ASSIGNMENT – 4 (miniProject)

SOCKET PROGRAMMING

School of Computing, National University of Singapore

This is a Group Project with group size of 4 members per group. As informed early, pls register your groups @ *Canvas->People->A4ProjectTeam*. We strongly recommend teams to **start early**.

Learning Objectives

This assignment/project provides you a chance to learn about details of Hyper-Text Transmission Protocol-HTTP, Secured HTTP (HTTPS) and develop services using socket programming.

Parallel Web Crawler

A web crawler (or web spider) is a program that retrieves and stores pages from the Web, commonly for a Web search engine (such as Google). A parallel crawler that runs multiple processes/threads in parallel. In this assignment you will implement a parallel web crawler using python, java, c++) that browses the WWW automatically by sending HTTP requests to many web servers in parallel. The crawler should start with a few web servers/web pages and should recursively discover more links (to more pages/servers).

Requirements

- The crawler should store the following in a database or text file and should display them.
    - URL of the web pages visited. [You can start with some known URLs in the text file/ DB and accumulate more URLs in the same text file/DB]. As parallel crawlers are accessing the text file, the access should be coordinated with proper *access-lock* mechanisms.
    - Response time of the servers [time from sending a request to receiving the reply]
    - IP addresses of the servers with region (IP geolocation)
    - Open Ended Requirement: Collect, analyses and report your views on any one another useful data related to 'Education' or 'eSports' or 'Video Games' or 'Cybersecurity' or 'Other interesting topics. The report should be in one or two pages, including references. [Examples: Popularity of few selected video games in Singapore or another region; Popularity of a video game in two or more different regions; Popularity of eSports in different regions or different universities; Regions discussing/reporting about a specific teaching and learning method such as Practice Based learning or AI in education; Regions discussing/reporting about a specific cyber-attack].

- Try to keep the request rate of your crawler low by introducing some delay between requests. Sending request to same web server several time may result in misinterpreting the crawler as a DoS attack. If needed, you may run the crawler for long time.
- The crawler should not make more than one request to the same web page. (Only one of the threads/processes should visit. This implies the use of a common/shared database of URLs.
- You can use high-level socket libraries (such as, *http.client* or *requests* libraries in python; *HttpURLConnection* or *HttpsURLConnection* classes in Java) for HTTP and HTTPS connections and any high-level library for parsing the contents. However, you should not call/use any existing web crawler class or tool in your application.
- Crawler should stop after some time. (You can use any reasonable strategy to stop).
- You are allowed to use any free/open-source classes/tools/packages for conversion (such as HTML to XML) and parsing.
- If you need to run your program for multiple days to collect data, you can apply for SoC VM through *mysoc->eForms->New Application->Computing Resources*. It may take couple of days for approval. Approver id: bhojan.
- Your codes should be well written and well commented.

Grading note:

- (3 marks) Starts with the initial set of URLS in a text file/DB.
- (7 marks) Coordinated access by multiple Threads to the text file/DB.
- (10 marks) Adds newly found URLs to the text file/DB.
- (5 marks) Response times to the severs, IP addresses and geolocation of the servers are printed (either in screen or in text file/DB).
- (8 marks) One page report on useful data collection/statistics (open ended requirement)
- (2 marks) Well written code with in-line comments.

Submission:

Submission Due: **Thursday 09-Nov-2023**

Submit **one ZIP/RAR file** (change filename to your *Group Number*) containing **your source code, short video demo showing your DB/text file/screen** showing new URLs found (minimum 10 new urls), one PDF report and a short text file with instructions for compiling/running your source code to '*Canvas ->miniProject-Submission*'.

If you have any question/clarification, discuss through **discord channel 'assignments**.

Bhojan Anand /NUS

'Students who approach education from a **deep-learning perspective** make significant improvements, remember what they learn, and feel empowered to make a difference. Those who take a surface approach to learning may get good grades, but they rarely benefit much in the long term'. – *from several research findings*.

Learning in depth is the key focus of CS3103. Try more beyond the questions above.

[We have a Zero Tolerance for Plagiarism Policy. If you are here only for marks/grades, just let me the grade/mark you prefer to have!]

More on deep learning approach …

Students Who Take a Deep Approach--

- *Attempt to understand material for themselves*
- *Seek rigorous and critical interaction with knowledge content*
- *Relate ideas to previous knowledge and experience*
- *Discover and use organizing principles to integrate ideas*
- *Relate evidence to conclusions*
- *Examine the logic of arguments*