Nan Teong z3386180

COMP9334 Assignment

Q1 (a) Service demand $D(j)$ ★Calculations in q1.py★

Visit ratio $V(j)$ ⇒ Forced Flow Law : $V(j) = \frac{X(j)}{X(0)}$ output rate of j; output rate of system

Computed from $X(j) = \frac{C(j)}{T}$ and $X(0) = \frac{C(0)}{T}$

Given : $D(j) = V(j).S(j)$ mean service time of j

Since $V(j) = \frac{X(j)}{X(0)}$:

$D(j) = \frac{X(j).S(j)}{X(0)}$ (and we know $U(j) = X(j).S(j)$)

$= \frac{U(j)}{X(0)}$ (Service Demand Law)

• Refer to q1.py for calculations:

Service demand of CPU $= \frac{U(CPU)}{X(0)} = 0.89$ sec

" Disk 1 $= \frac{U(D1)}{X(0)} = 0.75$ sec

" Disk 2 $= \frac{U(D2)}{X(0)} = 0.82$ sec

" Disk 3 $= \frac{U(D3)}{X(0)} = 1.02$ sec //

(b) According to Service Demand Law, $D(j) = \frac{U(j)}{X(0)}$

∴ Utilisation will increase with increasing throughput in system $X(0)$ and service demand of device $D(j)$.

Since Disk 3 is calculated to have the highest $D(j)$, it will have the highest utilisation for given system output rate as it will reach $U(j) = 1$ first, and will be the bottleneck of the whole system.

There are interactive users in the system, so there will be think time Z.

Little's Law:

$N =$ Avg Resp. Time R × System Throughput $X(0)$

$\geq \left(\sum_{i=1}^{k} D_i + Z\right) \times X(0)$

$\Rightarrow X(0) \leq \frac{N}{\sum_{i=1}^{k} D_i + Z}$

Previously, we have $X(0) \leq \frac{1}{\max D(j)}$ where bottleneck throughput limited by max. svc dd

where $X(0) \leq \min \frac{1}{D(j)}$

$$\therefore X(0) \leq \min \left[\underset{\substack{\uparrow \\ \text{Bound 1}}}{\frac{1}{\max D_i}} , \underset{\substack{\uparrow \\ \text{Bound 2}}}{\frac{N}{\sum_{i=1}^{k} D_i + Z}} \right]$$

$$\text{Bound 1} = \frac{1}{\max D_i} = \frac{1}{\text{Svc dd of Disk 3}} = 0.97$$

$$\text{Bound 2} = \frac{N}{\sum_{i=1}^{k} D_i + Z}$$

$$= \frac{40}{\sum_{i=1}^{k} D_i + 27} = 1.31$$

$$X(0) \leq \min [0.97, 1.31]$$

Asympt. bound $= 0.97$ jobs/sec //

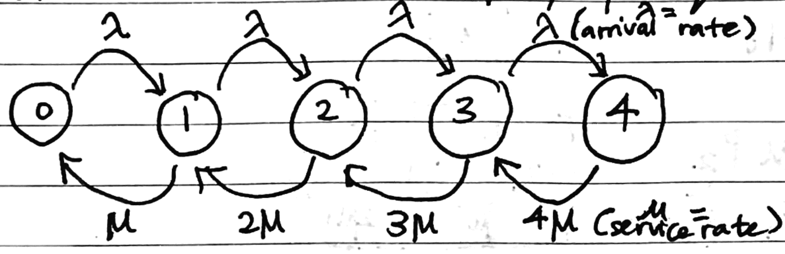
(c) No. of terminals = Max. system Throughput × (Think Time + Min. Response Time)

Min. Response Time $= \left(\frac{N}{\text{Asympt. Bound}} \right) -$ think time

$= 14.1$ sec //

Q2 (a) ★ Calculations in q2.py ★

State transition for M/M/m/m queue with no buffer (waiting m/



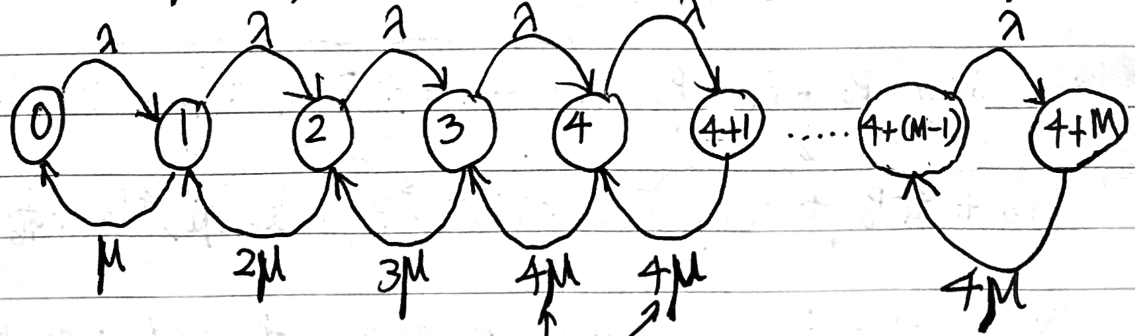
Erlang B Formula: where $m = 4$ & $\rho = \frac{\lambda}{\mu}$ (utilisation)

$$P_m = \frac{\frac{\rho^m}{m!}}{\sum_{k=0}^{m} \frac{\rho^k}{k!}} = \frac{\frac{\rho^4}{4!}}{\sum_{k=0}^{4} \frac{\rho^k}{k!}} \approx 0.24 \,//$$

(Prob. call is rejected = Prob. 4 customers in system)

(b)(i) Define states of queue:

- State 0 : 0 call in system (= operators idle)
- State 1 : 1 call in system (= 1 call @ operator, no call queue)
- State 2 : 2 calls in system (= 2 calls @ operators, no call queue)
- State 3 : 3 calls in system (= 3 calls @ operators, no call queue)
- State 4 : 4 calls in system (= 4 calls @ operators, no call queue)
- State (4+1) : (4+1) calls in system (= 4 calls @ operators, 1 call @ queue)
- State (4+M) = (4+M) calls in system (= 4 calls @ operators, M calls @ queue) ⇒ M calls @ queue as M is max. no. of holding slots @ queue, otherwise call is rejected if queue full



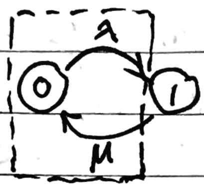
Max only 4 operators so stop @ 4μ for service rate

Max. no. of states = 4+M

(b) (ii) Balance equations:

- $\lambda P_0 = \mu P_1$

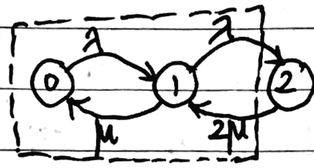
  $\Rightarrow P_1 = \frac{\lambda}{\mu} P_0$



- $\lambda P_1 = 2\mu P_2$

  $\Rightarrow P_2 = \frac{\lambda}{2\mu} P_1 = \frac{\lambda^2}{2\mu^2} \cdot P_0$

  $= \frac{1}{2}\left(\frac{\lambda}{\mu}\right)^2 P_0$

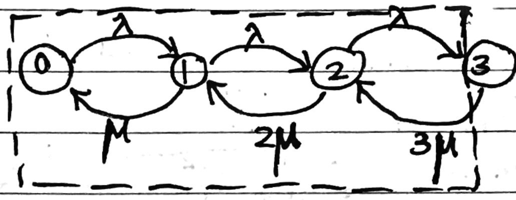


- $\lambda P_2 = 3\mu P_3$

  $\Rightarrow P_3 = \frac{\lambda}{3\mu} P_2$

  $= \frac{1}{3}\left(\frac{1}{2}\right)\left(\frac{\lambda}{\mu}\right)^3 P_0$

  $= \frac{1}{6}\left(\frac{\lambda}{\mu}\right)^3 P_0$

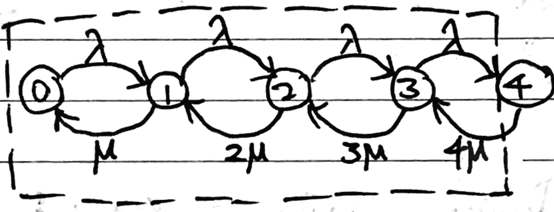


- $\lambda P_3 = 4\mu P_4$

  $\Rightarrow P_4 = \frac{\lambda}{4\mu} P_3$

  $= \frac{1}{4}\left(\frac{1}{6}\right)\left(\frac{\lambda}{\mu}\right)^4 P_0$

  $= \frac{1}{24}\left(\frac{\lambda}{\mu}\right)^4 P_0$

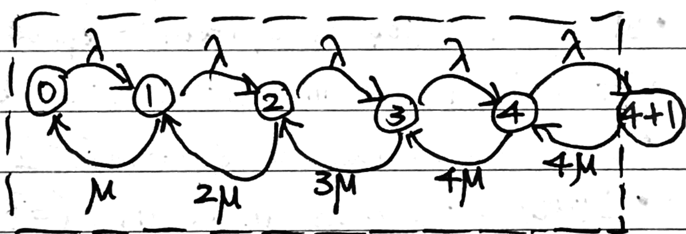


Start of diff. pattern →

- $\lambda P_4 = 4\mu P_{4+1}$

  $\Rightarrow P_{4+1} = \frac{\lambda}{4\mu} P_4$

  $= \frac{1}{4}\left(\frac{1}{24}\right)\left(\frac{\lambda}{\mu}\right)^{4+1} P_0$



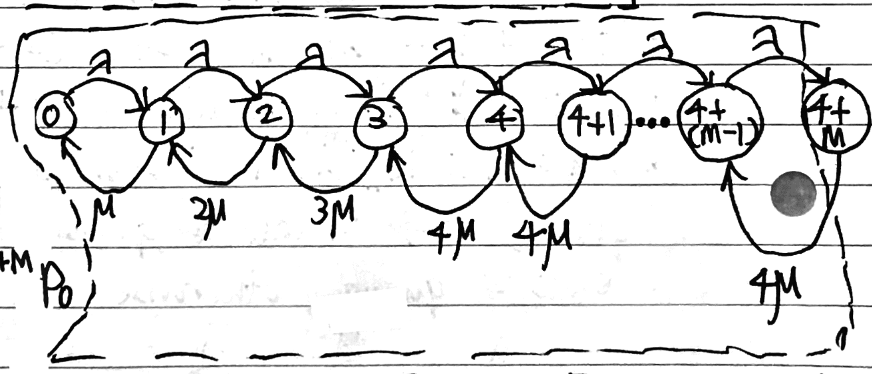
- $\lambda P_{4+1} = 4\mu P_{4+2}$

  $\Rightarrow P_{4+2} = \frac{\lambda}{4\mu} P_{4+1}$

  $= \left(\frac{1}{4}\right)^2\left(\frac{1}{24}\right)\left(\frac{\lambda}{\mu}\right)^{4+2} P_0$

- $\lambda P_{4+M-1} = 4\mu P_{4+M}$

  $\Rightarrow P_{4+M} = \left(\frac{1}{4}\right)^M\left(\frac{1}{24}\right)\left(\frac{\lambda}{\mu}\right)^{4+M} P_0$



(iii) Given $P_0 + P_1 + P_2 + P_3 + P_4 + P_{4+1} + P_{4+2} + \dots P_{4+M-1} = 1$

$$1 = P_0\left[1 + \frac{\lambda}{\mu} + \frac{1}{2}\left(\frac{\lambda}{\mu}\right)^2 + \frac{1}{6}\left(\frac{\lambda}{\mu}\right)^3 + \frac{1}{24}\sum_{i=0}^{M}\left(\frac{1}{4}\right)^i\left(\frac{\lambda}{\mu}\right)^{4+i}\right] //$$

(iv) Since P(rejected call) from (a) = 0.24 and we want it to be < 50% of 0.24:

$$P_{4+M} = \left(\frac{1}{4}\right)^M\left(\frac{1}{24}\right)\left(\frac{\lambda}{\mu}\right)^{4+M} P_0 < \frac{0.24}{2}$$

$$\left(\frac{1}{4}\right)^M\left(\frac{1}{24}\right)\left(\frac{\lambda}{\mu}\right)^{4+M} P_0 < 0.12$$

Solving for P_0 with smallest value of M,

$P_0 \approx 0.0312$ and $P_{4+M} = 0.0929$ and the smallest value of M required to reduce call rejection rate to less than 50% = 3 //

(v) • Rejected Rate = $P(M+4) = 0.0929$

• Throughput = Arrival Rate × (1 − Rejected Rate)

= 18.14 calls / hour

• Mean number of customers in queue

= Sum (State Probability × Calls # in State)

= 3.659

• Mean Response Time = $\frac{\text{Mean number of customers in queue}}{\text{Throughput}}$

= 0.2016 hour

• Waiting Time = Response Time − Service Time

$= 0.2016 - \frac{10}{60}$

= 0.035 hour (2.1 minutes) //

*----- : special case

*Circled: Based on state, where Disk sends job

CPU1 (faster)	CPU2
Busy	Idle
Idle	Idle
Idle	Busy
Busy	Busy

☆ Calculations in q3.py and q3.m ☆

Q3 (a) • CPU1 : 0.05 sec/job ⇒ 20 jobs/sec

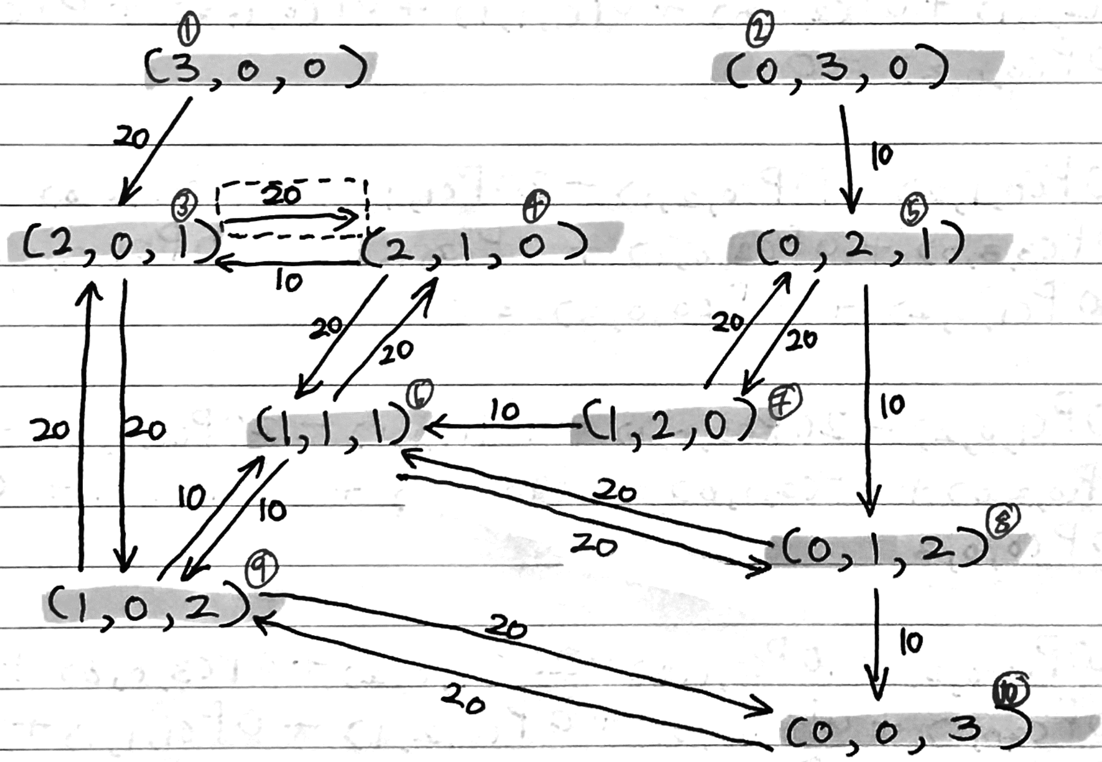
• CPU2 : 0.1 sec/job ⇒ 10 jobs/sec

• Disk : 0.05 sec/job ⇒ 20 jobs/sec

Number of states $= \frac{(n+1)(n+2)}{2} = 10$

(n = number of jobs)

Let (x, y, z) 3-tuple be (CPU1#, CPU2#, Disk#):



(b) Flow Balance Equations:

* Out arrows : (+)
* In arrows : (−)

① • $20P_{(3,0,0)} + 0P_{(0,3,0)} + 0P_{(2,0,1)} + 0P_{(2,1,0)} + 0P_{(0,2,1)} + 0P_{(1,1,1)} + 0P_{(1,2,0)} + 0P_{(0,1,2)} + 0P_{(1,0,2)} + 0P_{(0,0,3)} = 0$

② • $10P_{(0,3,0)} + 0P_{(3,0,0)} + 0P_{(2,0,1)} + 0P_{(2,1,0)} + 0P_{(0,2,1)} + 0P_{(1,1,1)} + 0P_{(1,2,0)} + 0P_{(0,1,2)} + 0P_{(1,0,2)} + 0P_{(0,0,3)} = 0$

③ • $40P_{(2,0,1)} - 20P_{(3,0,0)} - 10P_{(2,1,0)} - 20P_{(1,0,2)} + 0P_{(0,3,0)} + 0P_{(0,2,1)} + 0P_{(1,1,1)} + 0P_{(1,2,0)} + 0P_{(0,1,2)} + 0P_{(0,0,3)} = 0$

④ • $30P_{(2,1,0)} - 20P_{(2,0,1)} - 20P_{(1,1,1)} + 0P_{(3,0,0)} + 0P_{(0,3,0)} + 0P_{(0,2,1)} + 0P_{(1,2,0)} + 0P_{(0,1,2)} + 0P_{(1,0,2)} + 0P_{(0,0,3)} = 0$

5 • $30P_{(0,2,1)} - 10P_{(0,3,0)} - 20P_{(1,2,0)} + 0P_{(3,0,0)} +$
$0P_{(2,0,1)} + 0P_{(2,1,0)} + 0P_{(1,1,1)} + 0P_{(0,1,2)} + 0P_{(1,0,2)}$
$+ 0P_{(0,0,3)} = 0$

6 • $50P_{(1,1,1)} - 20P_{(2,1,0)} - 10P_{(1,2,0)} - 20P_{(0,1,2)} - 10P_{(1,0,2)}$
$+ 0P_{(0,3,0)} + 0P_{(2,0,1)} + 0P_{(0,2,1)} + 0P_{(3,0,0)} + 0P_{(0,0,3)} = 0$

7 • $30P_{(1,2,0)} - 20P_{(0,2,1)} + 0P_{(3,0,0)} + 0P_{(0,3,0)} +$
$0P_{(2,0,1)} + 0P_{(2,1,0)} + 0P_{(1,1,1)} + 0P_{(0,1,2)} + 0P_{(1,0,2)}$
$+ 0P_{(0,0,3)} = 0$

8 • $30P_{(0,1,2)} - 10P_{(0,2,1)} - 20P_{(1,1,1)} + 0P_{(3,0,0)}$
$+ 0P_{(0,3,0)} + 0P_{(2,0,1)} + 0P_{(2,1,0)} + 0P_{(1,2,0)}$
$+ 0P_{(1,0,2)} + 0P_{(0,0,3)} = 0$

9 • $50P_{(1,0,2)} - 20P_{(2,0,1)} - 10P_{(1,1,1)} - 20P_{(0,0,3)} + 0P_{(3,0,0)}$
$+ 0P_{(0,3,0)} + 0P_{(2,1,0)} + 0P_{(0,2,1)} + 0P_{(1,2,0)} + 0P_{(0,1,2)}$
$= 0$

10 • $20P_{(0,0,3)} - 10P_{(0,1,2)} - 20P_{(1,0,2)} + 0P_{(3,0,0)} + 0P_{(0,3,0)}$
$+ 0P_{(2,0,1)} + 0P_{(2,1,0)} + 0P_{(0,2,1)} + 0P_{(1,1,1)} +$
$0P_{(1,2,0)} + 0P_{(0,0,3)} = 0$

☆ Summary: Flow Balance Equations

1 • $20P_{(3,0,0)} = 0$
2 • $10P_{(0,3,0)} = 0$
3 • $40P_{(2,0,1)} - 20P_{(3,0,0)} - 10P_{(2,1,0)} - 20P_{(1,0,2)} = 0$
4 • $30P_{(2,1,0)} - 20P_{(2,0,1)} - 20P_{(1,1,1)} = 0$
5 • $30P_{(0,2,1)} - 10P_{(0,3,0)} - 20P_{(1,2,0)} = 0$
6 • $50P_{(1,1,1)} - 20P_{(2,1,0)} - 10P_{(1,2,0)} - 20P_{(0,1,2)} - 10P_{(1,0,2)} = 0$
7 • $30P_{(1,2,0)} - 20P_{(0,2,1)} = 0$
8 • $30P_{(0,1,2)} - 10P_{(0,2,1)} - 20P_{(1,1,1)} = 0$
9 • $50P_{(1,0,2)} - 20P_{(2,0,1)} - 10P_{(1,1,1)} - 20P_{(0,0,3)} = 0$
10 • $20P_{(0,0,3)} - 10P_{(0,1,2)} - 20P_{(1,0,2)} = 0$
• $Sum(P_{all}) = 1$

☆ Calculations in q3.m matlab file and q3.py ☆

(c) $P(3,0,0) = 0$

$P(0,3,0) = 0$

$P(2,0,1) = 0.1379$

$P(2,1,0) = 0.1954$

$P(0,2,1) = 0$

$P(1,1,1) = 0.1552$

$P(1,2,0) = 0$

$P(0,1,2) = 0.1034$

$P(1,0,2) = 0.1782$

$P(0,0,3) = 0.2299$ //

(d) Considering equilibrium, throughput of system = throughput of disk = sum of throughputs in CPUs

Disk util. where states with job @ disk

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

$= 0.1379 + 0 + 0.1552 + 0.1034 + 0.1782 + 0.2299$

The system throughput = disk throughput

= Disk utilisation × Service Rate of Disk

$= \text{Disk Utilisation} \times \frac{1}{\text{Service Time of Disk}}$

= 16.092 jobs/sec //

(e) CPU1 util. where states with job @ CPU1

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

$= 0 + 0.1379 + 0.1954 + 0.1552 + 0 + 0.1782$

$\text{CPU1 throughput} = \text{CPU1 Util} \times \frac{1}{\text{Service Time of CPU1}} = 13.334 \text{ jobs/sec}$

Little's Law : $R = N/X$

$\therefore$ Mean Response Time of CPU1 $= \frac{3}{13.334} = 0.2249$ sec //

(f) $\text{Disk Response Time} = \frac{\text{Jobs}}{\text{Disk Throughput}} = \frac{\text{Jobs}}{\text{System Throughput}}$

Wait Time = Disk Resp. Time − Service Time = 0.1364 sec //