ANLY 502 Spring 2016
Georgetown University
Problem Set #3 — Due February 19, 2016

Amazon AWS and Joins

This problem set beings with an exploration of Amazon Web Services. You will create and shut down machines in Amazon's cloud, access data sets larger than you can run on your personally owned computer, and run a massive mapReduce job. You will complete the "join" extra credit from PS02[1]

This problem set is assigned on February 8th and is due on February 19th at 9pm. Always check Blackboard for the most up-to-date information regarding this assignment.

START THIS PROBLEM SET NOW!!!
DO NOT WAIT UNTIL THE WEEK THAT IT IS DUE!

Before 11pm on Thursday, February 11th, you should:

- *Verify* that you can log into AWS, create an EMR cluster, run a simple command, and shut down the virtual machine.
- *Create* a private git repository and merge in changes from the course git repository.

When you shut down a server at AWS, everything on the server's virtual disks is lost. You must therefore come up with an approach for preserving your files. The recommended easy way to do this is to create your own git repository for your homework. You will *clone* your personal git repository and then *add the course git repository as a second remote.* To save your work, you will *commit* your changes to your local repository and then *push* your changes to your server.

You can create a private git repository at BitBucket and follow the instructions there.

1. Create a BitBucket account at https://bitbucket.org/.
2. Select Repositories ;-> Create New Repository. Make it private. **This example assumes that your repository is named "student."**
3. Click on the ⤓ icon to get the clone string. If you clone with HTTPS, you can use your BitBucket username and password. If you want to use SSH, you will need to register your SSH public key. (This is more secure, but harder to work within a VM.)
4. Go to the common instructions below.
5. You may need to install git on your EMR cluster. The command to do this is:

```
$ sudo yum install git
```

On your server, clone your git repository:

1. $ git clone your-repository-rul

[1] Students who did the extra credit on PS02 should update and resubmit their work according to the new submission specification.

2. `$ cd` ***`student`***
3. `$ git remote add anly502 git@bitbucket.org:simson_garfinkel/anly502.git`
4. `$ git pull anly502 master` # this applies the commits from the course repository #'master' branch to your master branch.

Congratulations! You have now have the course commits in your repository. You can make whatever changes you want. When you want to back up what you have done, you want to commit your changes and send them back to your personal repository:

1. `$ git commit -m “your log message goes here” -a`
2. `$ git push`

Note: You can repeat these steps in your Cloudera VM and use git to share your solutions between your VM and your AWS servers.

If you are using HTTPS with password verification, you will need to provide your password on every push. If you are using SSH, then you must register your SSH public key. Although you can create a public/private keypair for every VM that you create, a better approach is to create one for your laptop and then to enable SSH Agent Forwarding. From Mac or Linux, you do this by adding the “-A” flag when you SSH to a remote system.

Part 1: Getting Started with EC2

Note: To log in to your EC2 cluster, you will need to have an SSH client on your computer, create a public key, and upload the public key to Amazon’s control panel. If you are using a Mac, you have a client built-in. If you have Windows, your *easiest* approach will be to use the SSH client that is built in to your Cloudera VM. You can also download the putty SSH client from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.

Full instructions for creating and logging into your EC2 and EMR hosts can be found in:

- Amazon Elastic Compute Cloud, User Guide for Linux Instances. [PDF] [HTML]
- Amazon Elastic MapReduce, Management Guide. [PDF] [HTML]

Before you get started, please be sure that you have created an AWS Access Key and Secure Access Key:

- Go to AWS Security Credentials and create an access key.
- Copy your Access Key ID and Secure Access Key into a safe place.
- When you log into an EC2 instance, run “aws configure” to make sure that your access keys are in place. If they are not, provide them!

ANLY 502 Spring 2016
Georgetown University
Problem Set #3 — Due February 19, 2016

Please edit the file PS03/**answers.txt** and to include your answers to the questions below:

1. Determine the per-hour cost for the following clusters. For each answer provide a tuple consisting of the name of the cluster and the per-hour cost of the entire cluster. For example, if you were to build the cluster with 4096 t1.micro instances, you would provide (t1.micro, 0.013).
   a. A 4096-core cluster with the maximum memory per core allowable by Amazon.
   b. A 4096-core cluster with the maximum memory per instance allowable by Amazon. (Large memory per instance is useful for problems that need fast access to data structures that are relatively static.)
   c. A 4096-core GPU-accelerated cluster.

Remember: when you log into an EC2 Linux AMI instance, use "ssh ec2@hostname"

2. Create an EC2 t2.micro server in the US East (N. Virginia) region running Amazon Linux AMI 2015.09.1 micro server in the free tier.
   a. At what time (in ISO8601 format) did you launch the instance?
   b. Log in to the instance and type "date -In" to print the current time in ISO8601 format. Provide the result.
   c. Type "uptime" to find out how long the instance has been update. Provide the output.
   d. Provide the number of seconds that your instance took to boot. This is the number of seconds between times "a" and "b", minus "c" seconds.
   e. Show the amount of disk space installed by providing the output of the "df –h" command.
   f. Which version of Java is running on your server? Type "java -version" and provide the version number.
   g. Go to the AWS EC2 console. In which Availability Zone is your server running?
   h. What is your Instance ID?
   i. Go to the "Security Groups" section of your EC2 control panel and modify your server's Inbound security group so that it can receive ICMP packets from any host. (That is, from 0.0.0.0/0)[2]. Now back at your instance, type "traceroute www.qwest.net". Provide the results.

Shut down the t2.micro instance

[2] You cannot ping or traceroute from Amazon EC2 without opening up the firewall to allow the ICMP packets to return. For more information, see http://serverfault.com/questions/471466/how-to-ping-traceroute-an-aws-elb. For information about Traceroute, see https://en.wikipedia.org/wiki/Traceroute. For information on the Ping network utility, see https://en.wikipedia.org/wiki/Ping_(networking_utility)

Part 2: Getting started with EMR

Remember: when you log in to an EMR cluster, user "ssh hadoop@hostname"

Here are some key differences between Hadoop on Amazon EMR and on the Cloudera Quickstart VM:

- EMR uses Python 2.7 by default; Cloudera uses Python 2.6 by default.
- EMR pre-installs pip, so you don't have to.
- EMR's Hadoop user is "hadoop", whereas Cloudera's is "cloudera."
- EMR stores Hadoop files in hdfs:///user/hadoop.
- You still need to type: `export HADOOP_HOME=/usr/lib/hadoop`

Test mrjob with a simple job. If you get memory errors, you are not using a large enough nodes in your cluster. [3]

3. Create an EMR cluster with 1 Master and 2 Core nodes. (In preparing this exercise, we used a m3.xlarge node with a spot price of $0.05/hour. Each node has 10G root partition, a 38G /mnt partition, and a 38G /mnt1 partition. The root partition is used only for software; HDFS is stored on /mnt and /mnt1. An EMR cluster with 2 Core nodes can therefore store 38x2x2=152G in HDFS, since replication is disabled with 1 or 2 Core nodes.)
   a. What are the public IP address(es) associated with your instances? Separate multiple addresses with commas.
   b. What are the private IP address(es) associated with your instances?
   c. Log into the cluster and run the command "hdfs dfsadmin -report" to get a report on each of the HDFS nodes. Provide the output of the command.
   d. Run the command "yarn node -list -all" to get a report on each of the Yarn nodes (other than the head node). Provide the output of the command.

Install mrjob on EMR with the command:[4]

```
$ sudo pip-2.7 install mrjob
```

Optionally, install EMACS on your cluster's head node:

```
$ sudo yum install emacs
```

[3] By default, EMR will use m3.xlarge nodes, which are $0.266 per hour. Don't use smaller nodes to save money—they will not run Hadoop properly. Instead, you can save money by requesting spot pricing.

[4] If you want to be clever, you can create a "bootstrap" script that installs mrjob when the cluster starts up. Note that mrjob only needs to be installed on the head node, not on each of the worker nodes.

4. In this section we will be testing mrjob that you just installed, but this time with *The Count of Monte Cristo*, by Alexandre Dumas, which was downloaded from Project Gutenberg.

Download The Count of Monte Cristo[5] from http://www.simson.net/anly502/pg1184.txt and determine the 10 most frequent words using the wordcount_top10.py program in the PS03 directory.

Your commands for running the top-10 will be:

```
$ python wordcount_top10.py -r hadoop pg1184.txt
```

a. Provide your top-10 results in a file called **wordcount_top10.txt**
b. Provide the output of the HADOOP Job Counters in a file called **count_counters.txt.** Notice that how many map and reduce tasks were run.

A number of books from Project Gutenberg have been placed into the Amazon S3 bucket `s3n://guanly502/`[6] with the prefix `gutenberg/` for this course.

5. Using the "aws s3 ls" command, list the contents of the bucket place them in the file **guanly502_gutenberg_ls.txt**

6. Run the top-10 program on the contents of the Gutenberg files in the guanly502 bucket. Place the contents in a file called **guanly502_gutenberg_top10.txt**

Part 3: Joins, using your Cloudera VM

There are two approaches for doing joins with MapReduce. One approach is to perform the Joins in the reducer by sending through a both the left and the right side of the join and relying on the reducer to perform the join (to match things up). The second is to store one table in memory (either in the mapper or the reducer) and perform the join as the data streams through. You can only do this if one of the tables is small enough to fit in memory.

The MaxMind corporation maintains a database that maps IP addresses to geographical locations. Their GeoLite2 database is a Creative Commons version of their database. It's a bit less accurate than their commercial database, but good enough for us.

The MaxMind datasets is at http://dev.maxmind.com/geoip/geoip2/geolite2/, but you do not need to download it at this time.

[5] $ wget http://www.simson.net/anly502/pg1184.txt

[6] Hadoop on AWS uses the prefix s3:// to specify HDFS blocks stored in S3 and s3n:// for actual files stored in S3. However, in some cases you may specify s3:// to access files.

We have created a table of IP addresses and Countries based on the most popular IP addresses in the forensicswiki log files. That dataset is distributed as a list of <IPADDRESS,COUNTRY> pairs.

7. Perform a join of the forensicswiki data from PS02 and the MaxMind data and produce a list of all the log file entries that can be geolocated. Create a Hadooop counter and count the number that cannot be counted. The format of the lines should be:

   ```
   Location    "Original Web Log Line"
   ```

   Name your program **join1.py** and provide the first 50 lines of output of your joint job in a file called **join1.txt.**
8. Modify your program so that it computes the number of hits per country. Call this revised script **join2.py.** Run it and put the output in **join2.txt**. **You will need a two-step mapreduce job.**
9. Modify your program so that it computes the top-10 countries. Call this revised script **join3.py.** Run it and put the output in **join3.txt**. You will need a three-step mapreduceJob.

Sample code to get you started on each of these problems can be found in the git distribution in the PS03 directory.

Part 4: Joins, using EMR and S3

For this last part, we have uploaded the Apache access logs for the forensicswiki.org website for the year of 2012 to Amazon S3. You will find them in the bucket and prefix: s3n://guanly502/ps03/forensicswiki/2012.

Many Hadoop implementations allow you to directly access files stored in Amazon S3 by supplying the URL.

10. To verify that you have access to the S3 data storage, provide the first 50 weblog lines chronically that downloaded the URL `/wiki/Main_Page.` We have provided a skeletal program that does this for you called first50.py. Modify it as necessary.

    Please provide:

    first50.py — The program which displays the first 50
    first50.txt — The first 50 entries chronically

11. Perform a join of the forensicswiki 20013 data and the MaxMind data and produce a list of all the log file entries that can be geolocated. Create a Hadooop counter and count the number that can and cannot be counted. We have given you a program called **first50join1.py** that will (eventually) produce output with the first 50 lines of the joined weblog.

The format of the lines should be:

```
 “First50Geolocated"       [Date,Location,”Original Web Log Line”]
```

Please provide:

first50join1.py — The program which displays the first 50 lines joined
first50join1.txt — The output of the program.

12. Modify your program so that it computes the number of hits per country and sorts the results by country. You will need a three-step mapReduce job to do this.

    Please provide:

    sortedjoinbycountry.py — The program which produces an output of the number of hits per country, sorted.
    sortedjoinbycountry.txt — The output

Part 5: Getting Started with EBS

This last problem involves an analysis of the Wikipedia Extraction Dataset, a 66GB dataset that is available on Amazon EBS. For information about the dataset, please see:
https://aws.amazon.com/datasets/wikipedia-extraction-wex/?tag=datasets%23keywords%23encyclopedic

13. Follow these steps:
    a. Create an EBS volume from the snapshot. Review the class slides and the Amazon documentation for information on how to do this.
    b. Create a directory “/wikipedia” on the Master instance. You will need to do create the directory as root using the **sudo** command (e.g. **sudo mkdir /wikipedia**).
    c. Mount the EBS volume at **/wikipedia**
    d. Create a HDFS directory **hdfs://user/hadoop/infiles**
    e. Copy the files in the **/wikipedia/rawd** directory to the HDFS directory /user/hadoop/infiles/
    f. The file **freebase-wex-2009-01-12-articles.tsv** is a list of Wikipedia articles as of 2009. Each article has a name, a date of most recent modification, and the text as an XML block. You do not need to parse the XML block — you are only considering the article date.
    g. Write an mrjob script that computes the number of articles modified each month. Provide the script, the table of results (sorted by date), and a graph.

You should provide for us:

wikipedia_stats.py — Your mrjob program that computes the number of articles modified each month.
wikipedia_stats.txt — Your output, a file consisting of the Date and number of modifications for each month
wikipedia_stats.pdf — Your graph

What to turn in

Please turn in a ZIP file[7] containing the following files:

answers.txt
wordcount_top10.py
wordcount_top10.txt
guanly502_gutenberg_ls.txt
guanly502_gutenberg_top10.txt
join1.py
join1.txt
join2.py
join2.txt
join3.py
join3.txt
first50.py
first50.txt
first50join1.py
first50join1.txt
sortedjoinbycountry.py
sortedjoinbycountry.txt
wikipedia_stats.py
wikipedia_stats.txt
wikipedia_stats.pdf

NOTE 1: In PS03/ you will find a script called **validator.py** that will validate your homework ZIP file. Eventually it may even grade it! "git pull" to get the latest version. It is ***strongly*** recommended that you validate your ZIP file before turning it in.

[7] Only submissions in ZIP format will be accepted. Files ending ".txt" or ".py" ***MUST*** be in Plain Text (Unicode UTF-8 or ASCII); RTF will not be accepted. Files ending in ".pdf" ***MUST*** be in Adobe Portable Document Format (PDF). Other formats will not be accepted.

Want more practice? Try these:

- Rework problem #10, but output the articles that have had the most modifications.
- Rework problem #10, but produce your output using Pig.
- Rework problem #10, but produce your output using Hive.