Lab 3 Specification – Exploring Networked Services in Cloud
Due (via your git repo) no later than 2 p.m., Monday, 28th September 2020.
50 points

Lab Goals

- Implement networked services in Cloud.
- Develop computing blocks on different machine(s) hosted in private and public Cloud platforms.

Summary

You will do a few hands on exercises to connect MultiThreading with the Networked Services. So far, these are the two major concepts discussed in this course. It is the right time to combine these two items, and assess the performance advantage in such a scalable computation. Additionally, we will watch a video clip, which is a segment of an interview with the famous scientist, and one of the inventors of the C programming language, Brian Kernighan. We will take up an additional programming exercise to further solidify our knowledge in C programming.

Learning Assignment

If you have not done so already, please read all of the relevant ”GitHub Guides”, available at https://guides.github.com/, which explains how to use many of the features that GitHub provides. In particular, please make sure that you have read guides such as ”Mastering Markdown” and ”Documenting Your Projects on GitHub”; each of them will help you to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, you should also read

- **Basic Networking reading material in the class repository.**

Assignment Details

Now that we have discussed some fundamental principles behind networking and development of networked services, it is now time to implement some challenging requirements of combining MultiThreading and networked services.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share code files and/or other implementation details. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.
2. Submitting a copy of the other’s program(s) is strictly not allowed. Please note that all work done during laboratory sessions will be an opportunity for students to learn, practice, and master the materials taught in this course. By doing the work individually, and in teams, students maximize the learning and increase the chances to do well in other assessments such as lab assignments, skill tests, projects, etc.

At any duration during and/or after the lab session, students are recommended to team up with the Professor and/or the Technical Leader(s) to clarify if there is any confusion related to the items in the lab sheet and/or class materials.

Section 1: Cloud Computing

This section is worth 15 points. The points breakdown is provided below:

- Task 1 = 15 points, a maximum of 3 points awarded for each question.

John Easton is an IBM Distinguished Engineer and lead Cloud Advisor in Europe. He is internationally known for his work helping commercial clients exploit scale distributed computing infrastructures, particularly those utilizing new and emerging technologies. Over his time at IBM, John has led initiatives around hybrid systems, computational acceleration, grid computing, energy efficiency and mission-critical systems (taken from video description). This video is a segment from the series of talks given at a TEDx event in a TED conference. In this video, John gives his view on recent technology developments and talks about the impact that cloud computing will have in the future on everyday life. In this section, we will watch a video and reflect on the points discussed in the clip. This reflection is instrumental to further advance our understanding of Cloud Computing and in general to foster the learning from our recent discussions on Networked Services and understand the importance of Cloud. To complete this part, it is required to do the following:

- **Task 1:** Watch this a little more than 10-minute video, by using the link below:

  `https://www.youtube.com/watch?v=8H3WaMzDiTo`

  After watching the video, each member of the team should create an individual markdown file and name it as `video-reflection`. Add your first initial and last name to the end of the file name. For example: `video-reflection-amohan.md`. In this file, provide a detailed description of the questions presented below:

  1. How does the speaker connect Robin with Cloud Computing?
  2. What did the speaker say about the financial stability provided by the Cloud?
  3. How does the speaker compare owning an asset to the Cloud?
  4. Why does the speaker compare Mainframes with the Cloud?
  5. The famous quote by Ken Olsen (in 1977) ”There is no reason anyone would want a computer in their home.” was discussed in the video. Why was it discussed? How does the speaker connect this quote with the modern day Cloud Computing?

Section 2: Election Poll Analyzer

This section is worth 35 points. The points breakdown is provided below:

- Task 2 = 5 points
- Task 3 = 5 points
- Task 4 = 5 points
- Task 5 = 5 points
- Task 6 = 10 points
- Task 7 = 5 points

In this section, we will develop a scalable tool to analyze the election poll results and find out the percentages of each party. To develop such a scalable tool, it is important to think through and realize the implementation details of a Diamond Networked Service discussed during our recent class session. Here we assume that three Cloud instances are used in the program. These instances, include two AWS instances, and one OpenStack instance. The sequencing of the operations in the Cloud are provided below:

1. Local machine identified by the node 'a'.
2. OpenStack instance identified by the node 'b'.
3. AWS instances identified by the node 'c', and 'd'.
4. Execute the loader program in machine 'a'.
5. Push (Upload) the output file from the previous step to both nodes 'b', and 'c'. This step should be done using MultiThread concurrently.
6. Execute the analyzer code blocks in machines 'b', 'c', and push the respective output files to the nodes 'd'. This step should be done using MultiThread concurrently.
7. Once both threads complete their execution, execute the analyzer code block on machine 'd' to generate the final result.
8. Finally, pull the result from machines 'b', 'c', and 'd' machines and download those to the local machine. This step should be done using MultiThread concurrently.

Note: In this part, we will learn how to parallelize the development of networked services in the Cloud. It is required to implement the sequencing outlined above by completing the tasks provided below:

- To submit these tasks outlined in this section, create a new folder called **poll_cloud**. A starter-code in **diamond-build** discussed in class is provided in the repository for reference.

- **Task 2:** Develop the right set of code in the Driver program, which is in machine 'a', to generate a file with a randomized sequence of numbers. Assume that these numbers correspond to a special **poll_code**, which represents a translation of the voter's poll results. To further clarify, there is one **poll_code** for each voter. In this assumption, if the poll_code is odd numbered, then the corresponding voters poll result suggests Democratic winning, and Republican winning otherwise. The goal of the analyzer tool is to find out the distribution of the poll result between the parties and classify the percentages for each party. It is highly recommended to utilize the **Loader** program provided in the previous lab and practicals, as a starter-code to generate randomized numbers in a text file. Note: The Loader program has an implementation that allows the configuration of the total number of randomized inserts to the file. This configuration is critical to control the size of the file and to conduct experiments in Task 7.

- **Task 3:** Develop an instance block code, named **code_b**. This instance block code should read through an input file and find out the total number of even numbered **poll_codes**. The result should be written to an output file named **b_output.txt**.

- **Task 4:** Develop an instance block code, named **code_c**. This instance block code should read through an input file and find out the total number of odd numbered **poll_codes**. The result should be written to an output file named **c_output.txt**.

- **Task 5:** Develop an instance block code, named **code_d**. This instance block code should read through the two input files, that is the total number of even and odd numbered **poll_codes**. After reading through these files, the instance block should output the percentage of the corresponding parties. Note: the total number of people participated in the polls is available through the input files. This output should be written to an output file named **d_output.txt** file. A sample output is shown below for your reference:

  Republican: 44% Democratic: 56%

- **Task 6:** Develop the right set of code in the Driver program to set up the coordination between the nodes in the Diamond Networked Service. Implement the appropriate `config.xml` file, to ensure correct configurations for the tool execution using the Cloud. Micro details for this part is purposefully left out from the specification, to provide an opportunity for the learner to think through the design issues and plan out the required sketch to address the issues in setting up the co-ordination between the nodes. Add a **Readme** file that clearly outlines the steps to be followed to execute the Driver program.

- **Task 7:** Finally, develop two charts, using the chart latex template file provided in the previous lab (lab-1) repository. These charts should include a single threaded and multithreaded experimental results. In these charts, clearly plot the running time versus **(5)** different file sizes. There is no hard and fast rule given to test out the file sizes. Based on the size of the file, the processing time taken can vary across machines. Note: Machine (a) is local to your machine, and creates the source input file and orchestrates the internal flow of the Diamond Setup. It is recommended that you try out different file sizes. For example: 5,10,15,20,25 MB. Make sure to submit your final charts using the `charts.tex` and `charts.pdf`. In these charts, a general observation to make is that, the running time with multiple threads should make things faster. Add a reflection markdown file, using the name `reflection`. In this file, summarize your understanding of the experimental results and logically reason why you are seeing such experimental results.

Submission Details

For this assignment, please submit the following to your GitHub lab repository.

1. **video-reflection** markdown file.
2. upload of **poll-cloud** code folder.
3. upload of **reflection** file.
4. upload of **charts.tex** and **charts.pdf** files.
5. It is recommended to upload a readme file, with the details that you would like the Professor to know while grading the work. For example, it may be reflection of your experience in the lab by highlighting some of the challenges faced and a brief mention of how team members addressed those challenges while implementing this lab. The readme file may also include a brief mention of any details that one should know about executing your program and what to expect during the execution.
6. It is highly important, for you to meet the honor code standards provided by the college and to ensure that the submission is made before the deadline. The honor code policy can be accessed through the course syllabus. Make sure to add the statement ”This work is mine unless otherwise cited.” in all your deliverables such as source code and PDF files.

Grading Rubric

1. Details including the points breakdown are provided in the individual sections above.
2. If a student needs any clarification on their lab credits, it is strongly recommended to talk to the Professor. The lab credits may be changed if deemed appropriate.