Department of Computer Science BOSTON UNIVERSITY

CS-350 - Fundamentals of Computing Systems

Midterm Exam #1
Fall 2022

Name: _____

BU Username: _____ BU ID: _____

NOTE: *Please use only the provided space and the included extra pages to answer the questions. If you do use the extra pages, make sure you reference them properly in your solutions.*

Problem #1:	/24
Problem #2:	/20
Problem #3:	/22
Problem #4:	/22
Problem #5:	/22
Total Score:	/110

Remarks:

- This is a closed-book/notes exam.
- Basic calculators are allowed.
- You have 80 minutes to complete your exam.
- There are 110 total points.
- If your score is more than 100, it is capped at 100.
- Show your work for full marks.
- Problems and sub-problems weighted as shown.
- **Explain all your assumptions clearly.**

Problem 1

Label each of the statements below with either **True (T)** or **False (F)**:

	Statement	T/F
a.	If the inter-arrival beween events is exponentially distributed, then the number of such events per unit of time is a uniformly distributed random variable.	**FALSE**
b.	It is always possible to solve a system using Jackson's Theorem if the system includes a mix of M/M/1 and M/M/1/K queues.	**FALSE**
c.	Two systems can have identical reliability and yet different availability.	**TRUE**
d.	The probability that all the servers are busy in an N*M/M/1 system is identical to that of an M/M/N system under the same conditions (same N, λ, and T_s).	**FALSE**
e.	In a generic system, the control plane drives the behavior of the data plane, while the data plane reports status information to the control plane.	**TRUE**
f.	A process that is considered *blocked* from the point of view of resource A might be considered *ready* from the point of view of resource B.	**TRUE**
g.	A system is able to reach steady-state so long as at least one of its resources has utilization greater than or equal to 100%.	**FALSE**
h.	In an M/M/1/K system, the rate of rejected requests is always lower than the rate of accepted requests.	**FALSE**
i.	Increasing the MPL of a system always leads to an increase in throughput.	**FALSE**
j.	Consider a program with single-CPU runtime T, and with $f = 40\%$ of T that is infinitely parallelizable. A speedup of 1.5× or higher is possible.	**TRUE**
k.	An M/M/1 system has infinite capacity because the size of its queue is infinite.	**FALSE**
l.	For a system at steady state where no request completion (*death*) goes unaccounted, the average population size is directly proportional to the request arrival rate and to the average request response time.	**TRUE**
m.	Consider two identical systems A and B subject to different input rates λ_A and λ_B, respectively. A has greater capacity than B if $\lambda_A > \lambda_B$.	**FALSE**
n.	An M/G/1 system with mean service time T_s and standard deviation σ_{T_s} behaves exactly as an M/M/1 system with mean service time T_s if $T_s = \sigma_{T_s}$.	**TRUE**
o.	The queue buildup w (i.e. how many requests on average are queued waiting for service) increases linearly as the arrival rate λ increases.	**FALSE**
p.	One can analyze a system with two request classes (X and Y) with mean service times $T_{s,X} \neq T_{s,Y}$ sharing the same queue using an M/M/1 model.	**FALSE**
q.	Little's Law does not apply if the average input rate of requests does not match the average output rate of completed requests leaving the system.	**TRUE**
r.	A request arriving at an M/M/1 system will observe a response time equal to its service time with probability ρ, where ρ is the utilization of the system.	**FALSE**

Note: There are 18 questions. A correct answer will get you 2 points; an incorrect answer -1 points; a blank answer 0 points. The final score is capped at 24.

Problem 2

A redundant array of disks—also known as a RAID system—is comprised of three independent disks. Each contains a full replica of the entire content that can be accessed by the users. Thus, any request can be served by any of the disks. You have benchmarked that 2,000 requests per second reach the disk, on average. When a request arrives at the RAID system, a load balancer instantaneously directs the request to one of the disks. With probability $p_1 = 0.2$, the request is directed to the first disk; with probability $p_2 = 0.5$ to the second. Otherwise, the request is sent to the third disk. The first disk can serve a request in about 2 milliseconds; the second in about 0.7 milliseconds; and the third in about 1.5 milliseconds.

(a) [6 points] Which disk constitutes the bottleneck of the RAID system?

Solution:

Consider the arrival of requests as Poisson-distributed with parameter (and rate) $\lambda = 2000$ req/sec $= 2$req/millisec.

We model each disk as an M/M/1 system. We can then compute the arrival rate of requests at each disk as follows:

$\lambda_1 = \lambda p_1 = 2 \cdot 0.2 = 0.4$ req/millisec.

$\lambda_2 = \lambda p_2 = 2 \cdot 0.5 = 1$ req/millisec.

$\lambda_3 = \lambda(1 - p_1 - p_2) = 2 \cdot 0.3 = 0.6$ req/millisec.

From here, we can compute the various utilizations:

$\rho_1 = \lambda_1 T_{s,1} = 0.4 \cdot 2 = 0.8$

$\rho_2 = \lambda_2 T_{s,2} = 1 \cdot 0.7 = 0.7$

$\rho_3 = \lambda_3 T_{s,3} = 0.6 \cdot 1.5 = 0.9$

(b) **[7 points]** What is total number of requests that are either being worked on or waiting for service within the entire RAID system?

Solution:

For this part, we can calculate the total average system population q_{tot} by summing up the average population at each of the individual disks.

$q_1 = \frac{\rho_1}{1-\rho_1} = 0.8/(1-0.8) = 4$

$q_2 = \frac{\rho_2}{1-\rho_2} = 0.7/(1-0.7) = 2.33$

$q_3 = \frac{\rho_3}{1-\rho_3} = 0.9/(1-0.9) = 9$

Therefore $q_{tot} = q_1 + q_2 + q_3 = 4 + 2.33 + 9 = 15.33$

(c) **[7 points]** What is the average response time for a generic request arriving at the RAID system?

Solution:

This can be solved in two ways. The first is to analyze each M/M/1 disk in isolation and then weighing the mean response time by the probability of being routed through that disk. This can be done as follows:

$T_{q,1} = q_1/\lambda_1 = 4/0.4 = 10$

$T_{q,2} = q_2/\lambda_2 = 2.33/1 = 2.33$

$T_{q,3} = q_3/\lambda_3 = 9/0.6 = 15$

$T_q = p_1 T_{q,1} + p_2 T_{q,2} + p_3 T_{q,3} = 0.2 \cdot 10 + 0.5 \cdot 2.33 + 0.3 \cdot 15 = 7.665$

Alternatively, we can apply Little's Law directly on the entire system. In this case:

$T_q = q_{tot}/\lambda = 15.33/2 = 7.665$

Problem 3

Your job is to fine-tune a single-processor server used to handle FTP file requests from a pool of users that submit, on average, 50 requests per minute. The goal of the tuning is to achieve an average system latency, as perceived by the users, that is below 0.09 minutes. To try and achieve this goal, you are evaluating 4 different alternatives. Reason on the ability of each alternative to achieve the goal by answering the following.

(a) [5 points] Alternative 1: A server with exponentially distributed service time and capable of processing 60 requests per minute on average. Is this system capable of achieving the goal?

Solution:

This system can be modeled as an M/M/1 system.

Its utilization is $\rho = 50/60 = 0.83$.

Therefore $q = \rho/(1-\rho) = 5$ and

$T_q = q/\lambda = 5/50 = 0.1$ minutes.

Thus, this alternative is NOT viable.

(b) [5 points] Alternative 2: A server with constant service time capable of processing exactly 57 requests per minute. Is this system capable of achieving the goal?

Solution:

This system can be modeled as an M/D/1 system.

Its utilization is $\rho = 50/57 = 0.877$.

Therefore $q = \frac{\rho^2}{2(1-\rho)} + \rho = 4.01$ and

$T_q = q/\lambda = 4.01/50 = 0.08$ minutes.

Thus, this alternative is viable.

(c) [6 points] Alternative 3: A server with service times that follow a bimodal distribution with mean 0.016 minutes and standard deviation 0.032. Is this system capable of achieving the goal?

Solution:

This system can be modeled as an M/G/1 system.

Its utilization is $\rho = 50 \cdot 0.016 = 0.8$.

Therefore we can compute $A = \frac{1}{2}(1 + (\sigma_{T_s}/T_s)^2) = 0.5 \cdot (1 + 2^2) = 2.5$.

Then, we can compute $q = A\frac{\rho^2}{1-\rho} + \rho = 8.8$ and

$T_q = q/\lambda = 8.8/50 = 0.176$ minutes.

Thus, this alternative is NOT viable.

(d) [6 points] Alternative 4: The same exact option as in Alternative 1, but where no additional requests are accepted if, at the time of their arrival, 4 other requests are already in the system (waiting for service or being worked on). Is this system capable of achieving the goal for those requests that are admitted and receive service?

Solution:

This system can be modeled as an M/M/1/K system.

Its utilization is $\rho = 50/60 = 0.83$.

We can then compute the probability of rejection as $P(S_{K=4}) = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} = 0.134$.

If follows that the rate of accepted requests is $\lambda_{acc} = (1-P(S_{K=4}))\lambda = (1-0.134)\cdot 50 = 43.3$.

We can compute the average system population $q = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} = 1.64$.

Finally, the response time average of accepted requests is $T_q = q/\lambda_{acc} = 0.033$.

Thus, this alternative is viable.

Problem 4

You have put together a 3D printing farm. To do that, you have purchased three printers and arranged them to accept requests from a pool of beta-test users. Requests from the users are sent to a single, central queue and dispatched to the first idle printer in FIFO order. If multiple idle printers are available, one of them is randomly selected. When a printing job is done, the printing surface is immediately cleaned up and the printer can pick up the next job in the queue (if any). You have installed a camera and monitored ONE of the printers. This has revealed that the printer is busy for 21 hours per day on average. You have also evaluated that the users are submitting 1 print job every 2 hours, on average.

(a) [7 points] How long is on average each printing job once it starts service?

Solution:

We can model the system as an M/M/3 queue.

The single-printer utilization is given as $\rho = 21/24 = 0.875$.

Since users send one job every two hours, the arrival rate $\lambda = 0.5$ job/hour.

We know that in an M/M/3, $\rho = \frac{\lambda T_s}{3}$, thus $T_s = 3\rho/0.5 = 5.25$ hours.

(b) **[7 points]** For how many hours in a 24-hours time window are ALL the printers busy, on average?

Solution:

We can compute the probability that all of the printers are busy as $C = \frac{1-K}{1-K\rho}$, where K is the Poisson Ratio.

$K = \frac{1+3\rho+(3\rho)^2/2}{1+3\rho+(3\rho)^2/2+(3\rho)^3/6} = \frac{7.07}{10.08} = 0.7$

Thus, $C = \frac{1-0.7}{1-0.875 \cdot 0.7} = 0.77$.

Finally, in a 24-hour window, the average number of hours for which all the servers are busy is $C \cdot 24 = 18.48$ hours.

(c) **[8 points]** What is the average time spent by a generic print job waiting in the queue before starting to be worked on?

Solution:

In order to compute T_w, we first compute the average waiting queue length (excluding any in-service request) $w = C\frac{\rho}{1-\rho} = 5.39$ jobs.

Next, we can compute $T_w = w/\lambda = 5.39/0.5 = 10.78$ hours.

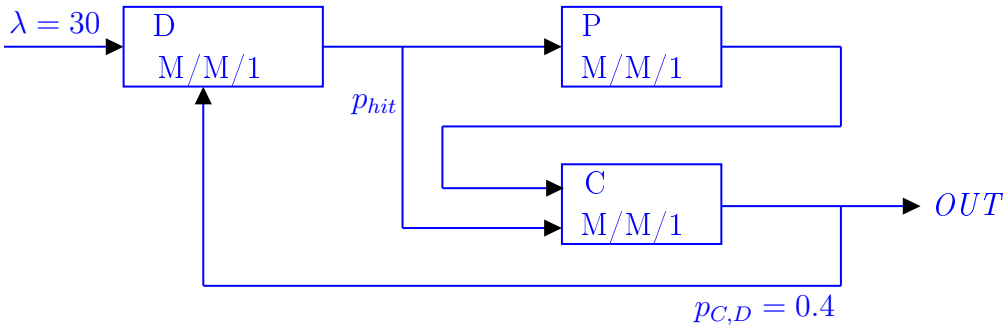
We can also verify that $T_q = T_w + T_s = 10.78 + 5.25 = 16.03$ hours.

Problem 5

A request processing system is comprised of three machines. Requests always arrive at the first machine called the decoder (D). The decoder understands a request's payload and also checks if the response is already available in the system cache. If the response is not in the cache (cache miss), the request is sent from D to the processing engine (P) for full processing. After full processing, the request is sent from P to the cache machine (C) so that the response produced by P can be added to the cache. Conversely, in case of a cache hit at D, the request is directly sent from D to C for fast response retrieval—in this case, the request essentially bypasses P. Cache hits at D occur with probability p_{hit}. Regardless of how the request reached the C machine, after being served at C, the request might request additional processing from the system. This happens with probability 0.4, in which case it will go back to the decoder (D) as if it was a new request. Otherwise, it leaves the system and it is considered completed. On average, the system receives 30 requests per second. The average service time at D is 11 milliseconds; 28 milliseconds at P; and 9 milliseconds at C.

(a) [2 points] Provide a diagram of the system which shows the inter-connections between the D, P, and C machines. Describe the queuing models used for each machine, the known parameters, and the assumptions you will employ to solve the system.

Solution:



Assumptions:

(1) System is at steady state;

(2) All queues infinite and with FIFO discipline;

(3) Arrivals from the outside are Poisson;

(4) All service times are exponentially distributed;

(b) [5 points] Assume that the probability p_{hit} of a cache hit (a.k.a. hit rate) is 30%. What resource constitutes the bottleneck in the system? Motivate your answer.

Solution:

The arrival rate from the outside is $\lambda = 30$ req./sec.

After completing at C, requests can go back to D with probability $p_{C,D} = 0.4$.

We can find $\lambda_D = \lambda + p_{C,D}\lambda_D$. It follows that $\lambda_D = \frac{\lambda}{1-p_{C,D}} = \frac{30}{0.6} = 50$ req./sec.

Since any request going through D also goes through C, we have $\lambda_C = \lambda_D$.

Conversely, if we call the cache hit rate p_{hit}, $\lambda_P = (1 - p_{hit})\lambda_D = 50 \cdot 0.7 = 35$ req/sec.

We can compute all the utilizations.

$\rho_D = \lambda_D \cdot T_{s,D} = 50 \cdot 0.011 = 0.55$;

$\rho_C = \lambda_C \cdot T_{s,C} = 50 \cdot 0.009 = 0.45$;

$\rho_P = \lambda_P \cdot T_{s,P} = 35 \cdot 0.028 = 0.98$;

Thus, P is definitely the bottleneck since it has the highest utilization.

(c) [5 points] With the same hit rate, what is the average response time of the entire system?

Solution:

For this part, we first compute the various q.

$q_D = \rho_D/(1 - \rho_D) = 1.22$ requests;

$q_C = \rho_C/(1 - \rho_C) = 0.82$ requests;

$q_P = \rho_P/(1 - \rho_P) = 49$ requests;

$q_{tot} = q_D + q_C + q_P = 51.04$ requests; and

$T_{q,tot} = q_{tot}/\lambda = 21.04/30 = 1.7$ seconds.

(d) [5 points] With the same hit rate, what is the capacity of the system?

Solution:

The capacity is the value of throughput λ^* at which the bottleneck reaches 100% utilization.

We can setup the following equation: $\rho_P = \lambda_P \cdot T_{s,P} = 1$

We know that $\lambda_P = 0.7\lambda_D = \frac{(1-p_{hit})\lambda^*}{0.6} = \frac{0.7\lambda^*}{0.6}$.

Thus, we can write $\frac{0.7\lambda^*}{0.6}T_{s,P} = 1$ and solve for λ^*.

It follows that $\lambda^* = \frac{0.6}{0.7T_{s,P}} = 30.6$ req./sec.

(e) [5 points] Suppose you were able to introduce a new caching heuristic that increases the hit rate p_{hit} from the previous 30% to 50%. What is the resulting increase in system capacity?

Solution:

The previous capacity was $\lambda^* = 30.6$, which was computed as $\lambda^* = \frac{0.6}{(1-p_{hit}^{old})T_{s,P}} = 30.6$ req./sec.

With the new heuristic, the new capacity is $\lambda'^* = \frac{0.6}{(1-p_{hit}^{new})T_{s,P}} = \frac{0.6}{0.5 \cdot 0.028} = 42.86$ req/sec. req./sec.

This corresponds to an increase in overall system capacity of about 1.40×, a.k.a. about 40% better.

[EXTRA BLANK PAGE (1)]

[EXTRA BLANK PAGE (2)]

Some Probability Density Functions

- Poisson: $f(x) = \frac{\lambda^x}{x!}e^{-\lambda}$
- Exponential: $f(x) = \lambda e^{-\lambda x}$, $F(x) = 1 - e^{-\lambda x}$
- Standard Normal: $f(z) = \frac{1}{\sqrt{2\pi}}e^{-0.5z^2}$

Equations for Some Queuing Systems

- M/G/1 System: $q = \frac{\rho^2 A}{1-\rho} + \rho$, $w = \frac{\rho^2 A}{1-\rho}$, where $A = \frac{1}{2}\left[1 + \left(\frac{\sigma_{T_s}}{T_s}\right)^2\right]$
- M/D/1 System: $q = \frac{\rho^2}{2(1-\rho)} + \rho$, $w = \frac{\rho^2}{2(1-\rho)}$
- M/M/1/K system: $q = \begin{cases} \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} & \text{for } \rho \neq 1 \\ \frac{K}{2} & \text{for } \rho = 1 \end{cases}$,

  $P(\text{“rejection”}) = P(S_K) = \begin{cases} \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} & \text{for } \rho \neq 1 \\ \frac{1}{K+1} & \text{for } \rho = 1 \end{cases}$
- M/M/N System: $q = C\frac{\rho}{1-\rho} + N\rho$, $w = C\frac{\rho}{1-\rho}$,

  where $\rho = \frac{\lambda T_s}{N} = \frac{\lambda}{N\mu}$, $C = \frac{1-K}{1-\rho K}$, and $K = \frac{\sum_{i=0}^{N-1} \frac{(N\rho)^i}{i!}}{\sum_{i=0}^{N} \frac{(N\rho)^i}{i!}}$.

