COMP9334-Assignment, Session 1, 2018

Jingxuan Li z5132086

Question 1 (2 marks)

An interactive computer system consists of a CPU and a disk. The system was monitored for 60 minutes and the following measurements were taken:

Number of completed jobs	1267
Number of CPU accesses	2,178
Number of disk accesses	2,412
CPU busy time	2,929 seconds
Disk busy time	2,765 seconds

Answer the following questions.

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

X(0) = 1267/(60*60) ≈0.35 jobs/s
The utilisation of the cpu = 2929 / 3600 ≈ 0.81
The utilisation of the disk = 2765 / 3600 ≈ 0.768

By service demand law, the service demand at the cpu is the utilisation of the cpu / system throughput:
D(cpu) = U(cpu) / X(0) = 0.81 / 0.35 ≈ 2.31 s
D(disk) = U(disk) / X(0) = 0.768 / 0.35 ≈ 2.19 s

(b) Use bottleneck analysis to determine the asymptotic bound on the system throughput when there are 20 active terminals and the think time per job is 14 seconds.

According to the throughput bound:

$$X(0) \leq \min\left[\frac{1}{\max D_i}, \frac{N}{\sum_{i=1}^{K} D_i}\right]$$

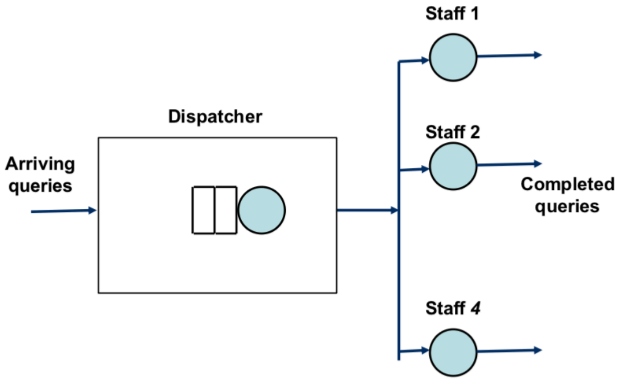
$$\frac{1}{\max D_i}$$ = 1/D(cpu) = 1267/2929≈0.43 jobs/s

$$\frac{N}{\sum_{i=1}^{K} D_i}$$ =20 / ((2929+2765)/1267+14)≈1.08 jobs/s

Hence, the asymptotic bound on the system throughput should be 0.43 jobs/s

Question 2 (7 marks)

A company has a support centre to deal with internal queries from its employees. The support centre currently has 4 staff. The centre has an automatic dispatcher to direct the calls to the support staff. The dispatcher currently has a queue that can hold 2 calls. However, there are no queueing facilities at the staff's terminals. The queueing network at the support centre is depicted in Figure 1.



The voice-over-IP record shows that the centre is getting on average 15 queries per hour. The arrivals can be modelled by using Poisson distribution.

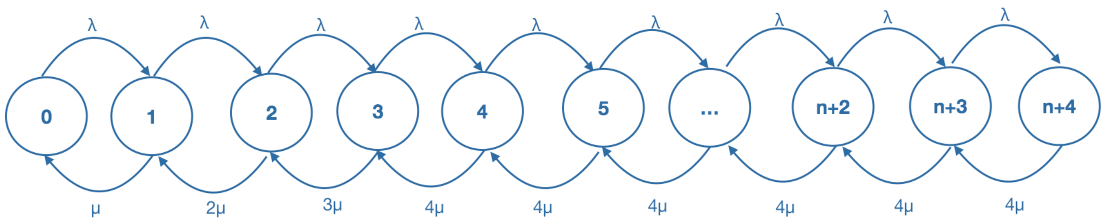
The record also shows that each support staff can complete on average 3 queries per hour. The amount of time required by each query is exponentially distributed.

When a query arrives at the dispatcher, it will accept the query if the dispatcher queue is not full, otherwise the query will be rejected. If a query is accepted and the queue is not empty, the query will be placed at the end of the queue. If a query is accepted and the queue is empty, then the query will be placed in the queue if all staff are busy, otherwise it will be sent to an idling staff. A query will leave the system after its processing is completed. Whenever a staff becomes idle, he/she will take the query from the front of the queue if there is one.

Answer the following questions:

(a) Formulate a continuous-time Markov chain for a system similar to that described above with 4 staff and n waiting slots. Your formulation should include the definition of the states and the transition rates between states. Note that we ask you to use n waiting slots because you will be varying the number of waiting slots in later parts of this question.

The continuous-time Markov chain for a system similar to that described above with 4 staff and n waiting slots.



Definite of the state:

State = 0: all staffs are idle,

State = 1: 1 staff is busy,

State = 2: 2 staffs are busy,

State = 3: 3 staffs are busy,

State = 4: all staffs are busy,

State = 5: all staffs are busy, and one query in queue

...

State = n+4: all staffs are busy, and n queries in queue

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

Based on the above diagram, the balance equations are

$\lambda P_0 = \mu P_1$

$\lambda P_0 + 2\mu P_2 = (\lambda+\mu)P_1$

$\lambda P_1 + 3\mu P_3 = (\lambda+2\mu)P_2$

$\lambda P_2 + 4\mu P_4 = (\lambda+3\mu)P_3$

$\lambda P_3 + 4\mu P_5 = (\lambda+4\mu)P_4$

$\lambda P_4 + 4\mu P_6 = (\lambda+4\mu)P_5$

...

$4\mu P_{n+4} = \lambda P_{n+3}$

(c) Derive expressions for the steady state probabilities of the continuous-time Markov chain that you have formulated.

Let $\rho = \lambda/\mu$

$P_1 = \rho P_0$

$P_2 = \rho^2/2 P_0$

$P_3 = \rho^3/6 P_0$

$P_4 = \rho^4/24 P_0$

$P_5 = \rho^4/24 * (\rho/4) P_0$

$P_6 = \rho^4/24 * (\rho/4)^2 P_0$

...

$P_{n+4} = \rho^4/24 * (\rho/4)^n P_0$

The expressions for steady state can be formulated as blew:

$$\mathrm{P(k)} = \begin{cases} P_0 \dfrac{\rho^k}{k!} & k \leq 4 \\ P_0 \dfrac{\rho^k}{24 \times 4^{k-4}} & k > 4 \end{cases}$$

Since $P_0 + P_1 + P_2 + P_3 + P_4 + P_5 + \ldots = 1$,

then $(\rho + \rho^2/2 + \rho^3/6 + \rho^4/24 + \rho^4/24 * (\rho/4) + \ldots + \rho^4/24 * (\rho/4)^n) * P_0 = 1$

So $$P_0 = \frac{1}{\sum_{i=0}^{4} \frac{\rho^i}{i!} + \frac{\rho^4}{24} \times \sum_{i=5}^{k} (\frac{\rho}{4})^{i-4}}$$

(d) For the current configuration, i.e. for n = 2, determine:

(i) The probability that an arriving query will be rejected. Let us denote the result of this by x.

λ = 15 queries/hour

μ = 3 queries /hour

For n = 2, an arriving query will be rejected when all staffs are busy and two waiting slots in queue. It should be State 6 as mentioned above.

According to q2.py file: when $\mathrm{k} = 6$, $\mathrm{P}(6) \approx 0.2935$

(ii) The mean waiting time of an accepted query in the queue.

Based on the Little's Law: N = XR

The sum of calls in the system: N = $\sum_{0}^{6} k * P(k)$

X = (1 – P(6)) * λ

The response time: R = N / X

The waiting time: W = R – S

According to q2.py file:

Response time: 1479.2256s

Waiting time: 279.2256s

(e) Assuming that you are the manager of the support centre and you think the current rejection rate is too high. You decide to add more waiting slots at the dispatcher to try to reduce the rejection rate. Determine the blocking probability if you add 5, 10, 15 and 20 waiting slots.

According to q2.py file:

waiting slots: 7, p0: 0.0017984046540452734 , pk: 0.22331931368078706

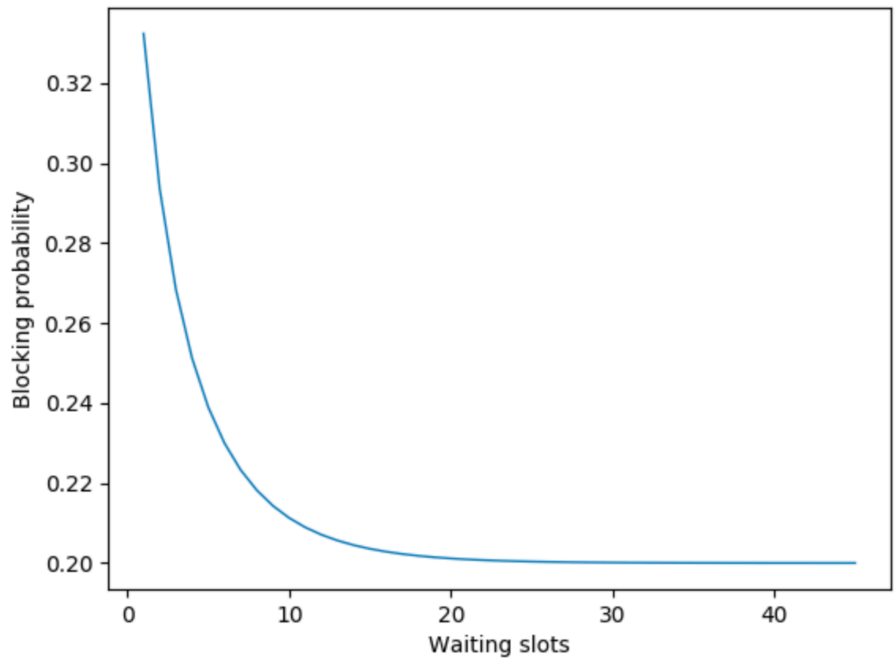
waiting slots: 12, p0: 0.0005464638257809098 , pk: 0.20708581427429246

waiting slots: 17, p0: 0.00017489922406055817 , pk: 0.20226785993865193

waiting slots: 22, p0: 5.68773652984537 97e-05 , pk: 0.2007375098367033

(f) Explain why there is little drop in blocking probability after adding 10 waiting slots. What should you do to reduce the blocking probability?

According to q2.py file:

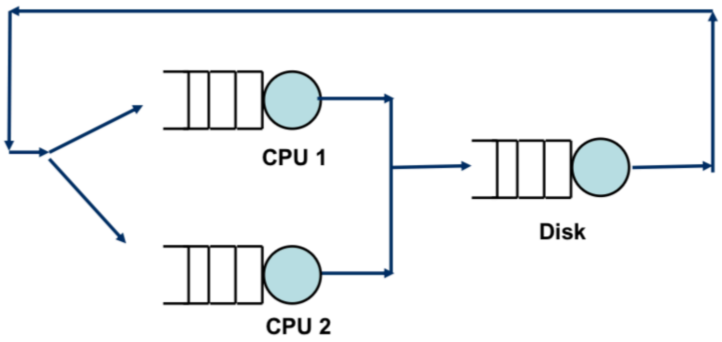


waiting slots: 1 , p0: 0.0017984046540452734 , pk: 0.22331931368078706
waiting slots: 2 , p0: 0.001405938209121901 , pk: 0.218230332111614
waiting slots: 3 , p0: 0.0011046131035646958 , pk: 0.21432314990955556
waiting slots: 4 , p0: 0.000871211983016017 , pk: 0.21129671537977435
waiting slots: 5 , p0: 0.0006891840701282998 , pk: 0.20893642010933028
waiting slots: 6 , p0: 0.0005464638257809098 , pk: 0.20708581427429246
waiting slots: 7 , p0: 0.0004340951430636066 , pk: 0.20562876702172472
waiting slots: 8 , p0: 0.00034533231928058304 , pk: 0.2044778090733382
waiting slots: 9 , p0: 0.00027503430432109817 , pk: 0.20356627814603023
waiting slots: 10 , p0: 0.00021924555283319518 , pk: 0.20284288400173708
waiting slots: 11 , p0: 0.00017489922406055817 , pk: 0.20226785993865193
waiting slots: 12 , p0: 0.00013960277969704714 , pk: 0.2018101827100717
waiting slots: 13 , p0: 0.00011148042382192381 , pk: 0.2014455294955576
waiting slots: 14 , p0: 8.905560655152405e-05 , pk: 0.20115475436495145
waiting slots: 15 , p0: 7.116231025283476e-05 , pk: 0.20092273795627846
waiting slots: 16 , p0: 5.6877365298453797e-05 , pk: 0.2007375098367033
waiting slots: 17 , p0: 4.546835887683262e-05 , pk: 0.20058957305343628
waiting slots: 18 , p0: 3.635325420238365e-05 , pk: 0.2004713805294909
waiting slots: 19 , p0: 2.906890084803346e-05 , pk: 0.20037692674766283
waiting slots: 20 , p0: 2.3246358504120807e-05 , pk: 0.2003014277819368
……

As it shown above, after adding 10 waiting slots, p0 p approaches 0, so pk gets little drop.

To reduce the blocking probability, we should reduce ρ by reducing μ. In other word, we should reduce the mean service time.

Question 3 (6 marks)



Consider the computer system shown in Figure 2. 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 four (4) 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 200 milli-seconds.
- The two CPUs have different mean processing times. The mean processing times for CPU1 and CPU2 are, respectively, 200 and 400 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 an attempt to utilise the faster CPU (i.e. CPU1), the choice of the CPU depends on the number of jobs at CPU2 at the time when a job leaves the disk. The following job assignment strategy is employed: (a) If at the time a job leaves the disk, the total number of jobs at CPU2 is less than 2, then there is an equal probability for that job to go to CPU1 or CPU2. (b)If at the time a job leaves the disk, the total number of jobs at CPU2 is 2 or more, then that job will to go 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 12 possible 3-tuple states (#cpu1, #cpu2, #disk):

(4,0,0),(3,1,0),(2,2,0),(3,0,1),(2,1,1),(1,2,1),(2,0,2),(1,1,2),(0,2,2),(1,0,3),(0,1,3),(0,0,4)

μ(disk) = 1/ 0.2 = 5 jobs/s

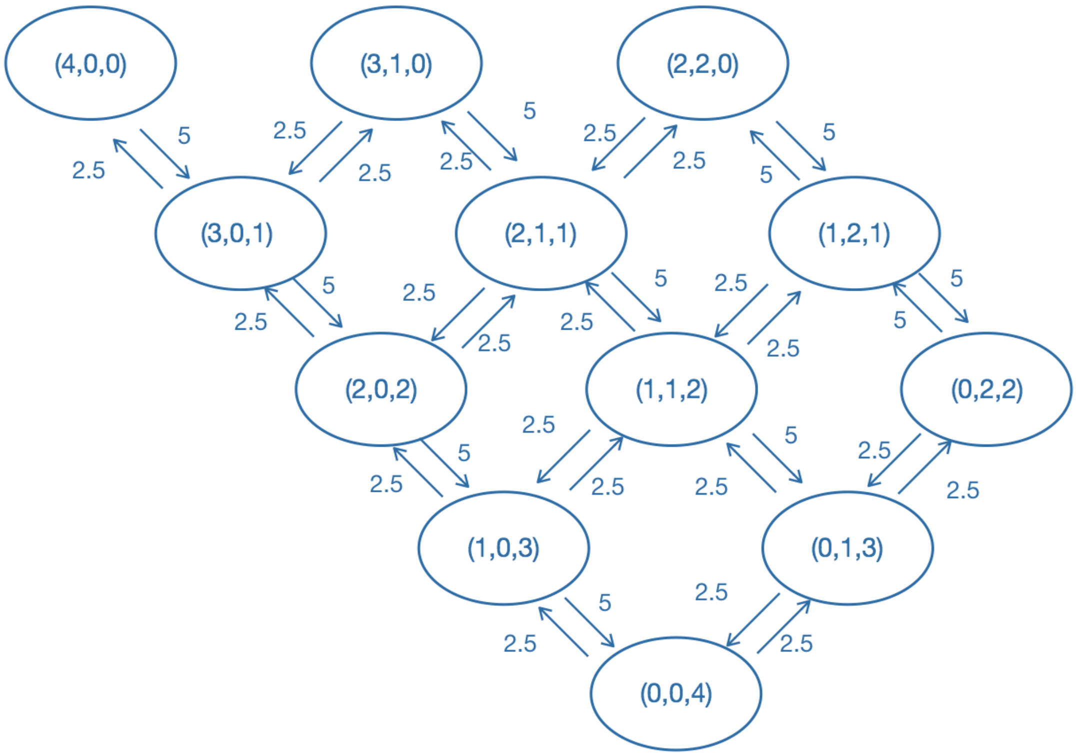
μ(cpu1) = 1/ 0.2 = 5 jobs/s

μ(cpu2) = 1/ 0.4 = 2.5 jobs/s

(a) when a job leaves the disk, the total number of jobs at CPU2 is less than 2, then there is an equal probability for that job to go to CPU1 or CPU2, then the μ should be 5/2 = 2.5 jobs/s

(b) If at the time a job leaves the disk, the total number of jobs at CPU2 is 2 or more, then that job will to go to CPU1. For example, (0,2,2) -> (1,2,1) and (1,2,1) -> (2,2,0). The μ should be 5 jobs/s

Therefore, the Markov chain can be draw as below:



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

$$
A = \begin{pmatrix}
5 & 0 & 0 & -2.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 7.5 & 0 & -2.5 & -2.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 7.5 & 0 & -2.5 & -5 & 0 & 0 & 0 & 0 & 0 & 0 \\
-5 & -2.5 & 0 & 10 & 0 & 0 & -2.5 & 0 & 0 & 0 & 0 & 0 \\
0 & -5 & -2.5 & 0 & 12.5 & 0 & -2.5 & -2.5 & 0 & 0 & 0 & 0 \\
0 & 0 & -5 & 0 & 0 & 12.5 & 0 & -2.5 & -5 & 0 & 0 & 0 \\
0 & 0 & 0 & -5 & -2.5 & 0 & 10 & 0 & 0 & -2.5 & 0 & 0 \\
0 & 0 & 0 & 0 & -5 & -2.5 & 0 & 12.5 & 0 & -2.5 & -2.5 & 0 \\
0 & 0 & 0 & 0 & 0 & -5 & 0 & 0 & 7.5 & 0 & -2.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -5 & -2.5 & 0 & 10 & 0 & -2.5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -5 & -2.5 & 0 & 7.5 & -2.5 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
$$

$$
x = \begin{pmatrix} P(4,0,0) \\ P(3,1,0) \\ P(2,2,0) \\ P(3,0,1) \\ P(2,1,1) \\ P(1,2,1) \\ P(2,0,2) \\ P(1,1,2) \\ P(0,2,2) \\ P(1,0,3) \\ P(0,1,3) \\ P(0,0,4) \end{pmatrix}
\qquad
B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

balance equations: A * x = B

(c) What are the steady state probabilities for each state?

According to q3.m file:

$P(4,0,0) \approx 0.0130$
$P(3,1,0) \approx 0.0277$
$P(2,2,0) \approx 0.0871$
$P(3,0,1) \approx 0.0259$
$P(2,1,1) \approx 0.0572$
$P(1,2,1) \approx 0.1021$
$P(2,0,2) \approx 0.0501$
$P(1,1,2) \approx 0.0935$
$P(0,2,2) \approx 0.1213$
$P(1,0,3) \approx 0.0912$
$P(0,1,3) \approx 0.1598$
$P(0,0,4) \approx 0.1711$

(d) What is the throughput of the system?

The throughput of the system is the throughput of disk.

U(disk) = 1 – P(4,0,0) – P(3,1,0) – P(2,2,0) = 0.8722

X(disk) = U(disk) $\times$ μ(disk) = 0.8722 $\times$ 5 = 4.361 transitions /s

(e) What is the mean number of jobs in CPU1?

N(cpu1) = 4$\times$P(4,0,0)+3$\times$P(3,1,0)+2$\times$P(2,2,0)+3$\times$P(3,0,1)+2$\times$P(2,1,1)+P(1,2,1)+2$\times$P(2,0,2)+P(1,1,2)+P(1,0,3)= 4$\times$ $0.0130 + 3 \times 0.0277 + 2 \times 0.0871 + 3 \times 0.0259 + 2 \times 0.0572 + 0.1021 + 2 \times 0.0501 + 0.0935 + 0.0912 =$ 0.8884 jobs

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

Based on the Little's Law: N = XR

X(cpu1) = U(cpu1) $\times$ μ(cpu1) = (P(4,0,0)+ P(3,1,0)+ P(2,2,0)+ P(3,0,1)+ P(2,1,1)+P(1,2,1)+ P(2,0,2)+P(1,1,2)+P(1,0,3)) $\times$ 5 = 0.5478 $\times$ 5 = 2.739 transitions /s

R = N(cpu1) / X(cpu1) = 0.8884/2.739 $\approx$ 0.3243 s