COMP9334 - Assignment 1

Hao Chen 5102446

Question 1 (3 marks)

An interactive computer system consists of three devices: a CPU and three disks (denoted by Disk1,Disk2 and Disk3). The system was monitored for 30 minutes and the following measurements were taken:

Number of completed jobs	1,231
Number of CPU accesses	2,789
Number of Disk1 accesses	17,412
Number of Disk2 accesses	13,424
Number of Disk3 accesses	15,978
CPU busy time	1102 seconds
Disk1 busy time	929 seconds
Disk2 busy time	1017 seconds
Disk3 busy time	1265 seconds
Think time	27 seconds

(a)Determine the service demand at each device of the system.

Since $D(j) = U(j) / X(0)$, $U(j) = B(j) / T$, $X(0) = C(0) / T$
therefore, the service demand

$$D(cpu) = U(cpu) / X(0) = B(cpu) / C(0) = 1102 / 1231 \approx 0.895 \text{ (s/jobs)}$$

$$D(disk1) = U(disk1) / X(0) = B(disk1) / C(0) = 929 / 1231 \approx 0.755 \text{ (s/jobs)}$$

$$D(disk2) = U(disk2) / X(0) = B(disk2) / C(0) = 1017 / 1231 \approx 0.826 \text{ (s/jobs)}$$

$$D(disk3) = U(disk3) / X(0) = B(disk3) / C(0) = 1265 / 1231 \approx 1.028 \text{ (s/jobs)}$$

(b)Use bottleneck analysis to determine the asymptotic bound on the system throughput when there are 40 active terminals.

Use bottleneck analysis to determine the asymptotic bound on the system throughput when there are 40 active terminals.

Since The Bottleneck analysis is (thinking time Z is considered)

$$X(0) \le \min\left[\frac{1}{\max Di}, \frac{N}{\sum_{i=1}^{K} Di + Z}\right]$$

the $\frac{1}{\max Di}$ is 1 / D(disk3) = 1231 / 1265 ≈ 0.973 (jobs/s)

and $\frac{N}{\sum_{i=1}^{K} Di + Z}$ the is 40 / ((1102+929+1017+1265)/1231+27) ≈ 14.24 (job/s)

So the asymptotic bound should be 0.973 (jobs/s)

(c)Using your results in Part (b), compute the minimum possible response time when the number of terminals is 40.

Since
the Response time + Think time = Number of terminals / System throughput

So the minimum response time = 40 / (1231 / 1265) - 27 ≈ 14.106 (s)

Question 2 (6 marks)

A call centre has 4 operators. Calls arrive at the call centre obey the Poisson distribution with a rate of 20 calls per hour. The service time required by each call is exponentially distributed with mean service time 10 minutes.

(a)Assuming the call centre has no facilities to place an incoming call on hold. This means that if all the operators are busy, an incoming call will be rejected. Compute the probability that an incoming call is rejected.

Note: You do not need to derive the Markov chain for this part. You are allowed to apply standard results from Queueing Theory.

This is a M/M/4/4 model.

Based on the given fact,
λ = 20 (call/hour)
S = 1/6 (hour/call), So μ = 1/S = 6 (call/hour)

Based on the equation
The probability that an arriving call is rejected
= The probability that m operator is all busy

$$= \quad P_m = \frac{\frac{\rho^m}{m!}}{\sum_{k=0}^{m} \frac{\rho^k}{k!}} \quad \text{where} \quad \rho = \frac{\lambda}{\mu}$$

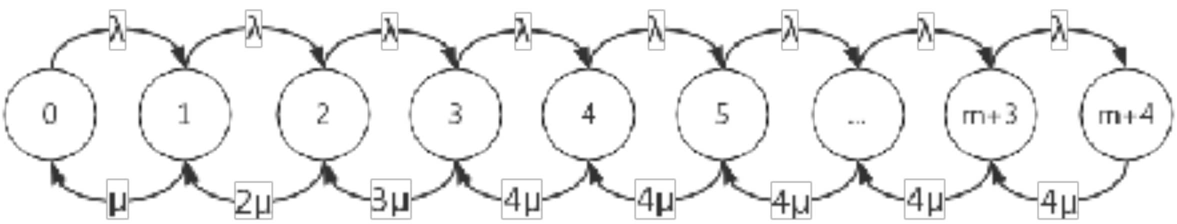
here, ρ = λ / μ = 20 / 6 = 10 / 3 , m = 4
P(rejected) = ((ρ^4)/ρ!)/(1+ρ+ρ^2/2!+ρ^3/3!+ρ^4/4!)
≈ 0.243

So the probability that one incoming call is rejected is 0.243

(b) The owner of the call centre would like to decrease the call rejection rate to less than 50% of the value calculated in Part (a). The owner decides to achieve this by introducing a queue which places the incoming calls on hold when all the operators are busy. The holding queue will consist of M holding slots where M is to be determined. If an incoming call arrives when all operators are busy, it will be placed in the holding queue provided that a vacant holding slot is available. If the call arrives when all operators are busy and all M holding slots are used, then the call is rejected. Assuming that the customers are infinitely patient in the sense that once their call is accepted in the (holding) queue, they will wait until they get to the operator and will only leave the system after they have been served.

(i) Formulate a continuous-time Markov chain for the call centre with 4 operators and M holding slots. Your formulation should include the definition of the states and the transition rates between states.

The Markov chain for the call centre with 4 operators and M holding slots.



This is the Markov chain for the call centre model.
P(0) means the probability that no call in the centre right now, while P(k) means the probability that k calls in the centre.
The max number in the queue is m, and there could be at most m+4 in the centre, 4 in operators and m in the queue.

(ii) Write down the balance equations for the continuous-time Markov chain that you have formulated in Part (b,i).

Define the states of the queue,
State 0 = There is zero job in the system
State 1 = There is 1 job in the system
…
State k = There are k jobs in the system
then we have

$\lambda P(0) = \mu P(1)$ => $P1 = \lambda/\mu * P(0)$
$\lambda P(1) = 2\mu P(2)$ => $P2 = \lambda/2\mu * P(1) = 1/2 * (\lambda/\mu)^2 * P(0)$
$\lambda P(2) = 3\mu P(3)$ => $P3 = \lambda/3\mu * P(2) = 1/6 * (\lambda/\mu)^3 * P(0)$
$\lambda P(3) = 4\mu P(4)$ => $P4 = \lambda/4\mu * P(3) = 1/24 * (\lambda/\mu)^4 * P(0)$
$\lambda P(4) = 4\mu P(5)$ => $P5 = \lambda/4\mu * P(4) = 1/96 * (\lambda/\mu)^5 * P(0)$
…
$\lambda P(m+2) = 4\mu P(m+3)$ =>
$P(m+3) = \lambda/4\mu * P(m+2) = 1/(24*4^{(m-1)}) * (\lambda/\mu)^{(m+3)} * P(0)$

λP(m+3) = 4μP(m+4) =>
P(m+4) = λ/4μ * P(m+3) = 1/(24*4^m) * (λ/μ)^(m+4) * P(0)
The probability of all states' sum is 1.
P(0) + P(1) + P(2) + P(3) +…+ P(m+4) = 1

Since , $\rho = \frac{\lambda}{\mu}$ we have

$$P(0)\left(1+\varrho+\frac{1}{2}\varrho^2+\frac{1}{6}\varrho^3+\frac{1}{24}\varrho^4+\ldots+\frac{1}{24\cdot 4^{m-1}}\varrho^{m+3}+\frac{1}{24\cdot 4^{m}}\varrho^{m+4}\right)=1$$

and it can be simplified to the equations below

$$P(0)\left[\sum_{k=0}^{4}\frac{1}{k!}\varrho^k+\sum_{i=5}^{m+4}\frac{1}{24\cdot 4^{k-4}}\varrho^k\right]=1$$

<u>(iii) Derive expressions for the steady state probabilities of the continuous-time Markov chain that you have formulated.</u>

The expression for the steady steady state probabilities can be formulated as below:

$$P(k)=\begin{cases} P(0)\frac{1}{k!}\varrho^k & k\le 4 \\ P(0)\frac{1}{24\cdot 4^{k-4}}\varrho^k & k>4 \end{cases}$$

Based on the (ii), the P(0) can be calculated by the below equation.

$$P(0)=\left[\sum_{k=0}^{4}\frac{1}{k!}\varrho^k+\sum_{k=5}^{m+4}\frac{1}{24\cdot 4^{k-4}}\varrho^k\right]^{-1}$$

(iv) Use your answer in Part (b,iii) to determine the smallest value of M required to reduce the call rejection rate to less than 50% of the value calculated in Part (a).

Note: There are many (in fact, infinite) choices of M that can reduce the call rejection rate to less than 50% of the value calculated in Part (a). We are only interested in the smallest value of such M's.

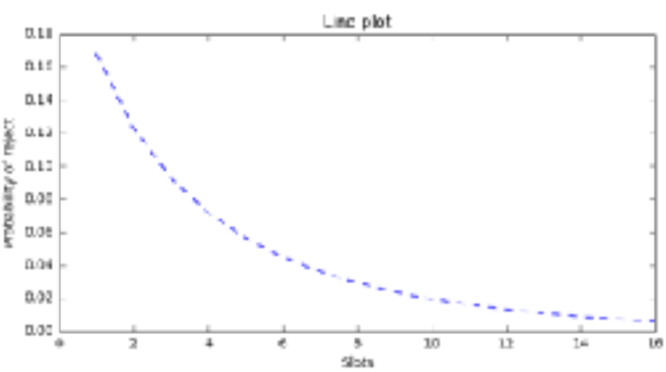
I use python as the tool to calculate the P(0) and P(k) when the k is growing and each k with different waiting slots.
(The code and the chart can be found in the A1_Qb.py)

The example solution can be shown as this

```
---------------------------------------------
Print the details as followings:
k from 1 - 20 and m from 1 - 16
---------------------------------------------
k is: 1, P(0) is: 0.23076923076923073, P(k) is: 0.7692307692307692
---------------------------------------------
k is: 2, P(0) is: 0.10112359550561797, P(k) is: 0.5617977528089888
---------------------------------------------
k is: 3, P(0) is: 0.06225980801537279, P(k) is: 0.3843197540353575
---------------------------------------------
k is: 4, P(0) is: 0.047156099552470405, P(k) is: 0.24257713053037066
---------------------------------------------
k is: 5, m is: 1, P(0) is: 0.03922729229444623, P(k) is: 0.16815540249677144
---------------------------------------------
k is: 6, m is: 2, P(0) is: 0.034405990706810126, P(k) is: 0.12290665387126462
---------------------------------------------
k is: 7, m is: 3, P(0) is: 0.03120045559454871, P(k) is: 0.09790657028364087
---------------------------------------------
k is: 8, m is: 4, P(0) is: 0.028966786172013006, P(k) is: 0.07185865524440036
---------------------------------------------
```

For example, the P(4) = 0.2425

Since no call will be rejected if k < operator + slots, so the probability of rejection will always equal to P(m+4). Also from the chart, we could now that the m we want will defiantly in the front of the x-axle.
P(k=m+4) < 0.5*P(m=0) = 0.1211
The relation can be also shown in this linear chart.



The smallest m is 3. P(m=3) = P(3+4=7) ≈ 0.093 < 0.5*P(m=0) = 0.1213
So, P(7) = 0.093 satisfy the requirement. (details in A1_Qb.py file)

(v) For the value of M that you have calculated in Part (b,iv), determine how long an accepted call will need to wait before it will be served by an operator.

Note: If you use a computer program to derive your numerical answers, you must include your computer program in your submission. Do not forget to show us your steps to obtain your answer.

Based on the little's law, we know that N = X*R

While when the waiting slots is 3, we have to find the sum of calls in the system.

Hence, $N = \sum_{k=0}^{7} k \cdot p(k) \approx 3.659$

Then we find the X(0), which is the rate that calls arrival the centre:

$$X(0) = (1\text{-}P(\mathrm{re}ject)) \cdot \lambda$$

X(0) = (1-0.0929) * 20 = 18.142 / h

So

R = N/X(0) = 3.6591 / 18.142 = 0.2017 h ≈ 726.07 s

Then the call need to wait before it be served is

T(wait) = R - T(service) = 726.07 - 600 = 126.07 s

Question 3 (6 marks)

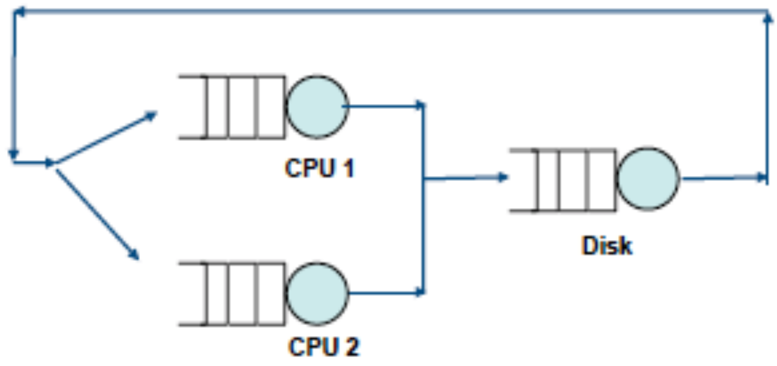


Figure 1: Figure for Question 3.

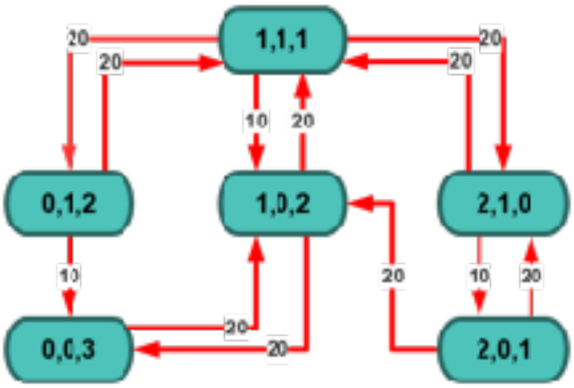
Consider the computer system shown in Figure 1. The system consists of three devices: a disk and 2 CPUs. Each device is modelled as a server and a queue. The system is at peak load and there are three (3) jobs circulating in the system at all times. During each round that a job circulates the system, the job requires processing from one of the CPUs and then followed by the disk. Assuming that:

- The processing time required by each job per visit to the disk is exponentially distributed with mean 50 milli-seconds.
- The two CPUs have different mean processing times. The mean processing times for CPU1 and CPU2 are, respectively, 50 andu 100 milli-seconds. Both processing time distributions are assumed to be exponential.
- After a job has left the disk, it will proceed to receive processing at one of the CPUs immediately. In any attempt to utilise the faster CPU (i.e. CPU1), a job will only be sent to CPU2 if CPU2 is idle and CPU1 is busy. In other words, if CPU2 is busy, the job will be sent to CPU1; and if both CPU1 and CPU2 are idle, the job will be sent to CPU1.

Answer the following questions.

(a)Let the states be the following 3-tuple: (number of users in the CPU1, number of users in CPU2, number of users in the disk), formulate a continuous-time Markov chain for this computer system. Your formulation should include (1) a list of states; (2) the transition rates between the states.

There are six possible states (1,1,1), (0,1,2), (1,0,2), (2,1,0), (0,0,3), (2,0,1)
Then the Markov chain can be drawn as below:



Note that states (0,1,2) can transfer to (0,03) while it cannot go back.
Also (2,0,1) to (1,0,2) only.

$$\mu(\text{disk}) = 1/(0.05) = 20\ /s$$
$$\mu(\text{cup1}) = 1/(0.05) = 20\ /s$$
$$\mu(\text{cup2}) = 1/(0.1) = 10\ /s$$

(b) Write down the balance equations for the continuous-time Markov chain that you have formulated in Part (a).

From part(a) we can find the balance equation:

50*P(1,1,1) - 20*P(0,1,2) - 20*P(1,0,2) - 20*P(2,1,0) + 0*P(0,0,3) + 0*P(2,0,1) = 0
-20*P(1,1,1) + 30*P(0,1,2) - 0*P(1,0,2) - 0*P(2,1,0) + 0*P(0,0,3) + 0*P(2,0,1) = 0
-10*P(1,1,1) - 0*P(0,1,2) + 40*P(1,0,2) - 0*P(2,1,0) - 20*P(0,0,3) - 20*P(2,0,1) = 0
-20*P(1,1,1) - 0*P(0,1,2) - 0*P(1,0,2) + 30*P(2,1,0) + 0*P(0,0,3) - 20*P(2,0,1) = 0
0*P(1,1,1) - 10*P(0,1,2) - 20*P(1,0,2) - 0*P(2,1,0) + 20*P(0,0,3) + 0*P(2,0,1) = 0
0*P(1,1,1) + 0*P(0,1,2) + 0*P(1,0,2) - 10*P(2,1,0) + 0*P(0,0,3) + 40*P(2,0,1) = 0

P(1,1,1) + P(0,1,2) + P(1,0,2) + P(2,1,0) + P(0,0,3) + P(2,0,1) = 0

(c) What is the steady state probability for each state?

Note that the balance equations are not independent, so we can only choose first five equations and the last one to solve. (details in A1_Qc.py file)

$P(1,1,1) \approx 0.1974$
$P(0,1,2) \approx 0.1316$
$P(1,0,2) \approx 0.2039$
$P(2,1,0) \approx 0.1579$
$P(0,0,3) \approx 0.2697$
$P(2,0,1) \approx 0.0395$

(d) What is the throughput of the system?

The throughput of the system is exactly the throughput of the disk, so we need to find the utilisation of the disk first.

U(disk) = 1- P(2,1,0) = 0.8421 (Since the P(2,1,0) is the only idle state.)

So

X(disk) = U(disk)/S(disk) = 0.8421/0.05 ≈ 16.842 /s

So

X(0) = X(disk) = 16.842 /s

(e) What is the mean response time of the CPU1?

Based on the Little's law: N = XR

$$N(cpu1) = \sum_{k=0}^{k} n(k) \cdot p(cpu1)$$

So N(cpu1) = P(1,0,2) + P(1,1,1) + 2*P(2,1,0) + 2*P(2,0,1) = 0.7961

The throughput of cpu1 is:

X(cpu1) = U(cpu1)/S(cpu1)
= P(1,0,2) + P(1,1,1) + P(2,1,0) + P(2,0,1) = 0.5987/0.05
≈ 11.974/s

The Response time is

R = N(cpu1)/X(cpu1) = 0.7961/11.974 ≈ 0.067s

(f) How long does a user have to wait, on average, at the disk before it gets served?

Note: If you use a computer program to solve for the steady state probabilities, you need to show us your code. Also, do not forget to show us the steps you use to get your answers.

Based on Little's law: N = XR

$$N(disk) = \sum_{k=0}^{k} n(k) \cdot pk(disk)$$

So

N(disk) = 3*P(0,0,3) + 2*P(1,0,2) + 2*P(0,1,2) + P(1,1,1) + P(2,0,1) = 1.717

Since throughput is already got from c),So

R = N(disk)/X(0) = 1.717/16.842 ≈ 0.1019s

Then the waiting time T(wait)

T(wait) = R - T(service) = 0.1019 - 0.05 = 0.0519s