Lab 08 Specification – Exploring Concurrency through Multithreading and Cloud computing.
Due (via your git repo) no later than 8 a.m., Tuesday, 11th December 2018.
50 points
Last Lab for the semester

Lab Goals

The goal for this Lab is to explore Concurrency through the use of Multithreading and Cloud computing. As part of this lab, you would also practice provisioning and utilizing the Amazon AWS cloud instances.

Assignment

- Part 1: Create a word count implementation using multiple threads, that is executed on one machine. Given an input file [hamlet.txt file in the lab repo], using multiple threads, count the total number of occurrences of each word and generate a word count statistics report. For example; the output of your program should be similar to:

  ```
  think 28
  says 31
  approve 24
  ```

  Note: The result shown above is just a sample. The occurrences of words and the list of words do not accurately project the results of "hamlet.txt" file.

  An important part of this implementation is that you are required to use multiple threads in your code. A good starting point for implementing this part of the lab is to look at the Multiple Thread Counter example discussed in class. The source code is available in the coding folder and you may start with this version of the code and gradually add new code snippets and/or modify the existing code in the file to construct your implementation that handles the requirement outlined above.

  Additionally, you may refer to online resources, in order to support your lab work. Make sure to cite any resources that is directly or indirectly helpful to produce your lab work. I am expecting you submit one or more Java source code files and/or other external dependencies (such as Jar files) that might be needed to execute your Java programs. Create a directory called src within part01, and upload all your source code files there.

- Part 2: Create a variation of the earlier word count implementation, by setting up the Cloud platform. By setting up this variation, we will take a targeted initiative that would achieve concurrency through the use of Cloud computing. At this point, I expect everyone in the class to know:

  1. how to create the Amazon AWS EC2 Cloud instances
  2. how to setup the access [login] to those Amazon AWS EC2 Cloud instances
  3. how to install Java on those Cloud instances; and execute Java programs both through physical and remote connectivity to the Cloud.

  Although we had talked about this briefly during our earlier discussions, I will discuss in today's lab, how to copy files from one machine to another through the use of SCP both inside and outside the Cloud.

  Okay, well here are some of the details connected to the implementation of handling the word count functionality in the Cloud platform. The implementation details are intentionally left at a high level and I expect the students to figure out the inner details of the implementation by thinking design principles from looking through one or more online resources, talking to peers by doing the appropriate discussions, and finally asking questions to the [TA's, and/or the Professor] during the lab session. The implementation details are spelled out below:

  – Identify the frequency of each word as (Part 01) through the use of counter.

- The code takes a command line argument; and simply does the word count implementation outlined above based on their starting character.

  For example if the command line argument is A-E then only words that starts with character A, B, C, D, E would be considered.
- Setup a range for the cloud based on the word starting character; A - E is for Cloud instance 01; F - J is for Cloud instance 02; K - O is for Cloud instance 03; P - T is for Cloud instance 04; U - Z is for Cloud instance 05;
- Move data and code file to all your Cloud instances
- Finally, execute the Java program on all the Cloud instance through remote execution and show the screenshots of the results received in the different cloud instances using a PDF document.

Required Deliverables

Please submit electronic versions of the following deliverables to your GitHub repository by the due date:

1. Properly completed and commented source programs for Part 1, 2.
2. A Read me (text or PDF) file, that states clearly how to run the program and a self reflection section that includes either the list of challenges faced and/or list of things liked and disliked in this lab.
3. A PDF file that shows the screenshot of the outputs generated from your Part 02 implementation.

Grading Rubric

1. If you complete [Part 1] completely as per the requirement outlined above, you will receive 20 points.
2. If you complete [Part 2] completely as per the requirement outlined above, you will receive 30 points.
3. If you fail to upload the lab solution file to your git repo by the due date, there will be no points awarded for your submission towards this lab assignment. Late submissions will be accepted based on the late submission policy described in the course syllabus. It is the student's responsibility to communicate to the professor if it is a late submission. If the student had not communicated in advance about the late submission, the lab work shall not be graded as such.
4. Partial credit will be awarded, based on the work demonstrated in the lab submission file.
5. If you needed any clarification on your lab grade, talk to the Professor. The lab grade may be changed if deemed appropriate.