list java processes
sudo yum install ant

(wget command stands for "web get" and lets you download files to your instance from a URL link)
wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/hadoop-2.6.4.tar.gz

(unpack the archive)
tar xzf hadoop-2.6.4.tar.gz

Modify the conf/hadoop-env.sh to add to it the JAVA_HOME configuration:
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64/

You can open it by running (using nano or your favorite editor instead of nano).
nano hadoop-2.6.4/etc/hadoop/hadoop-env.sh

Note that the # comments out the line, so you would comment out the original JAVA_HOME line replacing it by the new one as below:

```
  GNU nano 2.5.3     File: hadoop-2.6.4/etc/hadoop/hadoop-env.sh

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
# export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
```

modify the .bashrc file to add these two lines:
export HADOOP_HOME=~/hadoop-2.6.4
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

.bashrc file contains environment settings to be configured automatically on each login. You can open the .bashrc file by running
nano ~/.bashrc

```
# User specific aliases and functions

export HADOOP_HOME=~/hadoop-2.6.4
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

To immediately refresh the settings (that will be automatic on next login), run
source ~/.bashrc

Next, follow the instructions for Pseudodistributed Mode for all 4 files.

(to edit the first config file)
nano hadoop-2.6.4/etc/hadoop/core-site.xml

Make sure you paste the settings between the <configuration> and </configuration> tags, like in the screenshot below. NOTE: The screenshot below is only one of the 4 files, all files are different. The contents of each file are described in the **Appendix A** in the Hadoop book, the relevant appendix is also included with the homework assignment. I am also including a .txt file (HadoopConfigurationText) so that it is easier to copy-paste.


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

<configuration>

<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost/</value>
</property>

</configuration>
```

nano hadoop-2.6.4/etc/hadoop/hdfs-site.xml
(mapred-site.xml file is not there, run the following single line command to create it by copying from template. Then you can edit it as other files.)
cp hadoop-2.6.4/etc/hadoop/mapred-site.xml.template hadoop-2.6.4/etc/hadoop/mapred-site.xml
nano hadoop-2.6.4/etc/hadoop/mapred-site.xml
nano hadoop-2.6.4/etc/hadoop/yarn-site.xml

To enable passwordless ssh access (we will discuss SSH and public/private keys in class), run these commands:
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

test by running (and confirming a one-time warning)
ssh localhost
exit

Format HDFS (i.e., first time initialize)

hdfs namenode -format

Start HDFS, Hadoop and history server (answer a 1-time yes if you asked about host authenticity)

start-dfs.sh
start-yarn.sh
mr-jobhistory-daemon.sh start historyserver

Verify if everything is running:
jps

(NameNode and DataNode are responsible for HDFS management; NodeManager and ResourceManager are serving the function similar to JobTracker and TaskTracker. We will discuss function of all of those on Thursday.)

Create a destination directory
hadoop fs -mkdir /data

Download a large text file using

wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/bioproject.xml

Copy the file to HDFS for processing

hadoop fs -put bioproject.xml /data/

(you can optimally verify that the file was uploaded to HDFS by hadoop fs -ls /data)
Submit a screenshot of this command

Run word count on the downloaded text file, using the time command to determine the total runtime of the MapReduce job. You can use the following (single-line!) command. This invokes the wordcount example built into the example jar file, supplying /data/bioproject.xml as the input and /data/wordcount1 as the output directory. Please remember this is one command, if you do not paste it as a single line, it will not work.

time hadoop jar hadoop-2.6.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.4.jar wordcount /data/bioproject.xml /data/wordcount1

Report the time that the job took to execute as screenshot

(this reports the size of a particular file or directory in HDFS. The output file will be named part-r-00000)

```
hadoop fs -du /data/wordcount1/
```

(Just like in Linux, the cat HDFS command will dump the output of the entire file and grep command will filter the output to all lines that matches this particular word). To determine the count of occurrences of "subarctic", run the following command:

```
hadoop fs -cat /data/wordcount1/part-r-00000 | grep subarctic
```

It outputs the entire content of part-r-00000 file and then uses pipe | operator to filter it through grep (filter) command. If you remove the pipe and grep, you will get the entire word count content dumped to screen, similar to cat command.

Congratulations, you just finished running wordcount using Hadoop.

Submit a single document containing your written answers. Be sure that this document contains your name and "CSC 555 Assignment 1" at the top.