ITCS 3190– Cloud Computing For Data Analysis

Project 1: Write your first MapReduce program

Due: March 8, 2020

First part: tutorial on how to run MapReduce

If you are using Ubuntu or Mac, just open terminal and go ahead. If you are using Windows, please refer to section B first!

Section A:

Example explained with WordCount.java file

Replace <username> with your 49er username below and use your 49er password.

Step 1:
To connect to the dsba-hadoop cluster (from a Unix machine)

```
$ ssh dsba-hadoop.uncc.edu -l <username>
```

For example $ ssh dsba-hadoop.uncc.edu -l dyang33

Step 2:
Your home directory on dsba-hadoop.uncc.edu is /users/<username>/

To move files from local system to Home directory on the dsba-hadoop cluster (Done from your local system console, not from cluster):

```
$ scp file <username>@dsba-hadoop.uncc.edu:/users/<username>
```

Example, you have to upload your source code WordCount.java from your local PC.

```
$ scp WordCount.java <username>@dsba-hadoop.uncc.edu:/users/<username>
```

[Note the period at the end of the above command.]

Then, upload the input file wiki-micro.txt that you are going to process

```
$ scp wiki-micro.txt <username>@dsba-hadoop.uncc.edu:/users/<username>
```

Step 3:

Before you run the sample, you must create input and output locations in HDFS. Use the following commands to create the input directory/user/<username>/input in HDFS:

(Steps below follow a sequence similar to the Usage instructions at https://www.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_usage.html

```
$ hadoop fs -mkdir /user/<username>/input
```

And then move input file wiki-micro.txt to the /user/cloudera/wordcount/input directory in HDFS.

```
$ hadoop fs -put wiki-micro.txt /user/<username>/input
```

Then use the -ls command to check if the input file exist or not.

```
$ hadoop fs -ls /user/<username>/input
```

Step 4:

Compile the WordCount class:

```
$ mkdir build
$ javac -cp /opt/cloudera/parcels/CDH/lib/hadoop/*:/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/* WordCount.java -d build -Xlint
```

Create a JAR file for the WordCount application.

```
$ jar -cvf wordcount.jar -C build/  .
```

Run the WordCount application from the JAR file, passing the paths to the input and output directories in HDFS.

```
$hadoop   jar   wordcount.jar   org.apache.hadoop.examples.WordCount /user/<username>/input /user/<username>/output
```

When you look at the output, all of the words are listed in UTF-8 alphabetical order (capitalized words first). The number of occurrences from all input files has been reduced to a single sum for each word.

```
$ hadoop fs -cat /user/<username>/output/*
```

If you want to run the sample again, you first need to remove the output directory. Use the following command.

```
$ hadoop fs -rm -r /user/<username>/output
```

Step 5:

Create a output directory on cluster

```
$ mkdir result
```

get all the result from HDFS

```
$ hadoop dfs -get /user/<username>/result
```

To move the output file from cluster to your local system (done from your local system console, not from cluster):

```
$ scp     dyang33@dsba-hadoop.uncc.edu:/users/dyang33/result/*     /<local directory>
```

$ [Note the period at the end of the above command.]

This is what you should submit!!

Additional information:

1. You can monitor the cluster by launching Firefox and going to the following URLs:

Namenode (HA) info:	http://dbkhm1.uncc.edu:50070
	http://dbkhm2.uncc.edu:50070
Yarn/MapRed (HA) Info:	http://dbkhm1.uncc.edu:8088

	http://dbkhm2.uncc.edu:8088
MapRed Job History:	http://dbkhm3.uncc.edu:19888/jobhistory
HBase Information:	http://dbkhm3.uncc.edu:60010
Hue (Hadoop User Experience):	https://dsbahadoop.uncc.edu:8888

** Just FYI: access to Hue web UI can be done from your desktop browser **

2. To kill your Hadoop job on the cluster:

```
$ hadoop job -list | grep <username>
```

gives a list of your jobs. The job id is the first column returned by the list command.

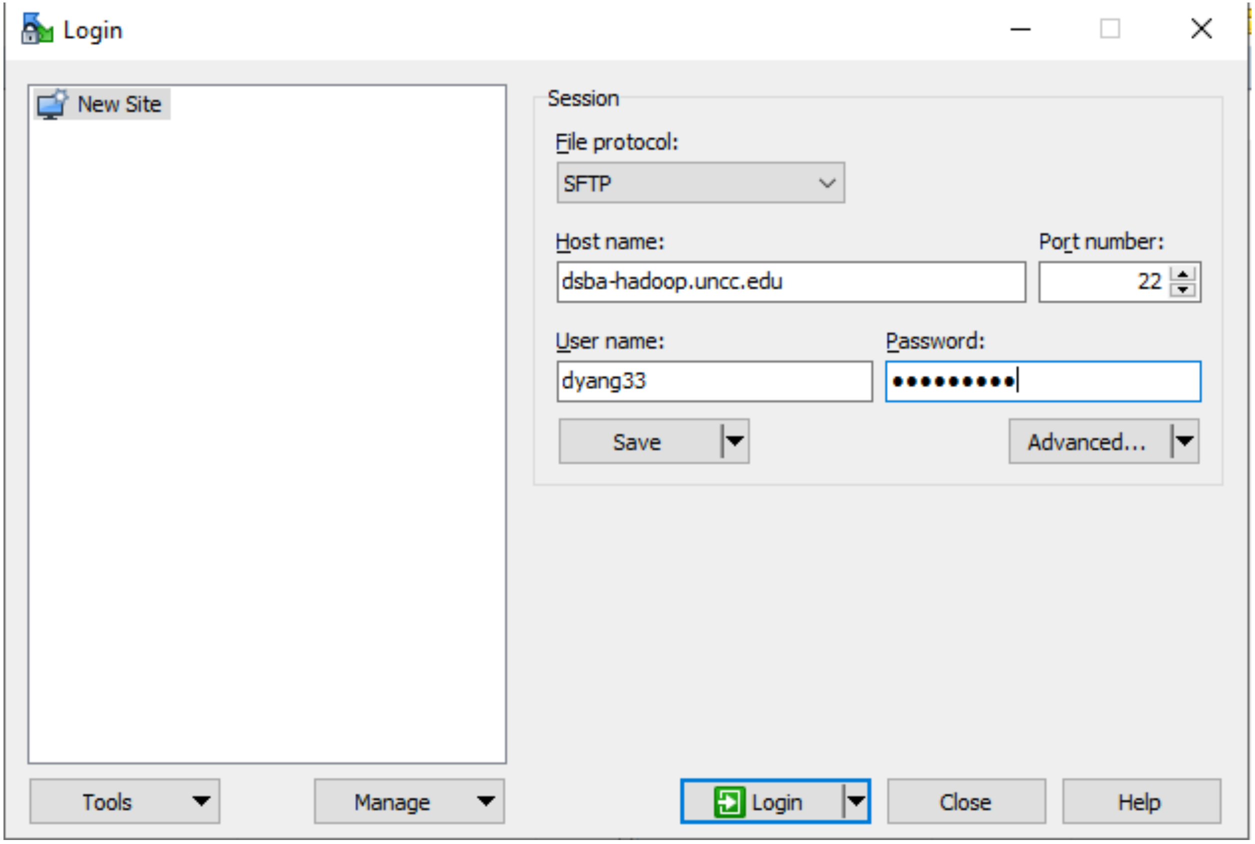
```
$ hadoop job -kill <job-id>
```

kills your job so you free up the stuck resources.

Section B:
If you are using windows, please download winscp so you can access cluster:

1. Dowload and install the winscp first:
https://winscp.net/eng/download.php

2. Login in your account:



Login

New Site

Session

File protocol: SFTP

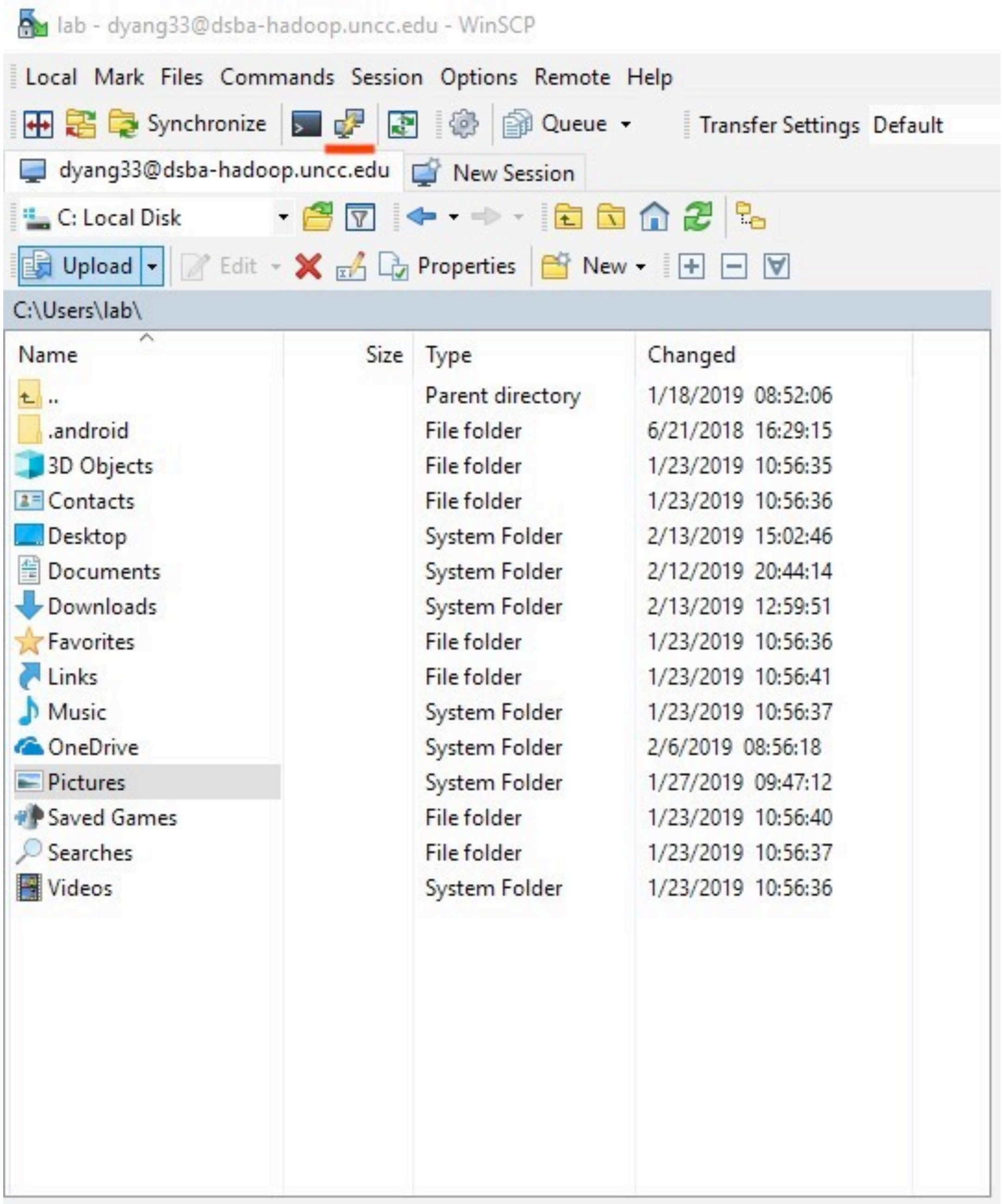
Host name: dsba-hadoop.uncc.edu

Port number: 22

User name: dyang33

Password: •••••••••

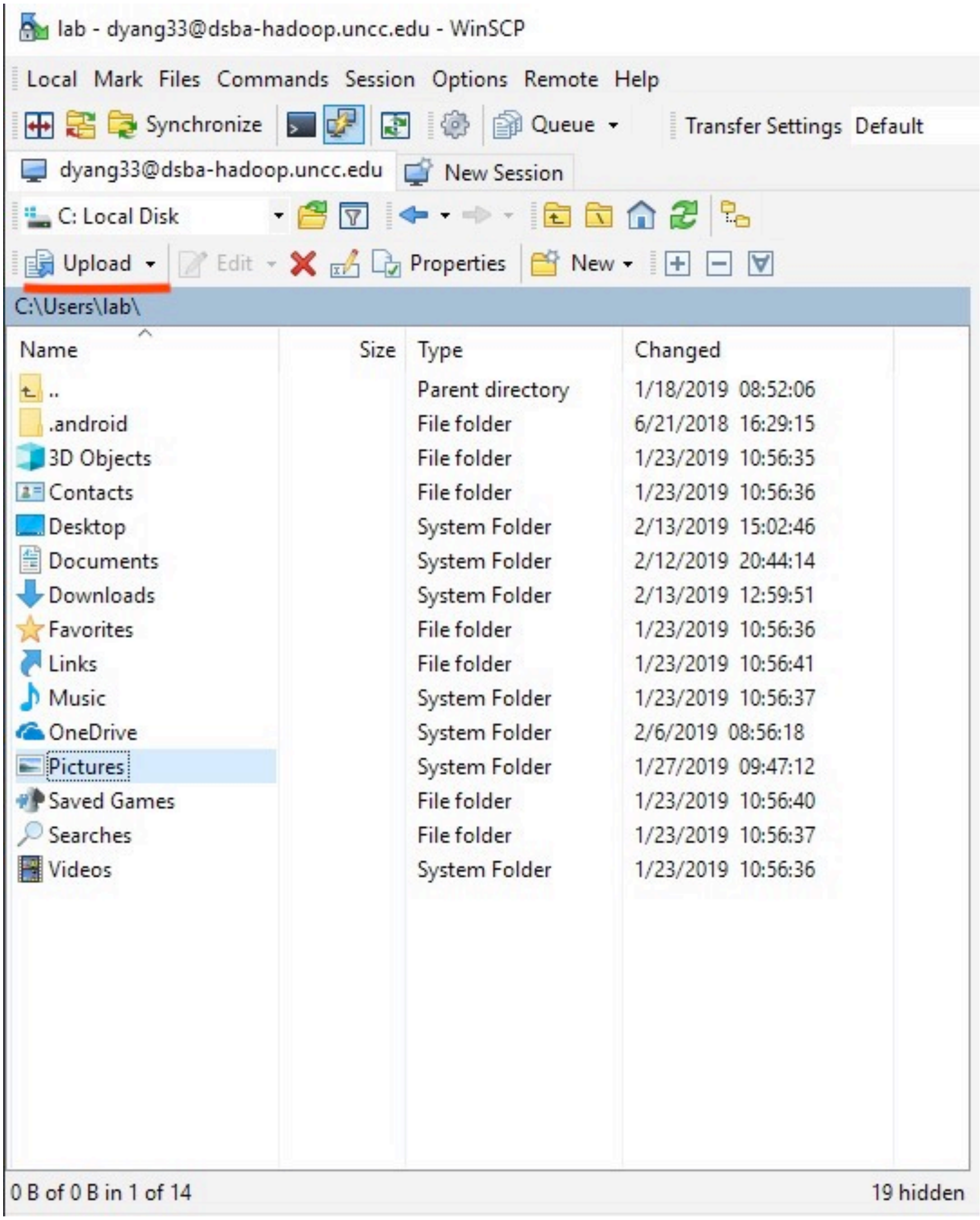
3. click on the button “open session in putty”, so you can access cluster via terminal


lab - dyang33@dsba-hadoop.uncc.edu - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
dyang33@dsba-hadoop.uncc.edu
New Session
C: Local Disk
Upload
Edit
Properties
New

C:\Users\lab\

Name	Size	Type	Changed
..		Parent directory	1/18/2019 08:52:06
.android		File folder	6/21/2018 16:29:15
3D Objects		File folder	1/23/2019 10:56:35
Contacts		File folder	1/23/2019 10:56:36
Desktop		System Folder	2/13/2019 15:02:46
Documents		System Folder	2/12/2019 20:44:14
Downloads		System Folder	2/13/2019 12:59:51
Favorites		File folder	1/23/2019 10:56:36
Links		File folder	1/23/2019 10:56:41
Music		System Folder	1/23/2019 10:56:37
OneDrive		System Folder	2/6/2019 08:56:18
Pictures		System Folder	1/27/2019 09:47:12
Saved Games		File folder	1/23/2019 10:56:40
Searches		File folder	1/23/2019 10:56:37
Videos		System Folder	1/23/2019 10:56:36

4. if you want to upload your file to cluster, you can choose your file on your local system, and click “upload” button.



C:\Users\lab\

Name	Size	Type	Changed
..		Parent directory	1/18/2019 08:52:06
.android		File folder	6/21/2018 16:29:15
3D Objects		File folder	1/23/2019 10:56:35
Contacts		File folder	1/23/2019 10:56:36
Desktop		System Folder	2/13/2019 15:02:46
Documents		System Folder	2/12/2019 20:44:14
Downloads		System Folder	2/13/2019 12:59:51
Favorites		File folder	1/23/2019 10:56:36
Links		File folder	1/23/2019 10:56:41
Music		System Folder	1/23/2019 10:56:37
OneDrive		System Folder	2/6/2019 08:56:18
Pictures		System Folder	1/27/2019 09:47:12
Saved Games		File folder	1/23/2019 10:56:40
Searches		File folder	1/23/2019 10:56:37
Videos		System Folder	1/23/2019 10:56:36

0 B of 0 B in 1 of 14 19 hidden

5. After these, please refer to step 4 from the Ubuntu tutorial.

If you want to run MapReduce on your own computer, please refer to vbox_tutorial provided on Canvas.

Second part: Your task!!!

1. You may need first to get Hadoop running and run the example WordCount program. Download the output from cluster and then submit your results.

2. You should then modify the WordCount program so it outputs the wordcount for each distinct word in each file. So the output of this DocWordCount program should be of the form ‘word#####filename count’, where ‘#####’ serves as a delimiter between word and filename and tab serves as a delimiter between filename and count. Submit your source code in a file named DocWordCount.java. **(The input here should be the Canterbury corpus provided in the package)**

   Explanation:

   Consider two simple files file1.txt and file2.txt.

   ```
   $ echo "Hadoop is yellow Hadoop" > file1.txt
   $ echo "yellow Hadoop is an elephant" > file2.txt
   ```

   Running ‘DocWordCount.java’ on these two files will give an output similar to that below, where ##### is a delimiter.

   Output of DocWordCount.java

   ```
   yellow#####file2.txt 1
   Hadoop#####file2.txt         1
   is#####file2.txt         1
   elephant#####file2.txt       1
   yellow#####file1.txt 1
   Hadoop#####file1.txt         2
   is#####file1.txt         1
   an#####file2.txt       1
   ```

Submissions:

Upload to Canvas a zip file named your_uncc_id.zip (e.g. dyang33.zip if your email address is dyang33@uncc.edu). The file should contain:

1. Output from running WordCount.java on the text file wiki-micro.txt
2. Source code for DocWordCount.java
3. A README file with execution instructions.
4. The output from running DocWordCount.java on the text files of the Canterbury corpus provided in the assignment package. Name the output files DocWordCount.out

Guidelines:

Please make sure your source code is adequately commented, and also make sure each of your files has your name and email id included at the top of the file. Please also include a README file where you provide any execution instructions.

Assignments are to be done individually. See course syllabus for late submission policy and academic integrity guidelines.

Grading Rubric:

Total 100.

1. Readme file (10 pts).
2. Output from running WordCount.java (30 pts).
3. DocWordCount.java Source code (30 pts).

- **MapReduce logic (10 pts)**
- **correct arguments (10 pts)**
- **comments (10 pts)**

4. Output from running DocWordCount.java (30 pts).