Computer System Modelling And Analysis

Laboratory #1
September 19/21 & 26/28, 2000

Purpose

The goal in this lab exercise is to write a Csim program to simulate a simple closed queueing network with three service centres. The simulation model should allow you to investigate the effects of various scheduling disciplines and various distributional forms for service times and think times on the performance of the system being modelled.

Preparation

Write a CSim program that allows you to evaluate the performance of the queueing network shown on page 2. The program should permit all model parameters (shown in the table under the figure) to be easily changed.

The model should have three service centres (each one represented by a *facility* in Csim). Naming the centres as *cpu*, *disk* 1, and *disk* 2, let the parameters, p_{th}, p_{d1}, and p_{d2} be respectively the probabilities that a *cpu* visit is followed by thinking, or visiting *disk* 1 or visiting *disk* 2. (Closed queueing networks with two or one service centres can then be represented by setting one or both of p_{d1} and p_{d2} to zero.)

At each of the three service centres, it should be convenient to choose the distribution from which service times are drawn to be one of Exponential, Erlang or Hyperexponential. For the latter two, not only the mean but also the variance of the distribution must be specified. Similarly, it should be possible to specify any of those three distributions as the form of the think time distribution.

For the cpu queue (only), it should be possible to change the scheduling discipline from first-come-first-served (fcfs) to one of Round-robin, Last-come-first-served, or Priority Preemptive Resume. In the case of Round-robin, it is necessary to specify a quantum length, and in the case of Preemptive-resume, it is necessary to set the priority of each job as a function of its characteristics. For example, if you wished to give priority to short jobs over long ones then you would set each job's priority to a value inversely proportional to its service time.

The primary performance measure of interest is the average response time to customers (i.e., the time from the end of one think time to the start of the next think time). Calculate and tabulate the response times, and print out the average response time at the end of the simulation. Your program should also accumulate statistics on observed think times and observed service times at various individual centres (so that you can make sure that the intended means and variances are realized). The program should also keep statistics on the distribution of the number of customers in the system (i.e., not thinking) over time.

All these can be done straightforwardly with CSim histograms and tables, but for convenience in localizing the important quantities, you should also print out a few consecutive summary lines in which you provide the means and variances of think times, service times at various centres, number in system, and response times.

model parameters	symbol
number of customers active	N
cpu scheduling discipline	*sched*
distributional form, mean and variance of *think* times	$dist_{th}$, m_{th}, var_{th}
distributional form, mean and variance of *cpu* service times	$dist_{cpu}$, m_{cpu}, var_{cpu}
distributional form, mean and variance of *disk* 1 service times	$dist_{d1}$, m_{d1}, var_{d1}
distributional form, mean and variance of *disk* 2 service times	$dist_{d2}$, m_{d2}, var_{d2}
probability of thinking after a *cpu* visit	p_{th}
probability of moving to *disk* 1 after a *cpu* visit	p_{d1}
probability of moving to *disk* 2 after a *cpu* visit	p_{d2}
number of completions to simulate	C

sched = (FCFS, LCFS, RR, PRI)
$dist_{xx}$ = (Exp, ERk, HExp, Detr)

Test Cases

Here are some sets of parameter settings to try as test cases. For the test, use *FCFS* service at each centre, and assume that the service time distribution at each centre as well as the think time distribution are *exponential*. (You should also try your program out with other service disciplines and other service time distribution forms so that you know how to make such changes conveniently.)

parameter	N	m_{th}	m_{cpu}	m_{di}	m_{d2}	p_{th}	p_{d1}	p_{d2}	C
test model 1	3	1.0	1.0	--	--	1.0	0.0	0.0	10000
test model 2	4	5.0	5.0	5.0	--	.4	.6	0.0	10000
test model 3	5	10.0	2.0	1.0	3.0	.2	.4	.4	10000