BIRLA INSTITUTE OF TECHNOLOGY & SCIENCES, PILANI

II Semester 2010-2011 IS C462 Network Programming

Comprehensive Exam Part B (Open Book)

Date: 5th May 2011 Max. Marks: 15

Note:
Write all parts of a question together
Use diagrams to express your models.

HTTP is the most used protocol in the internet. One of the most used applications is the web server. The performance measure of web server known as stress testing is about measuring to what degree the web server can handle simultaneous client requests. There is a need to develop a tool for testing a web server by simulating simultaneous web connections. The most basic operation of the tool is to generate a fixed number of HTTP GET requests and to measure how many replies (responses) came back from the server and at what rate the responses arrived. Q1-Q3 are dependent on this description.

Q1) If the tool is to be used to generate a large number of requests on a single machine, there are several bottle necks. Assume that for every request a new connection is made and the web server accepts any number of connections per client.

a. One of them is port space. Find out with respect to port availability, how many HTTP connections can be opened per second given that TIME_WAIT value is 4 minutes. How does a system with TIME_WAIT value of 1 minute affect this? [1]

b. Another bottleneck is number of file descriptors. If the system allows 1024 descriptors per process, and the time out value per request in the tool is 5 seconds, then how many HTTP connections per second can be opened? If TIME_WAIT value is 4 minutes, then how does it affect the number of file descriptors available per second? [1]

Q2) Assume that web server accepts any number of connections per client. A sample input and output of the tool is shown below. Identify the concurrency model. Write a near-program (marks will depend on netprog related details) for the tool that tests a given web server by making `num-conn` connections with `rate` no. of connections per second and collects enough details so as to print the output as shown below. [3]

```
Input:
http_perf_tool  --server csis --port 80 --uri /test.html --rate 150 --
num-conn 27000 --timeout 5

Output:
Total: connections 27000 requests 26701 replies 26701 test-duration
179.996 s

Connection rate: 150.0 conn/s (6.7 ms/conn)
Connection time [ms]: min 1.1 avg 5.0 max 315.0
Connection time [ms]: connect() 0.3

Request rate: 148.3 req/s (6.7 ms/req)
Request size [B]: 72.0

Reply rate [replies/s]: 148.3
Reply time [ms]: response 4.6
Reply status: 1xx=0 2xx=26701 3xx=0 4xx=0 5xx=0
```

Q3) It is observed that the tool is giving different outputs on different systems due to OS-dependent scheduling and context-switching operations. So it is decided that it will be a single-threaded single-process application. Define a model to achieve concurrency in these constraints with the help of a diagram. [2]

Q4) Chat servers generally use I/O multiplexing based concurrency due to uncertainty in timings of client's responses. But to have stability and scalability in the system, it is planned that the server will use combination of forking with select() as the model for concurrency. No. of clients per process are defined to be CPP. Assume a simple chat protocol that client sends two messages join and chat. In join message, client sends its nickname and in chat it sends the nickname of the person to whom it needs to be delivered and the contents. Define a model for the server with the help of a diagram. Write a near-program for the server that achieves concurrency for this protocol. [3.5]

For the following applications specify the methodologies of programming. Methodology should clearly outline the techniques, data structures, and communication methods etc relevant for the application. [3*1.5]

Q5) An application called '*DnloD*' is used to download large files utilizing the bandwidth to the full extent and in minimum possible time.

Q6) An application called '*connectMe*' is used to connect oneself with a specific group of people of one's interest in the internet. The communication across the group is handled without knowing the list of people in the group and using minimal bandwidth.

Q7) An application called 'tempLi' lists the temperatures of all the important cities in the world. Application knows the server name where these temperatures are stored, service name and the methods and parameters.

End of Part B