

Homework #3 (100 Points)

Due 03:00pm on Tuesday 02/23/2016 through gsubmit.

Feel free to make assumptions, if you feel that such assumptions are justified or necessary. Please state your assumptions clearly. Unreasonable assumptions lead to unreasonable grades!

1. The 96th percentile confidence interval for the response time of a database transaction system is given by the interval $[3-0.4, 3+0.4]$ in seconds. This confidence interval was computed using 40 samples. Answer the following questions (stating any assumptions you make):
 - a. Based on the above information, what do you think the standard deviation of the sample used to compute this confidence interval was?
 - b. Compute a 99th percentile confidence interval for this system.
 - c. How many more samples do you need to make the 96th percentile confidence interval be $[3-0.1, 3+0.1]$?
2. Write a simulator of an M/M/1 queue. Your simulator should follow the general structure described in class and in the lecture notes. Namely, the simulator will consist of two parts: (1) a controller, which is the same exact logic for all simulators you will be writing for the course, and (2) a simulated system, which will be different for each system you simulate.

As discussed in the notes, the two data structures needed for the controller are the current simulated time (initialized to 0) and a "calendar" or "schedule" of events which is a linked list of "future" events, each of which has a timestamp and a pointer to the function that needs to be executed at that time. The simulation controller's logic is very simple:

- a. Set $t=0$; and call a function to initialize the simulated world and the calendar
- b. While t is less than the desired time for the simulation
 1. Get the next event from the calendar
 2. Adjust the current time to be the time of that event, and
 3. Call the function that needs to be executed to reflect the occurrence of the event
- c. Call a function that prints the results of the simulation, etc.

The simulated "world" consists of data structures that represent the state of the "world", which for M/M/1 is nothing other than the state of a single queue (a record of pending requests for that queue, and any information you want to keep about each request).

In our M/M/1 "world", there are two possible events a "birth event" and a "death event". For each of these events you will need to define a function that captures what needs to be done when this event happens. Notice that in each of these events, you may also need to have code that would record important information, e.g., when was service started, completed, etc. In addition to figuring out the logic of these two functions, you will need to figure out how to initialize the simulated system.

Finally, in addition to the specifics of the M/M/1 system above, it is often desirable to measure a particular phenomenon (or value) repeatedly (say every millisecond) and report what is observed (and potentially calculate some statistics, e.g., mean, standard deviation, etc.) As discussed in the notes, a handy way to do this is to define a "special" event, call it the "monitoring event", which happens periodically (e.g., report some value every millisecond), or even better at random exponentially distributed intervals following the PASTA principle, and which simply reports the state of the simulated world (perhaps by writing some values to a file).

To test your simulator, use it to verify the following settings for an M/M/1 system:

- a. $\Lambda = 60$ and $T_s = 0.015$ and simulation time = 100

For each of the above, calculate T_w , T_q , and w , and q and compare them to whatever the M/M/1 analysis is able to tell you. Make sure that you measure the system after it had a chance to "warm up" (i.e., at steady state). One way to do this is to run the simulation for twice as long and collect measurement only for the second half of your simulation.

Now that you have some confidence in your simulator, try the following settings (and compare the results to analytical expectations).

- b. $\Lambda = 50$ and $T_s = 0.015$ and simulation time = 100
- c. $\Lambda = 65$ and $T_s = 0.015$ and simulation time = 100
- d. $\Lambda = 65$ and $T_s = 0.020$ and simulation time = 100

Your solution to this problem should consist of (1) the *source code* for the simulator with proper instructions as to how we would be able to run your code, (2) a *log* showing the results of executing your code, and (3) a short *write-up* answering parts a-d above.

- 3. The inter-arrival time of packets coming into a gateway is exponentially distributed, with an average incoming rate of 1,200,000 packets/min. Its outgoing transmission line has a rate of 24,000,000 bytes/sec. Packet lengths are distributed uniformly between 100 bytes and 1,500 bytes.
 - a. What kind of queuing model do you think is an appropriate model for this gateway?
 - M/D/1
 - M/M/1
 - M/G/1
 - None of the above
 - b. What is the mean time it takes a packet to be sent over the transmission line?
 - c. What is the standard deviation for the time it takes a packet to be transmitted?
 - d. What is the mean number of packets in the gateway ?
 - e. What is the mean number of packets in the wait queue?
 - f. What is the average latency?
 - g. What is the time that a packet spends in the wait queue?

h. What is the chance that a packet coming into the router will not have to wait in queue?

4. The execution of processes submitted to the system proceeds as follows:

- i. The process uses the CPU and then with a probability 0.1 proceeds to step (ii), with probability 0.4 proceeds to step (iii), and with probability 0.5 proceeds to step (iv).
- ii. The process performs a disk I/O and then with a probability 0.5 proceeds to step (i), and with probability 0.5 proceeds to step (iii).
- iii. The process performs a network I/O and then proceeds to step (i).
- iv. The process is done!

The following information is known about the above system.

- Process arrivals are Poisson with a rate of 40 processes per second
- The CPU is a dual-core design; the ready queue is serviced by any one of the 2 CPU cores.
- The CPU service time is exponential with mean of 20 msec
- Disk I/O service time is exponential with mean of 100 msec
- Network service time is exponential with mean of 25 msec
- There is plenty of memory, so all buffers are of infinite size

Answer the following questions:

- a. Draw a queuing network representation of the above system.
 - b. Find the average turnaround time (i.e. total response time) for processes submitted to the above system.
 - c. What service (CPU, Disk, or network) is the "bottleneck" in the above system? Why?
 - d. What process arrival rate will render this system unstable, i.e., will make one of the queues grow infinitely long?
-