Georgia Tech
CSE6242 / CS4803-DVA: Data and Visual Analytics
Spring 2013, Polo Chau

Homework, Due: April 26, 2013, 1:30PM

You will try out PIG (http://pig.apache.org) and HIVE (http://hive.apache.org)
You will submit a single archive file; detailed submission instructions are in the last section.

Task 0:

Familiarize yourself with AWS (Amazon Web Services). Read the AWS Setup Guidelines we provided to set up your AWS account and redeem your free credit ($100). The pricing for various services provided by AWS can be found at http://aws.amazon.com/pricing/. The services we would be primarily using for this assignment are the Amazon S3 storage, the Amazon Elastic Cloud Computing (EC2) virtual servers in the cloud and the Amazon Elastic MapReduce (EMR) managed Hadoop framework. Play around with AWS and try to create MapReduce job flows (not required, or graded) or try the sample job flows on AWS.

The questions in this assignment will ideally use up only a very small fraction of your $100 credit. AWS allows you to use up to 20 instances total (that means 1 master instance and upto 19 core instances) without filling out a "limit request form". **For this assignment, you should not exceed this quota of 20 instances**. You can learn about these instance types by going through the extensive AWS documentations.

Task 1: Analyzing large amounts of data with MapReduce/Hadoop.

You will have access to a fraction of the Google books *n*-grams dataset (full dataset is here) and you will perform some simple analysis on this dataset. An '*n*-gram' is a phrase with *n* words. This dataset gives us a list of all n-grams present in the books on books.google.com along with some statistics.

For this assignment, you will only use the Google books bigrams (2-grams), which we have already uploaded to this Amazon S3 bucket/directory:

s3n://cse6242-gtcse-data/gb-bigrams/
The files in this directory are stored in a TAB separated format. Each line in a file has the following format:

bigram TAB **year** TAB **match_count** TAB **volume_count** NEWLINE

An example for 2-grams (or bigram) would be:

```
I am       1936  342   90
I am       1945  211   10
very cool  1923  500   10
very cool  1980  3210  1000
very cool  2012  9994  3020
```

This tells us that, in 1936, the bigram 'I am' appeared 342 times in 90 distinct volumes (books). In 1945, 'I am' appeared 211 times in 10 distinct volumes. And so on.

What we want you to do is the following:

1. For each unique bigram, compute its average number of appearances per year of occurrence (we won't be considering `volume_count` at all). For the above example, the results will be the following:

   ```
   I am       (342 + 211) / 2 = 276.5
   very cool  (500 + 3210 + 9994) / 3 = 1568
   ```

2. Output the **5 bigrams** with the highest average number of appearances per year of occurrence along with their corresponding average sorted in the descending order. If multiple bigrams with the same average, write down any ones that you like (that is, break ties as you wish). For the above example, the output will be the following (the output can be comma-separated as shown below or tab-separated):

   ```
   very cool,1568
   I am,276.5
   ```

This is a fairly simple task. However, the sheer size of the data necessitates the need for large scale computing. We want you to solve this problem in **two** ways:

1. Write a PIG script to perform this task on the Amazon EC2 cluster and save the output.

   One way of doing this is to use the interactive PIG shell that EMR provides to perform this task from the command line (`grunt`). In this case, you can copy the commands you used for this task into a single file to have the PIG script and you can copy the output from the command line into a separate file.

   To load the data from the *s3n://cse6242-gtcse-data* bucket into a PIG table, you can use the following command:

```
grunt> bigrams = LOAD 's3n://cse6242-gtcse-data/gb-bigrams/*' AS
(bigram:chararray, year:int, mc:int, vc:int);
```

2. Write a HIVE script to perform this same task on the Amazon EC2 cluster and save the output.

   Similar to the previous part, you can use the interactive HIVE shell that EMR provides to perform this task from the command line (`hive`). In this case, you can copy the commands you used for this task into a single file to have the HIVE script and you can copy the output from the command line into a separate file.

   To load the data from the *s3n://cse6242-gtcse-data* bucket into a HIVE table, you can use the following command:

```
hive> CREATE EXTERNAL TABLE bigrams (bigram STRING, year INT,
match_count INT, volume_count INT)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    > LOCATION 's3n://cse6242-gtcse-data/gb-bigrams/';
```

While working with the interactive shell (or otherwise), you should first test on a small subset of the data instead of the whole data (the whole data is over 125GB). Once you think that your PIG/HIVE commands are working as desired, you can start them up on the complete data in the *s3n://cse6242-gtcse-data* bucket and ...wait... since it will take some time.

Deliverables.

1. **[15 points]** The PIG script for the task named **hw3-*{GT-USERNAME}*-pig.txt** and the corresponding output in **hw3-*{GT-USERNAME}*-pig.out** (the output can be comma-separated or tab-separated).
2. **[15 points]** The HIVE script for the task named **hw3-*{GT-USERNAME}*-hive.txt** and the corresponding output in **hw3-*{GT-USERNAME}*-hive.out** (the output can be comma-separated or tab-separated).

Submission details

Submit the deliverables as individual files on the t-square submission site. Please specify the name(s) of any students you have collaborated with on this assignment, using the text box on the T-Square submission page for this assignment.

Please adhere to the naming convention specified here. In case your submission does not comply with this, *it will be returned to you ungraded*. You would need to resubmit in the specified form to be graded. While your resubmission will be graded, the delayed resubmission will be counted as a late submission.