CIS 4930 Internet Networking Technologies, Spring 2024

Programming Assignment #2 – Networking Performance

Due: Please submit your source codes and report (**in PDF format)** via CANVAS by 12:50PM (**firm deadline**) on Friday, March 29th

Introduction

In this programming assignment, you will measure the timing performance and analysis of a simple client/server implementation for socket communication. Its aim is to allow students to measure socket performance with different factors. The implementation will have a server program and a client program running on two machines. They communicate with each other using socket communication and TCP and UDP protocols. This is again a group assignment where you will be working in teams of 3 students.

Requirements

Step 1: Using your PA1 (TCP socket implementation) as the foundation, convert your source codes into a UDP-socket version. Make sure UDP works as well as the TCP version.

Step 2: Collect (or create) 10 joke meme images from the Internet and save them on the server machine. Revise your source code to download these 10 memes on the client machine in both TCP and UDP versions.

Step 3: Implement a random-number generator for values between 1 and 10 to retrieve these 10 memes in random order. This step is critical to ensure the server's file system cache performs in average, thus the average access time at the server can be measured.

Step 4: Place the measurement probes (e.g. "*gettimeofday*" command or similar ones) in proper places to report the following time components: (1) The total round-trip time (in msec) to retrieve each meme remotely in the client process; (2) The TCP setup time including IP address resolution in the client process; and (3) The meme access time locally in the server process.

Step 5: Once the measurement is stable, AUTOMATE the measurement 10 times randomly in one run (because user input slows down the execution). After 10 measurements, both server process and client process shall calculate the statistics as (*min*, *mean*, *max*, *stddev*), i.e., like PING statistics. You must visually confirm all 10 meme images are viewable on the client machines. If there are errors for any meme image, take a note of it.

Experiments

Summarize your statistics/error findings in your report on all three experiments as the following:

Experiment#1 (Baseline Performance): The statistics shall be measured via the "localhost" on the same machine on both TCP and UDP versions. Note there is no connection setup for UDP, the second time-component is thus replaced with DNS-lookup-only for UDP.

Experiment#2 (WLAN Performance): Extend the TCP and UDP measurements for actual communication via the wireless LAN (WLAN) between two laptop machines in Lab 114. Compare and analyze these statistics between this experiment and Experiment#1.

Experiment#3 (MAN Performance): Extend the TCP and UDP measurements by placing the server process at a CISE Sun machine (storm/rain/thunder) remotely, and the client process at your local apartment. Compare and analyze the statistics between this experiment and Experiment#2.

Some useful Unix commands

- To check your running processes: `ps -u <your-username>`
- To kill a process: `kill -9 pid`
- To kill all Java processes: `killall java`
- To check processes on remote hosts: `ssh <host-name> ps -u <your-username>`
- To clean Java: `ssh <host-name> killall java`

Port number

You have to be careful when using a port number, i.e., some port numbers are already reserved for special purposes, e.g., 20: FTP, 23: Telnet, 80: HTTP, etc. Our suggestion is to use the last 4-digit of your UFID as the port number to avoid potential conflicts with other students. Keep in mind that the port number must be less than 216 (=65,536).

Getting the most points

1. The server source code needs to be tested on CISE Sun (Unix) machines. Please state your testing machines and results in your report.
2. Please try to complete the source codes step by step. After finishing Step 5, your programs will be able to report the statistics accordingly.
3. Try to finish the simplest operations first (e.g. Experiment#1).
4. Have backups whenever your program can do more. You can avoid the situation where your program used to work but not anymore (e.g. after trying to implement exception handling).

Reminder

1. For your programming conveniences, your server program does not need to be implemented with thread functionalities (multi-threading) in this assignment. In other words, your server program doesn't need to handle concurrent requests by several clients, i.e., we will test your server program by one client ONLY.
2. Using blocks of code from other people or any other resources is strictly prohibited and is regarded as a violating of UF Honor Code! Please refer to the UF honor code if you are unfamiliar with it (http://itl.chem.ufl.edu/honor.html).

Report

Submitted report file should be named “report.pdf” and include the following:

- Personal information of each team member: Full name, UF ID, and Email
- How to compile and run your code under Sun (Unix) environment.
- Description of your code structure.
- Show some execution results. Include screenshots of the results you get from your code wherever possible.
- Statistics/Error report on ALL THREE experiments. Include screenshots of the statistics or errors obtained from the code.
- Lesson learned.
- Any additional comments.

Submission Guidelines

1. The name of a source codes for the TCP server program should be named “TCPServer.java” and the TCP client program “TCPClient.java”. UDP version shall be “UDPServer.java” and “UDPClient.java”. Measurement probes/codes must be clearly marked in source codes with comments.
2. Only submit your Java source code files and report, do not include .obj/.class or executable files in your submission.
3. Include 10 joke meme images.
4. Include the report (with results and discussions) in PDF format.
5. Zip all your files into a packet: PA2_GroupNumber.zip

Grading Criteria

Correct Implementation / Outputs	50%
Readability / Comments / Code structure	15%
Graceful Termination / Exception handling	10%
Report	10%