ASSIGNMENT 2 and PROJECT - Client/Server Application Development

School of Computing, National University of Singapore

Learning Objectives

This assignment/project provides you a chance to learn about popular transport and application layer protocols on the Internet (Transmission Control Protocol-TCP, Hyper-Text Transmission Protocol-HTTP) and to understand how web crawlers are developed for search engines.

Parallel Web Crawler – Description

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 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).

ASSIGNMENT 2 (Individual Work) - Requirements

- The crawler should store the following in a database or text file and should display them.
    - Base URL of the web application / server
    - Response time of servers [time from sending a request to receiving the reply]
- 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 resource.
- You should do the assignment by creating the proper HTTP messages as per the RFCs and using basic Socket library. You should not call/use any existing web crawler class or tool in your application.
- You are allowed to use any free/open-source classes/tools/packages for conversion (such as HTML to XML) and parsing.
- Your codes should be well written and well commented. All exceptions must be handled.

Assignment 2 Submission

Due: Week 7 Wednesday (28 Sep 2016)

Upload the ZIP/RAR file containing all the deliverables into **"Workbin->Assign2Submission"** folder before the deadline. Rename the ZIP/ZAR file to your **Matric number**.

Demo and Interview: Week 7 Thursday & Friday (29-30 Sep) (Pls arrange for presentation time slot with the TAs – Mr Pravein and Ms Bayan).

Platform: The applications should be developed in C/C++ platform. To get in-depth ideas on socket programming and to maintain consistency in grading and , other platforms are not allowed.

Questions/Feedback: If you have any question, please put it in the IVLE-forum instead of e-mail. This will help your classmates/friends.

Assignment 2 Grading [Max Marks 50]

Please demonstrate the following to the evaluators:-

A. *Web Crawler (45 marks)*

a. *Text file with list of server name (base URL) and response time is generated.*

b. *Proper HTTP messages are constructed and sent using basic socket class*

c. *Request rate is controlled with some time-delay between calls*

d. *Crawler stops after sometime (When it stops? What strategy is used?)*

e. *The crawler is multi-threaded (or multi-process).*

f. *Each page is visited only once by the parallel crawler. (Only one of the threads should visit). This implies the use of a common/shared database of URLs.*

B. *(5 marks) Readable, well-written and properly documented code with error checking (exception handling - such as File Read/Write exception, Socket Exception…); unspecified additional features; submission on time.*

(A+B = 50)

Bonus Marks:

C. *(5 marks) Any additional feature/technique to enhance performance of crawler.*

PROJECT IDEA's (Group Work)

Modify and extend the above crawler (Assignment 2) to search and provide useful information (It is open-ended!). You can use higher-level APIs for communication and you can use any programming language (Java/C/C++/Python).

Some possible enhancements:-

- You can report any interesting statistics about the servers.
- You can read more information form each page and index based on the texts available in the page. You can build a simple special purpose search engine that uses your crawler.
- Describe a policy/architecture (you may also implement it) to build a search engine based on the principles of peer-to-peer networks with a distributed crawler.
- You can improve the crawler and make it capable of searching Rich Internet Applications (RIA) efficiently.
- You can analysis of popularity of a specific product in various regions,
- You can report important statistics about popularity of a news in news channels, social networks, micro-blogs or rss-feeds…..
- You can extract and match authors and affiliations in scholarly documents, etc.
- READ the Papers and Lecture Notes in "**workbin->project**" folder and search the web for 'state of the art crawlers' or 'goal oriented crawlers – that have specific objectives' or 'web analytics' to get more niche ideas, useful purpose that is not easily extractable using current tools….

Virtual machine:

Each group can apply for a Virtual Machine (Windows/Linux) through "mysoc" to run your project application.

Group Registration & Submissions:

- Group Size: Min 2, Max 4.
- Register groups at IVLE->Workspace->Project "CS3103 PROGRAMMING PROJECT" .
- All the deliverables indicated below are to be submitted by any one member of the Group. [That is, one copy per group].

Project Missions:

Mission 1) Proposal Submission:

Due: Week 7 Monday (26-Sep-2016)

Describe the features and their uses in 1 page (maximum 1 page).
Upload the DOC/PDF file into **"Workbin->Project Mission1"** folder before the deadline. Rename the DOC/PDF file to your **Project Group number**.

Mission 2) Prototype/Level 1 Presentation

Due: Week 10 Monday (20-Oct-2016)

a. Progress and Project Timeline Presentation- max 10 slides, using **sntboard.com**. Create your presentation in sntboard.com and share it with the class "CS3103 2016/17 Sem1" within sntboard.com (share with view-only access). You should use IVLE-login (NUSNET account) to see the "CS3103 2016/17 Sem1" group. The first slide should contain 'Project Title', 'Group Number and Team Name', 'Name of all members'.
(You may use *real-time collaborative mode* for annotating collaboratively if members are not present at same place – only text and drawing are possible in this mode; no audio support).

b. Upload the ZIP/RAR file containing the source codes into **"Workbin->Project Mission2"** folder before the deadline. [It may be separate codes without integration and UI; but, most of the functions should be in by this time]. Include a text file stating how to run your project or part of the functionalities of your project. Rename the ZIP/RAR file to your **Project Group Number**. [Add a DOC/PDF if you have made some changes to the Initial Proposal]

Mission 3) Final Version Submission:

Due: Week 12 Friday (4-Nov-2016) [or During the week-end, 6-Nov]

Upload the ZIP/RAR file containing project source codes into **"Workbin->Project Mission3"** folder before the deadline. Rename the ZIP/ZAR file to your **Project Group number**.

Demo and Interview: Week 13 (During Lecture Hour) ----- Submit a copy of your project PPT slides into "Workbin->ProjectSlides" immediately after the demo and Interview (within a day). Filename should be your **project group number.**

Bhojan Anand /NUS

'If you are poor, work. If you are rich, work. If you are burdened with un-seemingly unfair responsibilities, work. If disappointments come, work. If sorrow overwhelms you and loved ones seem not true, work. If health is threatened, work. When dreams are shattered and hope seems dead, work. Work as if your life is in peril. Its really is. No matter what ails you, work. Work faithfully. Work in faith. Work is the great remedy available for both mental and physical afflictions. All, without exception, perform work. There is no way of renouncing work altogether (always there is another work!)'. – *The Inspired Talks of Swami Vivekananda and The Gospel of Sri Ramakrishna.*