COMP9334 Capacity Planning Assignment, Session 1, 2018

Senlin Deng, z5014567

Question1:

a).

From **service demand** law: $D_{(j)} = U_{(j)} / X_{(0)}$
By definition **throughput**: $X_{(0)} = C_{(0)} / T$
By definition **utilization:** $U_{(j)} = B_{(j)} / T$

From above three equations, it can be delivered that: $D_{(j)} = B_{(j)} / C_{(0)}$

As a result, the service demand:

$D_{(CPU)} = B_{(CPU)} / C_{(0)} = 2929 / 1267 \approx 2.312$ s/jobs
$D_{(Disk)} = B_{(Disk)} / C_{(0)} = 2765 / 1267 \approx 2.182$ s/jobs

b).

From the Bottleneck analysis (thinking time is considered):

$$x_0 \le \min[\frac{1}{\max(D_{(j)})}, \quad \frac{N}{\sum_{i=1}^{K} D_{(i)} + Z}]$$

For the first part, the minimum value is:

$\min = 1/D_{(CPU)} = 1/2.312 \approx 0.433$

For the second part, the minimum value is:

$\min = 20 / (2.312 + 2.182 + 14) \approx 1.081$

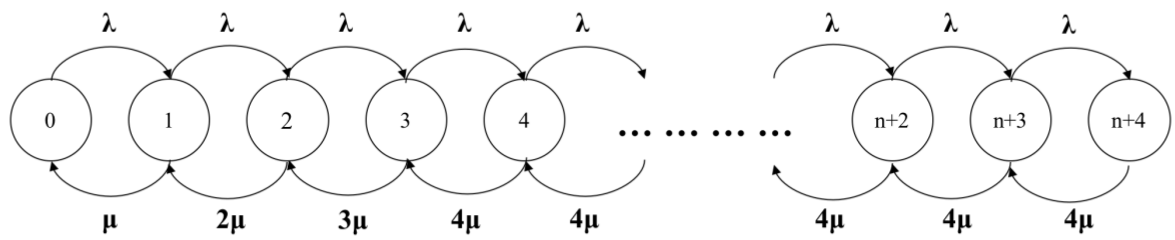
Since 0.433 < 1.018, the asymptotic bound should be 0.433 jobs/s

Question2:

a). The Markov Chain of a system, in which it has 4 staff (operators) and n waiting slots. Each circle with a number illustrates a specific probability of a state.

State 0: idle
State 1: 1 job in the system
State 2: 2 jobs in the system
… …
… …
… …
State n: n jobs in the system

λ = mean arrival rate
μ = service rate



b). The balance equations for the Markov Chain:

$P_{(k)}$: The probability that k jobs in the system, where k = 0, 1, 2……, n, n+1, n+2, n+3, n+4. Since there are 4 operators as well as n slots of queue.

$\lambda * P_{(0)} = \mu * P_{(1)}$
$\lambda * P_{(1)} = 2\mu * P_{(2)}$
$\lambda * P_{(2)} = 3\mu * P_{(3)}$
$\lambda * P_{(3)} = 4\mu * P_{(4)}$
$\lambda * P_{(4)} = 4\mu * P_{(5)}$
$\lambda * P_{(5)} = 4\mu * P_{(6)}$
……
……
$\lambda * P_{(n+2)} = 4\mu * P_{(n+3)}$
$\lambda * P_{(n+3)} = 4\mu * P_{(n+4)}$

The sum of all probabilities will be 1:
$P_{(0)} + P_{(1)} + P_{(2)} + P_{(3)} + P_{(4)} + \ldots\ldots + P_{(n+2)} + P_{(n+3)} + P_{(n+4)} = 1$

Also, we have $\rho = \frac{\lambda}{\mu}$
From the above two equations, we have:

$P_{(0)} * (1 + \rho + \frac{1}{2} * \rho^2 + \frac{1}{6} * \rho^3 + \frac{1}{24} * \rho^4 + \cdots + \frac{1}{24*4^{n-1}} * \rho^{n+3} + \frac{1}{24*4^n} * \rho^{n+4}) = 1$

$P_{(0)} = \frac{1}{\sum_{k=0}^{4} \frac{1}{k!} * \rho^k + \frac{1}{24} * \sum_{k=5}^{k=n+4} \frac{1}{4^{(k-4)}} * \rho^k}$ (where $\rho = \frac{\lambda}{\mu}$)

c). Steady state

Steady state means: Rate of entering a state = Rate of leaving a state = λ

From question b). the equation should be:

$P_{(0)} = \frac{1}{\sum_{k=0}^{4} \frac{1}{k!} * \rho^k + \frac{1}{24} * \sum_{k=5}^{k=n+4} \frac{1}{4^{(k-4)}} * \rho^k}$ (where $\rho = \frac{\lambda}{\mu}$)

d). $n = 2$:

i). The probability that an arriving query will be rejected.

Probability that an arrival is blocked = Probability that there are **m** customers in the system

λ = 15 queries/hour
μ = 3 queries/hour
$\rho = \frac{\lambda}{\mu} = \frac{15}{3} = 5$

Since there are 4 operators and 2 slots of queue, the total waiting list will be 6 in this case:
$P_{(0)} * \left(1 + \rho + \frac{1}{2} * \rho^2 + \frac{1}{6} * \rho^3 + \frac{1}{24} * \rho^4 + \frac{1}{24 * 4} * \rho^5 + \frac{1}{24 * 4 * 4} * \rho^6\right) = 1$
$P_{(0)} = 0.007124$
$P_{(6)} = P_{(0)} * \rho^6 * \frac{1}{24 * 4 * 4} = 0.2935$

As a result, the probability that a query will be rejected:

$\boldsymbol{x = 0.2935}$

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

mean waiting time = mean number of jobs in the queue / throughput
In steady state the throughput will be: λ = 15 queries/hour
mean number of jobs in the queue: 0*($P_{(0)}$ + $P_{(1)}$ + $P_{(2)}$ + $P_{(3)}$ + $P_{(4)}$) + 1* $P_{(5)}$ + 2* $P_{(6)}$

From i):
$P_{(5)} = P_{(0)} * \rho^5 * \frac{1}{24 * 4} = 0.2348$

As a result, avg. # jobs in queue = 0 + 0.2348*1 + 0.2935*2 = 0.8218
mean waiting time = 0.8218/15 = 0.05478 hour = 3.287 mins ~= 3 mins

e). Determine the blocking probability if you add 5, 10, 15 and 20 waiting slots.

Add 5 slots:

4 operators, 7 slots of queue.

$$P_{(0)} * \left(1 + \rho + \frac{1}{2} * \rho^2 + \frac{1}{6} * \rho^3 + \frac{1}{24} * \rho^4 + \frac{1}{24 * 4} * \rho^5 + \frac{1}{24 * 4 * 4} * \rho^6 + \cdots + \frac{1}{24 * 4^7} * \rho^{11}\right) = 1$$

$$P_{(0)} = 0.001798$$

$$P_{(11)} = P_{(0)} * \rho^{11} * \frac{1}{24 * 4^7} = 0.2935$$

Add 10 slots:

4 operators, 12 slots of queue.

$$P_{(0)} * \left(1 + \rho + \frac{1}{2} * \rho^2 + \frac{1}{6} * \rho^3 + \frac{1}{24} * \rho^4 + \frac{1}{24 * 4} * \rho^5 + \frac{1}{24 * 4 * 4} * \rho^6 + \cdots + \frac{1}{24 * 4^{12}} * \rho^{16}\right) = 1$$

$$P_{(0)} = 0.0005465$$

$$P_{(16)} = P_{(0)} * \rho^{16} * \frac{1}{24 * 4^{12}} = 0.2071$$

Add 15 slots:

4 operators, 17 slots of queue.

$$P_{(0)} * \left(1 + \rho + \frac{1}{2} * \rho^2 + \frac{1}{6} * \rho^3 + \frac{1}{24} * \rho^4 + \frac{1}{24 * 4} * \rho^5 + \frac{1}{24 * 4 * 4} * \rho^6 + \cdots + \frac{1}{24 * 4^{17}} * \rho^{21}\right) = 1$$

$$P_{(0)} = 0.0001749$$

$$P_{(21)} = P_{(0)} * \rho^{21} * \frac{1}{24 * 4^{17}} = 0.2023$$

Add 20 slots:

4 operators, 22 slots of queue.

$$P_{(0)} * \left(1 + \rho + \frac{1}{2} * \rho^2 + \frac{1}{6} * \rho^3 + \frac{1}{24} * \rho^4 + \frac{1}{24 * 4} * \rho^5 + \frac{1}{24 * 4 * 4} * \rho^6 + \cdots + \frac{1}{24 * 4^{22}} * \rho^{26}\right) = 1$$

$$P_{(0)} = 0.00005688$$

$$P_{(26)} = P_{(0)} * \rho^{26} * \frac{1}{24 * 4^{22}} = 0.2007$$

e). Why there is little drop in blocking probability after adding 10 waiting slots.

The blocking probability is the probability that there are **m** customers in the system.

$$P_{(drop)} = P_{(0)} * \rho^n * \frac{1}{24 * 4^{n-4}}$$

$$P_{(0)} = \frac{1}{\sum_{n=0}^{4} \frac{1}{n!} * \rho^n + \frac{1}{24} * \sum_{5}^{n} \frac{1}{4^{(n-4)}} * \rho^n}$$

As a result:

$$P_{(drop)} = \frac{\rho^n * \frac{1}{24 * 4^{n-4}}}{\sum_{n=0}^{4} \frac{1}{n!} * \rho^n + \frac{1}{24} * \sum_{5}^{n} \frac{1}{4^{(n-4)}} * \rho^n}$$

Where ρ = 5

$$P_{(drop)} = \frac{\rho^n * \frac{1}{24 * 4^{n-4}}}{65.375 + \frac{1}{24} * \sum_{k=5}^{k=n} \frac{1}{4^{(k-4)}} * \rho^k} = \frac{5^n}{4^{n-4} * \left(1569 + \frac{5^5}{4} \sum_{k=0}^{n-5} (5/4)^k\right)}$$

From the summation of a Geometric sequence:

$$P_{(drop)} = \frac{\rho^n * \frac{1}{24 * 4^{n-4}}}{65.375 + \frac{1}{24} * \sum_{k=5}^{k=n} \frac{1}{4^{(k-4)}} * \rho^k} = \frac{5^n}{4^{n-4} * \left(1569 + \frac{5^5}{4} \sum_{k=0}^{n-5} (5/4)^k\right)}$$

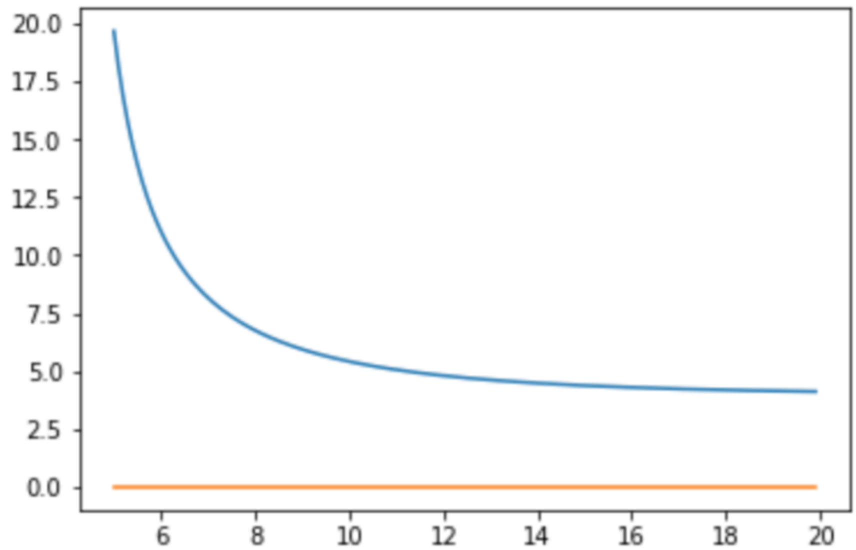
$$P_{(drop)} = \frac{1}{\frac{1}{4} - 1556 * \frac{4^{n-4}}{5^n}} = \frac{1}{0.25 - 0.6078 * 0.8^n}$$

Using the python note book to plot the graph:

```
import math
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline


def f(x):
    return 1/(0.25 - 0.6078*0.8**x)

xvals = np.arange(5, 20, 0.1)
yvals = np.array([f(x) for x in xvals])
plt.plot(xvals, yvals)
plt.plot(xvals, 0*xvals)
plt.show()
```



From the above, we can see that the probabilities of blocking drops smoothly after n =10, which means the additional slots is 10.

As a result, to reduce the blocking probability efficiently, 10 more slots can be added to the system. And the new system will be: M/M/4/12

Question3:

a). formulate a continuous-time Markov chian for this computer system.

Since there are four jobs in the system, the total number of states will be $\frac{(4+1)(4+2)}{2} = \mathbf{15}$

They are

(4,0,0)

(3,1,0) (3,0,1)

(2,2,0) (2,0,2) (2,1,1)

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

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

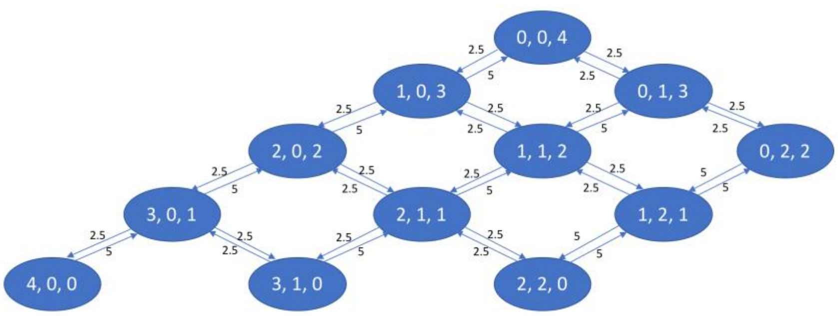
A job for disk will go to CPU1 directly, if there are 2 or more jobs in CPU2. This means the maximum number of jobs in CPU2 is 2. As a result, the cases (0,3,1), (1,3,0) and (0,4,0) are invalid.

So, there are only **12** valid cases, which are (4,0,0) (3,1,0) (3,0,1) (2,2,0) (2,0,2) (2,1,1) (1,2,1) (1,1,2) (1,0,3) (0,2,2) (0,1,3) (0,0,4) respectively.

Jobs complete at disk at a rate of 5 transactions/s

Jobs complete at CPU1 at a rate of 5 transactions/s

Jobs complete at CPU2 at a rate of 2.5 transactions/s



b). Balance equations for the continuous-time Markov Chain

$5*P_{(0,0,4)} = 5*P_{(1,0,3)} + 2.5*P_{(0,1,3)}$

$10*P_{(1,0,3)} = 2.5*P_{(0,0,4)} + 5*P_{(2,0,2)} + 2.5*P_{(1,1,2)}$

$7.5*P_{(0,1,3)} = 2.5*P_{(0,0,4)} + 5*P_{(1,1,2)} + 2.5*P_{(0,2,2)}$

$10*P_{(2,0,2)} = 2.5*P_{(1,0,3)} + 5*P_{(3,0,1)} + 2.5*P_{(2,1,1)}$

$12.5*P_{(1,1,2)} = 2.5*P_{(1,0,3)} + 2.5*P_{(0,1,3)} + 5*P_{(2,1,1)} + 2.5*P_{(1,2,1)}$

$7.5*P_{(0,2,2)} = 2.5*P_{(1,0,3)} + 5*P_{(1,2,1)}$

$10*P_{(3,0,1)} = 2.5*P_{(2,0,2)} + 5*P_{(4,0,0)} + 2.5*P_{(3,1,0)}$

$12.5*P_{(2,1,1)} = 2.5*P_{(2,0,2)} + 2.5*P_{(1,1,2)} + 5*P_{(3,1,0)} + 2.5*P_{(2,2,0)}$

$12.5*P_{(1,2,1)} = 2.5*P_{(1,1,2)} + 5*P_{(0,2,2)} + 5*P_{(2,2,0)}$

$5*P_{(4,0,0)} = 2.5*P_{(3,0,1)}$

$7.5*P_{(3,1,0)} = 2.5*P_{(3,0,1)} + 2.5*P_{(2,1,1)}$

$7.5*P_{(2,2,1)} = 2.5*P_{(2,1,1)} + 5*P_{(1,2,1)}$

(where $\sum P_{(CPU1,CPU2,Disk)} = 1$)

c). Steady state probabilities for each state.

The following is the code of matlab to get the probability of each state.

```
A = [ 5    -5  -2.5   0   0   0   0   0   0   0   0   0
      -2.5  10   0   -5  -2.5   0   0   0   0   0   0   0
      -2.5   0   7.5 0   -5  -2.5   0   0   0   0   0   0
      0  -2.5    0   10  0   0   -5  -2.5   0   0   0   0
      0  -2.5    -2.5    0   12.5    0   0   -5  -2.5    0   0   0
      0  0   -2.5    0   0   7.5 0   0   -5  0   0   0
      0  0   0   -2.5    0   0   10  0   0   -5  -2.5    0
      0  0   0   -2.5    -2.5    0   0   12.5    0   0   -5  -2.5
      0  0   0   0   -2.5    -5  0   0   12.5    0   0   -5
      0  0   0   0   0   0   -2.5    0   0   5   0   0
      0  0   0   0   0   0   -2.5    -2.5    0   0   7.5 0
      1  1   1   1   1   1   1   1   1   1   1   1 ];
b = [0 0 0 0 0 0 0 0 0 0 0 1]';
x = A\b;
```

mand Window

```
0.1711
0.0912
0.1598
0.0501
0.0935
0.1213
0.0259
0.0572
0.1021
0.0130
0.0277
0.0871
```

As a result:

$P_{(0,0,4)}$ = 0.1711
$P_{(1,0,3)}$ = 0.0912
$P_{(0,1,3)}$ = 0.1598
$P_{(2,0,2)}$ = 0.0501
$P_{(1,1,2)}$ = 0.0935
$P_{(0,2,2)}$ = 0.1213
$P_{(3,0,1)}$ = 0.0259
$P_{(2,1,1)}$ = 0.0572
$P_{(1,2,1)}$ = 0.1021
$P_{(4,0,0)}$ = 0.0130
$P_{(3,1,0)}$ = 0.0277
$P_{(2,2,1)}$ = 0.0871

d). throughput of the system.

Throughput of the system = CPU throughput
Throughput = Utilisation * Service rate
In this case, it is the sum of throughput of both CPU1 and CPU2
For CPU1:

$(P_{(1,0,3)} + P_{(2,0,2)} + P_{(1,1,2)} + P_{(3,0,1)} + P_{(2,1,1)} + P_{(1,2,1)} + P_{(4,0,0)} + P_{(3,1,0)} + P_{(2,2,1)})$*5jobs/s
= (0.0912+0.0501+0.0935+0.0259+0.0572+0.1021+0.0130+0.0277+0.0871) *5jobs/s
=2.739 jobs/s

For CPU2:

$(P_{(0,1,3)} + P_{(1,1,2)} + P_{(0,2,2)} + P_{(2,1,1)} + P_{(1,2,1)} + P_{(3,1,0)} + P_{(2,2,1)})$*2.5jobs/s
= (0.1598+0.0935+0.1213+0.0572+0.1021+0.0277+0.0871) *2.5jobs/s
= 1.622 jobs/s

Total throughput = Th.$_{\mathrm{CPU1}}$+ Th.$_{\mathrm{CPU2}}$ = 4.361 jobs/s

e). Mean number of jobs in CPU1.

$P_{(0,0,4)}$ = 0.1711
$P_{(1,0,3)}$ = 0.0912
$P_{(0,1,3)}$ = 0.1598
$P_{(2,0,2)}$ = 0.0501
$P_{(1,1,2)}$ = 0.0935
$P_{(0,2,2)}$ = 0.1213
$P_{(3,0,1)}$ = 0.0259
$P_{(2,1,1)}$ = 0.0572
$P_{(1,2,1)}$ = 0.1021
$P_{(4,0,0)}$ = 0.0130
$P_{(3,1,0)}$ = 0.0277
$P_{(2,2,1)}$ = 0.0871

Mean number of jobs in CPU1 = $\sum_{k=0}^{4} k * P_k$(where k is the #jobs in CPU1)
=1*(0.0912+0.0935+0.1021) + 2*(0.0501+0.0572+0.0871) +3*(0.0259+0.0277) + 4*0.0130
= 0.8884 jobs

f). Mean response time of CPU1.

Mean response time = mean number of jobs / throughput of CPU1
= 0.8884/2.739s
= 0.3244s