Problem 1

Your career as an independent game developer is going better than expected! You are almost ready to release your first multi-player, called *Goat Simulator: The Heard*[1]. You decide to implement the game server using a single-processor machine that you found in your garage. You tested the first client and noticed that whenever a user plays the game, an average of 10 requests per second are sent to the server. It also appears that the arrival of requests is Poisson-distributed. Furthermore, you have noticed that if any of the requests is dropped/rejected, then the client generating the request will disconnect — which is frustrating for the wannabe goat users. On the other hand, if the average per-request latency grows beyond 0.5 milliseconds, then all the players will experience lags. You can also assume that the service time for each request is exponentially distributed.

a) At the beginning, you strive to have exactly 0% rejection rate by creating a very large request queue in your server. What should be the capacity of the server to prevent the occurrence of lags when 1500 users are playing the game?

b) Assume that you were able to implement the server such that the average per-request service time is 0.06 milliseconds. How many active users can you support before the probability of a generic request of experiencing queuing (i.e. of not being processed right away) grows beyond 81%?

c) You quickly realize that you are not able to meet the ideal specifications demanded by Part a). Apparently, the best you can do is to achieve an average per-request service time of 0.07 msec. You then decide to limit the size of the queue to introduce some rejection in order to limit the lag for everyone. Note that a request occupies space in the queue while it is being processed. Is it possible to have 1500 active users and to set a cap on the request queue somewhere between 10 and 15 to guarantee the constraint on the server latency (less then 0.5 milliseconds) while rejecting less than 11% of all the requests?

Regardless of what you computed above, assume that you set the queue size to 10. Keep considering the given service time of 0.07 msec, and 1500 active users.

d) What is the probability that a request will be processed right away without experiencing any queuing?

e) What is the probability that when a generic request arrives, it finds exactly 5 or 4 other requests in the queue?

[1] A much acclaimed multi-player spin on the award winning Goat Simulator 3

$T_s = 0.06\,msec = 0.00006$

$\rho \le 81\%$

$K \in \{10 \text{ to } 15\}$

$K = 10$

a). $\lambda = 1500(10) = 15000$

$\therefore T_q \le 0.5\,msec = 0.0005$

$T_q = \frac{q}{\lambda} \quad \therefore \frac{q}{\lambda} \le 0.0005$

≤ 0.005

$q \le 0.0005\lambda = 7.5$

$q = \frac{\rho}{1-\rho} \Rightarrow \rho = \frac{q}{1+q} = \frac{7.5}{1+7.5} = 0.88 \quad \therefore \rho \le 0.88$

$\rho = \frac{\lambda}{\mu} \quad 0.88 = \frac{15000}{\mu} \Rightarrow \mu = 17045 \text{ requests/s}$

b). $\rho = \lambda \cdot T_s \cdot \#\text{ of users}$

$\rho \le 81\%$

$81\% \ge 1/10 \cdot (0.00006) \cdot \#\text{ of users}$

$\therefore \#\text{ of users} \le 13500$

c). $\rho = 1500\lambda \cdot T_s = 1500(10)(0.07)(0.09)$

$= 1.05$

$Pr(S_K) = \frac{(1-\rho)\rho^K}{(1-\rho^{K+1})}$

$Pr(S_{10}) = \frac{(1-1.05)1.05^{10}}{(1-1.05^{11})} = 0.1146 > 0.11$

$Pr(S_{11}) = \frac{(1-1.05)1.05^{11}}{(1-1.05^{12})} = 0.1075 < 0.11$

$\therefore K = 11$

$\lambda = (1-0.1075)(1500)(10) = 13387.5$

$q = \frac{\rho}{(1-\rho)} - \frac{(K+1)\rho^{K+1}}{(1-\rho^{K+1})}$

$= \frac{1.05}{1-1.05} - \frac{(12)1.05^{12}}{(1-1.05^{12})} = 6.078$

$T_q = \frac{q}{\lambda} = \frac{6.078}{13387.5} = 0.0004545$

$< 0.0005 s.$

d). $Pr(S_0) = \frac{(1-\rho)\rho^K}{(1-\rho^{K+1})} = \frac{(1-1.05)1.05^0}{(1-1.05^{11})}$ ($K = 10$)

$= 0.07039$

e). $Pr(S_4) = \frac{(1-1.05)1.05^4}{(1-1.05^{11})} = 0.0856$

$Pr(S_5) = \frac{(1-1.05)1.05^5}{(1-1.05^{11})} = 0.0898.$

$\therefore Pr(S_{5 \text{ or } 4}) = 0.0856 + 0.0898 = 0.1754$

Problem 2

The latest and greatest version of the AliImpress e-commerce infrastructure is structured as a three-tier platform, with three single-CPU servers S1, S2 and S3. S1 handles the arrival of new users that have never seen before—to create a new account. After S1, a request from these new users must visit S3, which is responsible for session creation. After a valid session has been established, a request will move from S3 to S2 for processing. With 0.6 probability, a generic request completes at S2 and leaves the platform. Otherwise, additional processing is required. In this case, a new session needs to be created by visiting S3, and from there the request continues just like any other request visiting S3. Differently from new users' requests, requests from well-known users arrive directly at S2 and from there proceed just like described above — i.e., they might complete and leave rigth away after S2, or go to S3 for an additional session and so on.
At steady-state, the system receives $\lambda_a = 10$ requests per second from new users, and $\lambda_b = 50$ requests per second from well-known users. Moreover, S2 has an average service time of 9 milliseconds; S3 has an average service time of 19 milliseconds. The service time of S1 is unknown, but you were able to measure the average response time at S1, which is 525 milliseconds.

$\bar{T}_{q_{S1}} = 525 = 0.525$ $\bar{T}_{S2} = 0.009$ $\bar{T}_{S3} = 0.019$

a) Provide a diagram of the system and specify the queuing model used for S1, S2, and S3, the assumptions that need to be made and the name of the theorem used to decompose the system.

b) What is the average service time of S1? $\bar{T}_{S_1}$?

c) What is the utilization of S2?

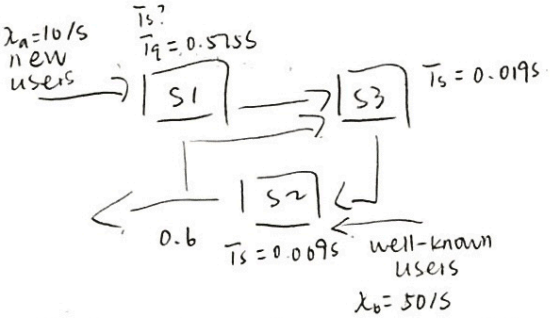
d) What is the average total number of requests in the system—either waiting for service or being served?

e) What is the capacity of the system, assuming that it always holds that $\lambda_b = 5 \cdot \lambda_a$? λ_{total}?

f) What is the response time measured from entry to exit of a generic request, regardless of whether it is from new or existing users? $\bar{T}_N$?

a) assumption:
(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;
(5) Jackson's Theorem

a).



b). $\bar{T}_q = \frac{\bar{T}_s}{1-\lambda\bar{T}_s}$

$\therefore \bar{T}_s = \frac{\bar{T}_q}{1+\lambda\bar{T}_q} = \frac{0.525}{1+10(0.525)} = 0.084s$

c). $\lambda_{S2} = \lambda_b + (1-0.6)\lambda_{S2} + \lambda_a$

$\lambda_{S2} = 50 + 0.4\lambda_{S2} + 10$

$\lambda_{S2} = 100$

$\rho_{S2} = \lambda_{S2} \cdot \bar{T}_{sS2} = 100(0.009) = 0.9$

d). $\rho_{S1} = \lambda_a \cdot \bar{T}_{sS1} = 10(0.084) = 0.84$

$\lambda_{S3} = 0.4\lambda_{S2} + \lambda_a = 0.4(100) + 10 = 50$

$\rho_{S3} = \lambda_{S3} \cdot \bar{T}_{sS3} = 50(0.019) = 0.95$ → bottleneck

$q_{S1} = \frac{\rho_{S1}}{1-\rho_{S1}} = \frac{0.84}{1-0.84}$ $q_{S3} = \frac{\rho_{S3}}{1-\rho_{S3}} = \frac{0.95}{1-0.95}$

$q_{S2} = \frac{\rho_{S2}}{1-\rho_{S2}} = \frac{0.9}{1-0.9}$ $q_{total} = 5.25 + 9 + 19 = 33.25$

3

e). $\rho_{S3} = 1$ because S3 is the bottleneck

$1 = \lambda_{S3}(0.019) \rightarrow \lambda_{S3} = 52.63$

$\lambda_{S3} = 0.4\lambda_{S2} + \lambda_a$

$= 0.4(\lambda_b + \lambda_{S3}) + \lambda_a$

$\lambda_b = 5\lambda_a$

$\therefore = 0.4(5\lambda_a + \lambda_{S3}) + \lambda_a$

$52.63 = 0.4(5\lambda_a + 52.63) + \lambda_a \rightarrow \lambda_a = 10.526$

$\lambda_b = 5 \cdot \lambda_a = 5(10.526) = 52.63$

$\lambda_{total} = \lambda_a + \lambda_b = 10.526 + 52.63 = 63.156$ → capacity

f). $\bar{T}_q = \frac{q}{\lambda} = \frac{33.25}{10+50} = 0.5541s$

Problem 3

The Karatepeli Systems Inc. enterprise has put together an online media conversion system. It allows users to submit multi-party media files comprised of potentially many data payloads. Some of these payloads are audio streams (AS), and others are video streams (VS).

In each file, there is a specific order and sequence in which the payloads need to be processed, for instance, a file might look like the following: AS→VS→VS→AS.... For the same file, the next payload cannot start processing until the previous one has completed processing. The system receives 11 requests per hour that begin with an AS payload. It receives 17 requests per hour that begin with a VS payload. $\lambda_{AS}=11/h$ $\lambda_{VS}=17/h$

The media conversion system is comprised of three subsystems. A single-CPU server, namely AUD, is dedicated to pre-process AS payloads, which on average takes 1.5 minutes. Another single-CPU server, namely VID, is dedicated to pre-process VS payloads which takes about 1.1 minutes. Once pre-processing is performed at either the VID or AUD server, the current payload being worked on is passed to the media converter server CONV which is a 3-CPU machine. Here, all the files being worked on are put into a single queue. Whenever any of the 3 CPUs becomes available, the file at the head of the queue can begin processing on the current payload. On average, it takes 2 minutes for a generic payload to be processed at the CONV machine. T_S CONV

Once processing for the current payload is completed at the CONV machine, the next payload can be an AS with probability 0.25 or VS with probability 0.4 and the file is routed to AUD or VID accordingly. The current payload can also be the final one (after which the response can be returned to the user) with probability 0.35.

a) Provide a diagram of the system and state the assumption that must hold for you to analytically solve the system.

b) How many payloads per hour are processed by the CONV machine? λ_{CONV}?

c) Which server is the bottleneck of the system? $\rho = \lambda \cdot T_S$. λ_S?

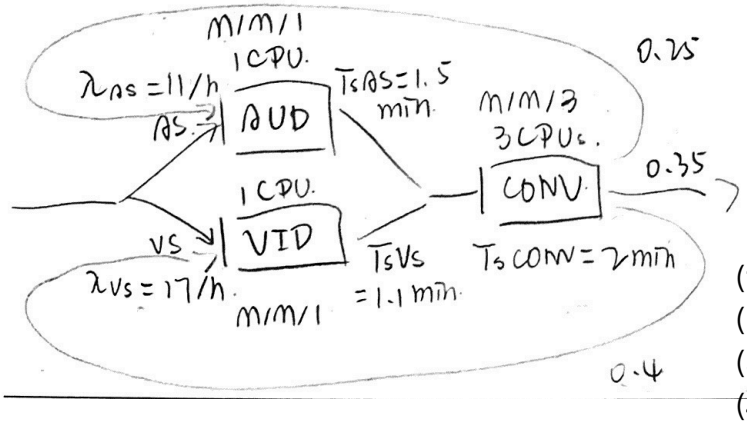
d) How many files, on average, are being concurrently handled by the entire system? q_{total}?

e) What is the average response time perceived by the end users when they submit a file for conversion to the system? $T_{q\ total}$?

f) Consider the case where the rate of arrival of new files at the VS does not change. By how much can the rate of arrival of new files at the AS increase while still allowing the system to reach steady-state? λ_{VS} λ_{AS}

g) What is the average number of payloads in each media file? $\lambda = M$. $\rho_{VS}=1$

h) What is the probability that a payload arriving at the CONV machine immediately starts service without having to wait in the queue?



(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).

$$\begin{cases} \lambda_{DS}' = \lambda_{DS} + (\lambda_{DS}' + \lambda_{VS}')\,0.25 \\ \lambda_{VS}' = \lambda_{VS} + (\lambda_{DS}' + \lambda_{VS}')\,0.4 \\ \lambda_{CONV} = \lambda_{DS}' + \lambda_{VS}' \end{cases}$$

$$\begin{cases} \lambda_{DS}' = 31 \\ \lambda_{VS}' = 49 \\ \lambda_{CONV} = 80 \end{cases}$$

c). $\rho = \lambda \cdot \bar{T}_S$ $\quad \bar{T}_{SDS} = 2\text{min} = 0.025\text{h}$.

$\rho_{DS} = \lambda_{DS}' \cdot \bar{T}_{SDS} = 31(0.025) = 0.775$

$\bar{T}_{SVS} = 1.1\text{min} = 0.0183\text{h}$

$\rho_{VS} = 49(0.0183) = 0.8967 \rightarrow$ bottleneck.

$\bar{T}_{SCONV} = 2\text{min} = 0.033\text{h}$ $\quad \rho = \frac{\lambda \bar{T}_S}{N} \rightarrow N = 3$.

$\rho_{CONV} = 80(0.033)\frac{1}{3} = 0.88$.

e). $q = \lambda \cdot \bar{T}_q$

$20.49 = (11+17)\cdot \bar{T}_q \rightarrow \bar{T}_q = 0.732\text{h}$.

d). $q = \frac{\rho}{1-\rho}$.

$$q_{DS} = \frac{\rho_{DS}}{1-\rho_{DS}} = \frac{0.775}{1-0.775} = 3.44$$

$$q_{VS} = \frac{\rho_{VS}}{1-\rho_{VS}} = \frac{0.8967}{1-0.8967} = 8.68$$

$$q = C\frac{\rho}{1-\rho} \qquad C = \frac{1-K}{\rho K} \qquad K = \frac{1+3\rho+\frac{(3\rho)^2}{2}}{1+3\rho+\frac{(3\rho)^2}{2}+\frac{(3\rho)^3}{6}}$$

$$K = \frac{1+3(0.88)+\frac{(3\cdot 0.88)^2}{2}}{1+3(0.88)+\frac{(3\cdot 0.88)^2}{2}+\frac{(3\cdot 0.88)^3}{6}} = 0.699$$

$$C = \frac{1-0.699}{1-0.88(0.699)} = 0.782$$

$$q_{CONV} = 0.782\left(\frac{0.88}{1-0.88}\right) + 3(0.88) = 8.37$$

$q_{total} = 3.44 + 8.68 + 8.37 = 20.49$

f). $\rho_{VS} = 1$

$1 = \lambda_{VS}'(0.0183) \rightarrow \lambda_{VS}' = 54.64$

$\lambda_{VS}' = \lambda_{VS} + (\lambda_{DS}' + \lambda_{VS}')\,0.4$

$54.64 = 17 + (\lambda_{DS}' + 54.64)\,0.4 \rightarrow \lambda_{DS}' = 39.46$

$\lambda_{DS}' = \lambda_{DS} + (\lambda_{DS}' + \lambda_{VS}')\,0.25$

$39.46 = \lambda_{DS} + (39.46 + 54.64)\,0.25 \Rightarrow \lambda_{DS} = 15.935$

$15.935 - 11 = 4.935$

g). $\frac{1}{0.35} = 2.857$

h). $C = 0.782$

$\therefore 1 - C = 1 - 0.782 = 0.218$