Queuing Systems

Computer Exercise 2

2022

1 Instructions

To be able to complete the computer exercise within the time limit, you have to prepare yourself by completing the home assignments before the exercise. If you have any questions regarding the home assignments, ask your tutor.

In this computer exercise, recording systems and queuing networks will analyzed theoretically and through simulations. Python files needed can be downloaded from the canvas pages of the course.

For every task, there are a number of questions to answer. Some of the questions are marked (**) and these shall be discussed with the tutor before proceeding with the computer exercise. You are supposed to do some thinking yourself before talking to the tutor. If the tutor is occupied elsewhere, you can continue until he becomes available.

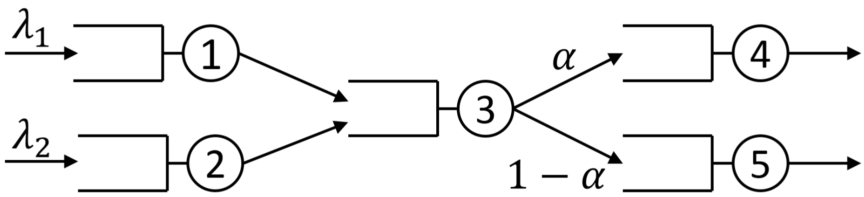

Figure 1: Queuing network,$\lambda_1 = 7.5, \lambda_2 = 10, \mu_1 = 10, \mu_2 = 14, \mu_3 = 22, \mu_4 = 9, \mu_5 = 11$ and $\alpha = 0.4$.

2 Home Assignments

These home assignments must be done before the laboratory. Feel free to ask for help if you need it!

2.1 Queuing networks, homework

Some facts that can be used to calculate e.g. the mean number of customers in the nodes of queuing networks:

- The output process from an M/M/1 system is a poisson process
- If you merge two poisson processes, you get a new poisson process
- If you divide a poisson process randomly, the fragmented processes are also poisson processes.

1. For the queuing network in Figure 1, the following applies: $\lambda_1 = 7.5, \lambda_2 = 10, \mu_1 = 10, \mu_2 = 14, \mu_3 = 22, \mu_4 = 9, \mu_5 = 11$ and $\alpha = 0.4$. Calculate the average number of customers in each of the queuing systems in the queue network. The nodes in the queuing network can be regarded as ordinary M /M/1 queues.

2. For each of the queuing systems in the queuing network, calculate the average number of customers waiting in the buffer.

3. What is the average number of customers in the entire queuing network?

4. Calculate $E(T_i)$ (mean time a job spends in the queuing system number i) for all queuing systems in the queuing network?

5. Modify the python file QueuingNetwork.py so that you can measure the number of customers in the buffer and the service time in each of the queuing systems. A hint: introduce the lists N_q, for the number of customers in the buffer and T_s, for the service times in the class queue.

6. Simulate the queuing network (simulation time 1000 seconds) with the values in figure 1 above. You should get the approximate results $E(N_{1q}) \approx 2.25$, $E(N_{3q}) \approx 3.1$, $E(T_{3s}) \approx 0.045$ and $E(T_{4s}) \approx 0.11$. Does the program give these results?

3 Lab assignment: Recording system

3.1 Limited number of users

The loss.py program can perform calculations on an M/M/m system with M users. It can calculate the probability of blocking and the probability that all the servers are busy (= the probability that the system is full).

1. From the program you get two graphs. One shows $P(block)$ and one shows the probability that the system is full. Which graph shows what? Justify the answer.

2. How many servers are there in the queuing system?

3. What happens to the curves when the number of customers becomes very large? (**)

3.2 Large number of users

AB Väntainte has one call center in Lund and one in Växjö, each of them with 10 customer recipients. As can be seen from the name of the company, there is no buffer where customers can wait if all customer recipients are busy, but then they are blocked. There is an average of 2 calls per minute to each customer center and on average it takes 5 minutes to help a customer.

1. Use the program in pkMMmloss.py to calculate $P(block)$.

2. At least, how many customer recipients are needed for $P(block)$ to be less than 5%? How large is the utilization of a server, i.e. how large proportion of the time does the customer receiver work?

3. In percent, how much does $P(block)$ increase if the arrival intensity increases by 10%? Use the number of customer recipients from the previous task.

4. AB Väntainte has decided to merge its two call centers into one that will be located in Hässleholm. The intensity of arrival at the new call center will be 4 calls per minute, ie the sum of the intensities for Lund and Växjö. How many customer recipients are needed in the new call center if the $P(block)$ is to be less than 5%? What will now be the utilization of the customer recipients?

5. Now increase the arrival intensity by 10% (use number of servants from the previous task). How much does $P(block)$ increase in percentage then?

6. What are the advantages and disadvantages of having two call centers compred to just one larger call center? (**)

4 Lab assignment: Queuing network

Figure shows 1 the queuing network that you should simulate in the lab.

1. Simulate the queue network (with QueuingNetwork.py) and compare your results with the calculations you have made in home assignment 1. Is it correct?

2. Is Little's theorem correct for the entire queuing network?

3. Increase the arrival intensity to queue system 2 by 20%. Now measure the average number of customers and the average time for a customer in each of the queuing systems.

4. Explain the result of the previous task. Where is the bottleneck in the queuing network?

5. How can you make the queuing network work better? Suggest at least two ways. (**)

4.1 Queuing network with M/G/1 system

We still use the queuing network in figure 1.

1. Change so that the service time is constant with the same average value as before, i.e. $1/\mu$ if the service rate of the exponential service time is μ. How does this change affect the time a customer spends in the queuing systems if you compare with when service times were exponentially distributed?

2. Change so that the service time has a hyper-exponential distribution. For an example of the hyperexponential distribution see page 118 in the book. If the average service rate for a queue is μ use the parameters $\alpha = 1/1.5, \mu_1 = 2\mu$ and $\mu_2 = \mu/2$. This means that we now have a greater variance than when we had exponentially distributed service times. How does the average time a customer spends in the queuing systems compare with when the service times were exponentially distributed?

3. Attempt to verify Little's theorem when service times are hyper-exponentially distributed.

4. Which of the five queuing systems are really M / G / 1 systems when the service times are deterministic? (**)

5 Lab assignment: load distribution

In many applications, e.g. cloud services, computer networking and logistics, load distribution is used. A simple queuing model of load distribution is that a dispatcher sends arriving jobs to a number of queuing systems. Jobs arrive to the dispatcher according to a poisson process. The dispatcher sends the jobs to a queuing system according to an algorithm, see figure 2.

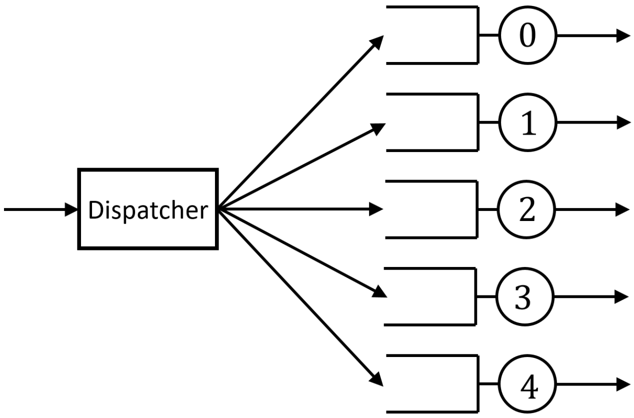

Figure 2: The queuing model of a dispatcher

Three algorithms the dispatcher can use are:

1. Chose one of the queuing systems randomly
2. Chose using round robin, i.e. first chose queuing system 0, then 1, then 2, then 3, then 4, then 0 again etc.
3. Chose the queuing system with the least number of customers

The file dispatcher.py contains a simulation program for this model. In this model we first assume that all the servers in the queuing systems have the same service rate, $\mu_i = 1$ for all i. The arrival rate to the dispatcher is $\lambda = 4$.

1. With these values, will the queuing systems be stable? Why?
2. Algorithm 2 and 3 are already implemented in dispatcher.py. Implement algorithm 1. Assume that a customer that arrives to the dispatcher is sent randomly to one of the queuing system with the same probability for all the queuing systems.
3. If algorithm 1 is used, calculate (using queuing theory) what the sum of the mean number of customers in all the queuing systems should be. Hint: when a poisson process is splitted randomly the resulting processes are also poisson processes.
4. When you run the simulation program and use your implementation of algorithm 1, do you get approximately the same value as in the previous task?

5. Simulate the system for all three algorithms. What is the mean of the total number of customers in the queuing systems for the three algorithms?

6. Which algorithm gives the shortest mean time in the system for a customer.

7. Change the service rates of the queues to $\mu_0 = 1$, $\mu_1 = 0.5$, $\mu_2 = 1.5$, $\mu_3 = 1$, $\mu_4 = 1$. Find the place in the code where these values are set and simulate with these values. Which of the algorithms work now? (**)