Carleton University, Dept. of Systems and Computer Engineering, Jan 2003

94.511 Design of High-Performance Software... C. M. Woodside

Assignment 2: Layered Concurrent Systems..... due date Feb 12 2003

1. *(Familiarity with the LQNS solver, and performance metrics and issues)*

The lqns model file printer-b.lqn gives a layered queueing model for the printer system in the slides and notes, with k = 3, single-threaded tasks and one printer. Note that Fig CC in the notes shows CPU demands in microsec, except for the printer and disk, which are (incorrectly) shown in millisec. All communications is synchronous, so the user has to wait for the print job to complete. Define the response time as the time from the user making a request, to the completion of printing.

(a) Suppose there are N users, with a think time of 500 sec. Also, in the file the disk access time of 14 ms from the notes has been replaced by 5 msec. Solve for a few values of N, increasing until the User throughput levels off.

---Plot the mean value of response time and User throughput against N.
---From task and processor utilizations, what is the bottleneck?

(b) To reduce the waiting, introduce second phases. Make the job handoff from the PrintService entry to the Print entry a second phase call, to allow the User to go on about his business without waiting for the job to complete. Make the work of Print, the handoff from Print to the embedded controller entries, and the work of the embedded controller, all second phase. Then PrintService can begin working on the next User input, as soon as Print has accepted the job, and Print can fetch the next page as soon as the embedded controller has taken its page.

---Analyse how to calculate the response time (until the print job is completed) from the task phase times. It may help to draw a timeline diagram of the operations in one print job, for the different tasks, showing the order of phases within the critical path until job completion.

---You expect the throughput to be improved when the system is lightly loaded, but not when it is heavily loaded... does your model show this?

(c) To make the system more scalable (to a larger number of users), consider introducing more printers into the system of (b).... (Np printers), and additional PrintController task threads (Nc threads). Try 5 of each.... how much additional throughput is obtained? Plot the mean response time against N again.

(d) Finally add threads to the PrintManager task. How many does it take to get the maximum benefit?

(e) Discuss (c) and (d). From these results, argue as to which entry would best repay the effort of reducing its CPU demands (by tightening the code).

2. *(Building a layered model from scenario data)* We will use the web server scenario from question 1 of assignment 1, with a few changes.

Let there be NU Users with a "think time" of ZU minutes between requests for web pages, and note that each User has its own PC. There are tasks “ProtocolStack”, “WebServer” and “CGIApplication”, running on WSP, and a task DBServer running on DBP.
(a) Build an LQN model with

... a User task for each user and browser together,
... all tasks single threaded
... as in question 1(a) of assignment 1, PS = probability of a static page request = 0.9, PCM = prob of cache miss = 0.3, and K = 7 embedded objects per page, on average.

(a1) Give the model
(a2) Find a range of NU up into saturation (leveling off of throughput).
(a3) Compare the throughput to your queueing solution in Assignment 1, explain the difference. What resources are saturated in the LQN?
(a4) How sensitive is the result to the cache hit rate PCM?

Remember: in the LQN results, the time delay to execute an entry is called its "Service Time"; the "Task Utilization" is the fraction of time the task is busy (or the mean number of busy threads, if more than one thread), and the task or entry throughput is the mean rate at with it receives requests (for User, it is the mean rate of cycling).

(b) ProtocolStack might be modeled as forwarding the message, since it does not block once it has processed an incoming message. Model it so that all its work is done on receiving a request, rather than both receiving and sending the reply. Then, also introduce multi threading in the WebServer and the DBServer tasks. The in-memory cache of web pages is assumed to be shared, with the same hit rate.

(b1) Give the model.
(b2) Use the model to choose threading levels that give close to maximum obtainable throughput, without wasting resources on unused threads (what resources would be wasted?).
(b3) Again compare results to the queueing network and discuss the sources of differences.

(c) Divide the Database host demand into first and second phases (30% in phase 2). Does it help? Why?

(d) Suppose the Database can split each query into two parallel parts, each with 60% of the host demand. To give parallel execution, make DBP a dual processor, and put in ND disks. When a disk access happens, each of the parallel branches accesses one of them with equal probability. In the dual processor we assume the in-memory cache of web pages is shared between the parallel paths and the database threads.

(d1) Give the model
(d2) Again as in (b2) find out if more threads are useful. Is there an increase in the achievable throughput?
(d3) Discuss.

(e) Don’t forget to include any overall comments you can derive about what can be learned about this system from layered modeling.