Fiche Queuing

Pierre Colson

Contents

Markdown verion on *github*
Compiled using *pandoc* and *`gpdf` script*

M/M/1

- Offered load : $a = \lambda * \bar{x} = \frac{\lambda}{\mu}$
- utilization : $\rho = \frac{a}{m}$ in our case $m = 1$
- Stability condition : $\rho < 1$
- balance equation :

$$\lambda p_0 = \mu p_1$$

$$p_k = (\frac{\lambda}{\mu})^k p_0 = (1 - \rho)\rho^k$$

$$p_0 = 1 - \rho$$

- Averagage number of customer in the system : $N = \frac{\rho}{1-\rho}$
- At least n customers : $P(\geq n) = \rho^n$
- Little property :

$$N = \lambda T \implies T = \frac{1}{\mu - \lambda}$$

$$N_s = \lambda \bar{x}$$

$$N_q = \lambda W \implies W = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda}$$

- Pasta property hold
- System time distribution : $T \sim Exp(\mu - \lambda)$
- Waiting time distribution : $w(t) = 1 - \rho e^{-(\mu-\lambda)t}$

M/M/1/K

- Offered load : $\rho = \frac{\lambda}{\mu}$
- Effectiv load : $\rho_{eff} = \frac{\lambda_{eff}}{\mu} = \frac{(1-P(block))\lambda}{\mu}$
- Steady state :
  - $p_0 = \frac{1-\rho}{1-\rho^{K+1}}$
  - $p_k = \frac{(1-\rho)\rho^k}{1-\rho^{K+1}}$
- Blocking probability : $p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}$
- Effective traffic : $(1-p_K)\lambda$
- utilization : $\frac{\lambda_{eff}}{\mu}$
- $\bar{N} = \frac{\rho}{1-\rho}(1-(K+1)p_K)$

M/M/m/m - Erland loss system

- $a = \lambda \bar{x} = \frac{\lambda}{\mu}$
- μ_i for $i \leq m$ is equal to $i\mu$ and for $i > m$ is equal to $m\mu$
- Steady state :
  - $p_0 = \frac{1}{\sum_{k=0}^{m} \frac{a^k}{k!}}$
  - $p_k = \frac{\frac{a^k}{k!}}{\sum_{i=0}^{m} \frac{a^i}{i!}}$
- $N = N_s = \lambda_{eff}\bar{x} = (1-p_m)\lambda x = (1-p_m)a$
- $W = 0, \quad T = x, N_q = 0$
- $\rho = \frac{\lambda_{eff}\bar{x}}{m} = (1-p_m)\frac{a}{m}$
- $p_m = \frac{\frac{a^m}{m!}}{\sum \frac{a^i}{i!}} = E_m(a) = B(m,a)$: blocking probability Erlang B form

M/M/m - Erlang wait system

- Offered load : $a = \frac{\lambda}{\mu}$
- Server utilization : $\frac{a}{m}$
- In markov chain representation $\mu_k = k\mu$ (see lectures 6 notes)
- Steady state :
  - $k \leq m \implies p_k = \frac{a^k}{k!} p_0$
  - $k > m \implies p_k = \frac{a^k}{m^{k-m} m!} p_0$
  - $p_0 \left(\sum_{i=0}^{m-1} \frac{a^i}{i} + \frac{\frac{a^m}{m!}}{1 - \frac{a}{m}} \right) = 1$
- Probability that the arriving customer has to wait :

$$\frac{\frac{\frac{a^m}{m!}}{1-\frac{a}{m}}}{\sum_{i=0}^{m-1} \frac{a^i}{i!} + \frac{\frac{a^m}{m!}}{1-\frac{a}{m}}} = D_m(a)$$

No close form, we can use Erland table :

$$D_m(a) = \frac{mE_m(a)}{m - a\big(1 - E_m(a)\big)}$$

- $N_s = a$
- $N_q = D_m(a) \frac{a}{m-a}$
- Time between completed service : $Exp(m\mu)$
- $W(k) = 1 - D_m(a) e^{-(m\mu - \lambda)t}$
- $\mathcal{L}(f_w(t)) = \sum_{k=0}^{\infty} \mathcal{L}(f_w(t \mid k)) p_k$
  - $\mathcal{L}(f_w(t \mid k)) = \left(\frac{m\mu}{s + m\mu}\right)^{k-(m-1))} \quad k \geq m$
  - $\mathcal{L}(f_w(t \mid k)) = \int_0^{\infty} \delta(t) e^{-st} = 1 \quad k \leq m$

M/M/m/m/C - Engset loss System

- A customer does not generate a nex request while under service
- State probability in steady state :

$$p_k = \frac{\binom{C}{k} \left(\frac{\lambda}{\mu}\right)^k}{\sum_{i=0}^{\infty} \binom{C}{i} \left(\frac{\lambda}{\mu}\right)^i} = \binom{C}{k} \left(\frac{\lambda}{\mu}\right)^k p_0$$

- Probability that the arriving node finds ths system in state k : PASTA does not hold

$$a_k = \frac{\lambda_k p_k}{\sum_{i=0}^{m} \lambda_i p_i}$$

- Time blocking : part of the time the system is in blocking state : p_m
- Call blocking $P(\text{arriving request gets blocked}) = a_m$
- Offered traffic :

$$\lambda^* = \sum_{i=0}^{m}(C-i)\lambda p_i$$

- Effectiv traffic :

$$\lambda_{eff} = \sum_{i=0}^{m-1} \lambda_i p_i$$

- Average number of requests under service :

$$N = N_s = \frac{\lambda_{eff}}{\mu}$$

- We consider a system as finite population when $C < 10m$

Erlang-r server (E_r)

- For each exponential stage : $b(x_i) = r\mu e^{-r\mu x_i}$
- For each exponential stage : $C_x^2 = \frac{V[X_i]}{E[X_i]^2} = 1$
- For the service time : $b(x) = \frac{(r\mu)^r x^{r-1}}{(r-1)!} e^{-r\mu x}$
- For the service time : $C_x^2 = \frac{1}{r} < 1$
- System state : number of remaining service stages + r * number of waiting customers
- Number of customer in the system in state i : $N_i = \lceil \frac{i}{r} \rceil$
- Little and pasta hold

Hyper-exponential server (H_r)

- For each server : $b(x_i) = \mu_i e^{-\mu_i x}$
- For the system : $b(x) = \alpha_1 \mu_1 e^{-\mu_1 x} + ... + \alpha_R \mu_R e^{-\mu_R x}$
- Server i is chosen with probability α_i
- $C_x^2 = \frac{E[X^2]}{E[X]^2} - 1 \geq 1$

M/G/1

- Arrival process memoryless (Poisson(λ))
- Servcie time general, identical, idenpendant, f(x)
- Single server
- $\rho = \lambda E[x] < 1$ for stability
- Little : $N = \lambda T$
- Pasta holds
- Pollaczek-Khinchin mean formulas : see slide 10
- R_s is the average remaining service time : $R_s = \frac{\lambda}{2} E[X^2]$

- $W = \frac{R_s}{1-\rho} = \frac{\lambda E[X^2]}{2(1-\rho)} = \frac{\rho E[X]}{2(1-\rho)}(1 + C_x^2)$
- For M/M/1 : $C_x^2 = 1$, for M/D/1 : $C_x^2 = 0$, Hyper-Exp : $C_x^2 = 4$ and Erlang-4 : $C_x^2 = 1/4$

With vacation

- Waiting time : $W = \frac{\lambda E[X^2]}{2(1-\rho)} + \frac{E[V^2]}{2E[V]}$

With priority

Non-preemptive

- The service is completed even if higher priority customer arrives
- $W_i = \frac{R_s}{\left(1-\sum_{j=1}^{i-1} \rho_j\right)(\left(1-\sum_{j=1}^{i} \rho_j\right)}, \quad R_s = \frac{1}{2}\sum_{i=1}^{K} \lambda_i E[X_i^2]$
- $T_i = W_i + E[X_i]$
- Average waiting time : $W = \sum p_i W_i = \sum \frac{\lambda_i}{\lambda} W_i$

Preemptive

- The service is interrupted if higher priority customer arrives

Other

- kendall's notation A/S/m/c/p/O
  - A: Arrival process (distribution of interarrival times)
  - S: Distribution of the service time
  - m: number of servers
  - c: system capacity (buffer positions and server included)
  - p: population generating requests
  - O: order of service
- Inter arrival time or service time :
  - M Markovian (exponentially distributed)
  - D Deterministic (same know value)
  - E_r Erlang with r stages (sum of r exponentials)
  - H_k Hyper exponential with k branches (mix of k exponentials)
  - G General (btu known), some times GI for general independant
- random plitting of a poisson process result in independant Poisson process.
- Multiplex of mutiple Poisson processes is a poisson process
- λ : arrival intensity, average interarrival time : $\frac{1}{\lambda}$
- x_n : service time requirement of customer n, average x (or $\bar{x}$),
  - μ : service intensity, $\bar{x} = \frac{1}{\mu}$
- T_n : time customer n spend in the system (system time), average T,
  - W_n : Waiting time of csutomer n, average W,
  - relation : $T = W + x$
- $N(t)$: number of customer in the system at time t, average N,
  - $N_q(t)$: number of customer waiting at time t, average N_q,

- $N_s(t)$: number of customer in service at time t, average N_s
- relation : $N = N_s + N_q$

- $p_k(t)$: probability of k customers in the system at time t, stationary p_k
- Offered load : $a = \lambda \bar{x} = \frac{\lambda}{\mu}$ (arrival intensity * length of service)
  - Is expressed in Erlang (E) [no unit]
  - sometimes denoted by ρ
- Server utilization in system with infinite buffer capacity, m servers : $\rho = \frac{a}{m}$
- For system with blocking :
  - Effective traffic : λ_{eff}
  - Blocked traffic : $\lambda_b, \lambda_{eff} + \lambda_b = \lambda$
  - Effective load : $\lambda_{eff}\bar{x} = \frac{\lambda_{eff}}{\mu}$
  - server utilization : $\frac{\lambda_{eff}\bar{x}}{m} = \frac{\lambda_{eff}}{m\mu}$
- Little Result : $N = \lambda T$, Likewise : $N_q = \lambda W$ and $N_s = \lambda \bar{x}$
- p_k : P(system is in state k at time t)
- a_k : P(customer arriving at time t finds the system in state k) $= P($ the system is in stake | a customer arrives)
- PASTA property : $p_k = a_k$
- Stability condition : server utilization < 1
- P(next customer does not wait) $= P$(inter arrival time $>$ service time) (inter arrival time often $Exp(\lambda)$)
- Coefficient of variation : $C_x^2 = \frac{V[X]}{E[X]^2}$
- Randomly splitting of a Poisson process gives two independant Poisson processes.