Lab 1: Introduction and Setup
Big Data Spring 2017

The goal of this lab is to get you set up with some of the tools we will be using throughout the course.

Before starting this lab, you will need your HPC account username and password. You should have received an email with these.

Part 1: HPC@NYU

First, we will get set up with NYU's high-performance computing (HPC) infrastructure, documented at http://wikis.nyu.edu/display/NYUHPC.

Windows users: before starting, you will need to download the PuTTY utilities package from http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html.

Step 1: HPC@NYU tutorial

Please follow the official HPC@NYU Tutorial #1, available at
http://wikis.nyu.edu/pages/viewpage.action?pageId=53859101

Note that there are many other tutorials and instructions for accessing and using the HPC clusters available at https://wikis.nyu.edu/display/NYUHPC.

Part 2: Running a Hadoop job on dumbo

Next, we will experiment with the Hadoop cluster (dumbo). In particular, we will explore the Hadoop-Streaming utility, documented at
http://hadoop.apache.org/docs/current/hadoop-streaming/HadoopStreaming.html
Hadoop-Streaming exposes a subset of Hadoop's functionality to other programming languages (Hadoop uses Java). We will use Hadoop-Streaming to execute MapReduce programs written in Python.

Information about and instructions for using the dumbo cluster can be found here:
https://wikis.nyu.edu/display/NYUHPC/Clusters+-+Dumbo.

Step 1: Accessing the Hadoop Cluster

Logging in to the cluster:

1. Log into the main HPC node. To do this,
    a. On MacOS, open the terminal and type `ssh your_netid@hpc.nyu.edu`
    b. On Windows, open PuTTY.exe. In the “Host Name” field, type `your_netid@hpc.nyu.edu` , and then click “Open” at the bottom.
2. Enter your password when prompted.
3. From the HPC node, log into the Hadoop cluster. To do this, type `ssh dumbo`. Enter password again (if prompted).
4. Once you’ve logged in to “dumbo”, make sure you are using the “bash” shell. The prompt should say something like “-bash-4.1$”. If not, type “echo $0” and it should say “-bash”. If it says something else, like “tsch”, type “bash” to switch to bash. (You are welcome to use another shell, but you’ll need to modify the following instructions, so we don’t recommend it unless you really know what you’re doing.)

You are now logged in to dumbo. Type `pwd` to see what directory you are currently in (it should be ‘/home/your-netID’.

Step 2: Create aliases for Hadoop

You will be using a set of commands to run hadoop jobs, and it will save you some time to first create aliases for them:

1. Log into "dumbo" following the instructions above and type the following commands on your terminal (Note: you should not have any spaces around "=" signs, and copying and pasting will not work):

```
alias hfs='/usr/bin/hadoop fs '
export HAS=/opt/cloudera/parcels/CDH-5.9.0-1.cdh5.9.0.p0.23/lib
export HSJ=hadoop-mapreduce/hadoop-streaming.jar
alias hjs='/usr/bin/hadoop jar $HAS/$HSJ'
```

2. To be able to re-use these aliases every time you login to dumbo, you can append the above lines to the end of your .bashrc file (full path: /home/your-netID/.bashrc).

    To edit a file from the command line, you can use any of the built-in text editors --- vi, emacs, or nano --- for example, type `vi .bashrc` (in your home directory). Of these three editors, nano is probably the easiest to learn, but has less advanced functionality. Emacs and vi have more advanced features and will take a few weeks to master. There are many tutorials and how-tos for these programs online (google for them).

3. Now type

```
source .bashrc
```

to create the aliases. Bash 'sources' .bashrc automatically at login, so you won't have to type this command again.

Step 3: Running a Hadoop Job

In this part, we will run an example MapReduce program on Dumbo using Hadoop-Streaming. Log in to Dumbo by following instructions above.

The example program we will run is a wordcount program. We give a large data file as input ("book.txt"), and the program counts the number of times each word occurs in the text.

1. First, we will create a folder in our home directory to store the files and output. Type

```
mkdir -p $HOME/example
```

2. Now navigate to the folder you just created:

```
cd $HOME/example
```

3. Copy the example into this directory:

```
cp -r /share/apps/examples/hadoop-streaming .
```

4. Look at the input files. To see the contents that were copied to the folder, type `ls`. You should see the input data "book.txt", a README file, and a directory called "src". This directory is where the code is. You can look at the code for the map function by typing `cat src/mapper.py` and `cat src/reducer.py`

5. Copy the input text file to HDFS: we need to move the book.txt file to the HDFS file system. To do this, type:

```
hfs -put book.txt
```

6. Make sure to clear out output from previous runs (otherwise your job will fail as HDFS will not overwrite the existing output file):

```
hfs -rm -r example.out
```

7. Run the Hadoop streaming job:

```
hjs -numReduceTasks 2 -file
/home/your-netid/example/hadoop-streaming/src -mapper
src/mapper.sh -reducer src/reducer.sh -input
/user/your-netid/book.txt -output /user/your-netid/example.out
```

8. After the job has finished, move the output from HDFS to your home directory. If you type

   `hfs -get example.out`

   you will now have a directory called 'example.out' in the current directory. To see the contents of this directory type `ls example.out` . Notice that it contains two files ("part-00000" and "part-00001") - this is because 2 reduce tasks were used. To view these files, you can type, for example, `cat example.out/part-00000`

   If you want the output in one file, Hadoop can merge it for you. Type

   `hfs -getmerge example.out examplemerged.out`

   You should then have a file called "examplemerged.out" which contains all the output. To view it type `cat examplemerged.out`

9. Now we will move "examplemerged.out" from our home directory on dumbo to our own local machine.

   On MacOS, open a new terminal window. Type (as one line)

   ```
   scp
   your_netid@dumbo.es.its.nyu.edu:/home/your_netid/example/hadoop-streaming
   /examplemerged.out .
   ```

   On Windows, run cmd.exe. Navigate to the folder where you saved pscp.exe (download this from the same site you downloaded PuTTY). Type

   ```
   pscp
   your_netid@dumbo.es.its.nyu.edu:/home/your_netid/example/hadoop-streaming
   /examplemerged.out .
   ```

   The file examplemerged.out is now on your local computer.

   Deliverable: (Due Monday, 1/30/17 at 6pm) Submit your examplemerged.out file to the Lab1 assignment on NYU Classes.

Part 3: Amazon Web Services (AWS)

We will lastly set up an AWS account. Feel free to use an existing account if you have one.

1. Sign up for an AWS account: http://aws.amazon.com
2. Sign up for AWS Educate (you will need this for $$ credits): https://aws.amazon.com/education/awseducate/apply/
3. Generate a public/private key-pair. Follow the instructions at: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair
   For windows users, you need to convert your .pem file to a .ppk file. Follow instructions at http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html#putty-private-key