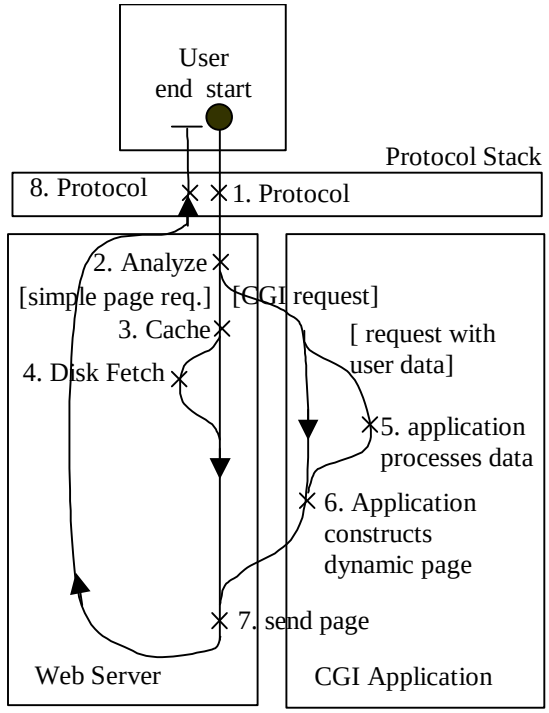
Carleton University
Dept of Systems and Computer Engineering
94.511 Design of High-Performance Software
Murray Woodside, January 2003

Assignment 1, <u>Performance of Sequential Software</u> (due Jan 23)

1. **Workload analysis** of a sequential distributed program (a basic web server), and bottleneck analysis of its performance. The main scenario is described by the following Use Case Map.

The User component runs on a PC and represents a population of users. The Protocol Stack, Web Server and Application run on the Web Server Processor WSP. Static pages are stored on a disk WSDisk. The Application invokes a database server which runs on a Database Processor DBP, which uses its own disk DBDisk.



The operations identified as responsibilities are, with their host demands in msec:
1 and 8: Protocol operations (0.2 ms per packet; a request has 1 packet, a static page has 16 packets, an embedded graphics object takes 2 packets)
2. Analyze the HTTP request to determine its type (1 ms per request)
[first type: request a static page, probability PS]
3. Retrieve page from cache if present (1 ms)
4. Retrieve page from disk if not in cache (probability PCM of cache miss, demand 1 ms)
[second type: CGI request with user data, prob. 0.4(1.0 – PS)]
5. Process the user data in the application, store it in the data base, determine next page to be sent.
[continue, or third type: CGI request without data, probability 0.6(1.0 – PS)]
6. Construct dynamic page
7. Send page, possibly with embedded objects

Operations 4,5,6,7 contain **embedded operations**:
4. Retrieve: 1 ms host demand, and one static web page page retrieve from disk WSDisk (operation 9)
5. Process user data: 15 ms host demand, 1.6 database update operations (11) to store user data
6. Construct dynamic page: 5 ms host demand, and 4.3 database read operations (12)
7. Send page: sends the HTML page (operation 13) and an average of K embedded graphic objects (operation 14).

The embedded operations have demands:
9. Static web page retrieve: average of 3 disk block read operations on WSP/WSDisk (10)
10. Disk block read or write: 0.1 ms host demand, one disk read on attached disk device
11. Database update: 30 ms host (on DBP), 2 disk reads and 4 disk write operations on DBP/DBDisk (10, on DBP/DBDisk)
12. Database read: 20 ms host (on DBP), 4 disk read operations ((10) on WSP/WSDisk)
13. Send static page: 0.5 ms host (WSP) for the HTML page

14. Retrieve and send embedded object: 1 ms host, 0.1 average operations on WSDisk (10)

Disk service times are 5 msec in both cases, and both are single disks; the processors are single processors. Remember that we assume there is no limit to the number of users that can be active in the software components at one time, but the disks and processors are single servers. The scenario is executed sequentially for each request. We will ignore the internet delay and the internal communications delays between WSP and DBP.

(a) **Find** the total average demands per user request for the devices WSP, WSDisk, DBP, DBDisk, for PS = probability of a static page request = 0.9, PCM = prob of cache miss = 0.3, and for K = 7 embedded objects per page, on average.

(b) **Bottleneck analysis**: identify the saturation throughput and the bottleneck device. Name the operations (in the range 1 – 14) which affect the saturation throughput. Do the values of K or PS affect it?

(c) **Closed system** analysis: suppose there are N users with a think time of Z seconds, nominally 3 sec. Determine the path bound and sketch the throughput bounds and the response time bounds.

(d) **What would you conclude** is a reasonable capacity for the server, based on your own expectations for response time?

(e) **What would you recommend** for a hardware expansion, to give increased capacity? What configuration parameters might be changed, for the same purpose? What software modules might be modified?

(f) **Discuss** (mandatory)

2. C**hange the model** to separate the users into **two classes**, those making static page requests (class 1) and those making CGI requests (class 2). There are two separate scenarios,

Class 1: Including operations 1,2,3,4,7,8

Class 2: Including operations 1,2,5,6,7,8, with probability 0.4 of user data, 0.6 not.

(a) find the demands

(b) sketch the bounds in the throughput (f1, f2) plane and identify the devices which have the possibility of being bottlenecks

(c) discuss

3. Queueing model with QNAP2.

Single class:

(a) define a model **on paper** with a "choose" node and the demands determined above.

(b) create a **QNAP model** and solve it for a stream of open requests, for some throughput values below and approaching the saturation value you found. Choose a capacity value that gives a satisfactory average delay to complete the request.

(c) Now create a **closed model** with N users and think time Z = 2.5 sec. Plot throughput values against N for a range of N around the breakpoint population.

(d) **Discuss**.

Two classes: as above

(e) define a model **on paper**, similarly,

(f) create a **closed QNAP model** and solve it for one value of N1, N2. Plot your solution on your sketch of the bounds. If your solution is not very close to the bounds, solve again for a large enough population to give a point very near them.

(g) Identify the **bottleneck** device(s) here, and describe how you might recommend the capacity of the site be increased.

(h) **Discuss** your experience with queueing models.