SFWR ENG 4E03

Kemal Ahmed
Fall 2015
Instructor: Vincent Maccio

Note: material covered in Stats 3Y03 Summary will not be covered in this summary

Contents

Statistics

Poisson parameter [λ]: rate
Service rate [μ]:
Continuous Random Variable (CRV):

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Think chemistry, i.e. cancelling units

Variance

$$\operatorname{var}(X) = E\left[X^2\right] - \left(E[X]\right)^2$$

- <u>Don't change probability</u>, but square X for calculation only

$$\operatorname{var}(x) = E\left[(X-\mu)^2\right]$$

$$= \sigma^2$$

$$= \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,\mathrm{d}x$$

$$= \int x^2 f(x)\,\mathrm{d}x - \mu^2$$

Exponential

- **Mean** [E[X]]: 1/λ
  - a.k.a. Expected value
- **Variance**: $1/\lambda^2$
- **Probability Distribution Function (PDF)** [P(X=x)]: $\lambda e^{-\lambda x}/x!$
- **Cumulative Distribution Function (CDF)** [f(x)]: CDF = ∫PDF, i.e. $1 - e^{-\lambda x}$
- Memoryless
- not always for time

Uniform

- **Variance**: $(b-a)^2/12$
- **Mean**: (a+b)/2
- **PDF**: 1 / (b–a) , a ≤ x ≤ b
- **CDF**: 1
- **Uniform Distribution**: no memoryless property

Binomial

- **Mean** [E[X]]: n × probability
- **Variance**: n × p × (1 – p)
- **Probability Distribution Function (PDF)** [P(X)]: (n c x)$p^x(1-p)^{n-x}$
- **Cumulative Distribution Function (CDF)** [f(x)]: $\sum_{i=0}^{\lfloor x \rfloor} \binom{n}{i} p_i (1-p)^{n-i}$

Operations Analysis

Device [i]: units that are in terms of *i* are specific to an individual device or node within a system
Total devices [k]:

Service Time [S]: time per specific job
1/μ

Visitation [V]: given or projected visits/jobs (closed system); cannot be calculated; basically a probability
[E(V)]: calculated visit/job ratio
P(visit)·total visits in previous node

Demand [D]: total service time for all jobs

$$D_i = E[S_i] \cdot V_i$$

$$D = \sum_{i=0}^{k} D_i$$

Bottleneck [D_{max}]: device with largest demand, utilization

Time in system [T]: time the job is in the system

$$E[T] = \frac{N}{X}$$

$$E[T] \geq \max\left(D, ND_{\max} - E[Z]\right)$$

If E[Z] = 0, T = R

Response Time [R]: time the job is *being processed* in the system
If E[Z] = 0, R = T

M/M/1: E[R] = 1/(μ–λ)
M/M/C: E[R] = E[R_Q] + E[S]

Users [M]:
Optimal users [M*]:

$$M* = \frac{D + E[Z]}{D_{\text{bottleneck}}}$$

Total Jobs [N]: N=M in a closed system

- *Little's Law*: $E[N] = \lambda E[T], \lambda = X$
- $E[N] = \lambda E[R], \lambda = X$
- Steady state probability
  - M/M/1:
    - E[N] = λ/(μ–λ) = $\rho/(1-\rho)$, if you have overall system λ
    - E[N] = $\sum_{i=0}^{\infty} i\pi_i$ ← probability × #jobs, if your λ or μ is different for each state
  - M/M/C: E[N] = ΣE[N_i] = ΣPλ/(μ_i–Pλ) = Σ(λ_i/(μ_i–λ_i))

Think time [Z]: time it takes the user to put a request in and start, it's kinda like the frequency that users put in requests (seconds / request)

$$E[Z] = E[T] - E[R]$$

Throughput [X]: out-rate, <u>max</u> jobs / hour of full system

$$X \leq \min\left(\frac{1}{D_{\max}}, \frac{N}{D + E[Z]}\right)$$

Note: $\frac{1}{D_{\max}}$ and $\frac{N}{D+E[Z]}$ converge at their lowest point, so equate them

$X_i = E[V_i]X$

Utilization [ρ]: ratio that the time is busy

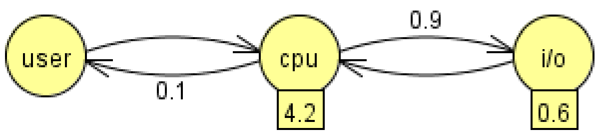
$\rho_i = X_i E[S_i]$

$\rho_i = XD_i$

$\rho = \lambda / c_i \mu$

Visitation Trick

If determining visitation at a node, establish a reference node from one of the incoming nodes, usually the user node, that has a returning percentage



$V_{user} = 1 = 0.1 \cdot V_{CPU}$

Summation Equations

Geometric Series: $\sum_{i=0}^{\infty} r^i = \frac{1}{1-r}$, where 0 ≤ r ≤ 1 (because otherwise it would be unstable)

$$\sum_{i=0}^{\infty} r^{i+1} = \sum_{i=1}^{\infty} r^i = \frac{r}{1-r}$$

$$\sum_{i=1}^{\infty} r^i = \sum_{i=0}^{\infty} r^i - 1$$

Geometric Sequence: $S_n = \sum_{i=0}^{n} r^i = \frac{1-r^{n+1}}{1-r}$

$$S_n = \sum_{i=1}^{n} r^i = \frac{r(1-r^n)}{1-r}$$

Removing the annoying factors:

$$\sum_{i=0}^{\infty} i(i+1)\rho^i$$

Take out a value so the integral takes out the i and i+1

$$= \rho \sum_{i=0}^{\infty} i(i+1)\rho^{i-1}$$

$$= \rho \frac{\mathrm{d}\rho}{\mathrm{d}i}\left(\sum_{i=0}^{\infty}(i+1)\rho^{i}\right)$$

$$= \rho \frac{\mathrm{d}\rho^2}{\mathrm{d}^2 i}\left(\sum_{i=0}^{\infty}\rho^{i+1}\right)$$

$$= \rho \frac{\mathrm{d}\rho^2}{\mathrm{d}^2 i}\left(\sum_{i=1}^{\infty}\rho^{i}\right)$$

$$= \rho \frac{\mathrm{d}\rho^2}{\mathrm{d}^2 i}\left(\sum_{i=0}^{\infty}\rho^{i} - \rho^0\right)$$

$$= \rho \frac{\mathrm{d}\rho^2}{\mathrm{d}^2 i}\left(\frac{1}{1-\rho} - 1\right)$$

DTMC

Discrete Time Markov Chains (DTMC):

[n]: number of tasks in queue / system

$$\pi_n = \frac{\prod_{i=0}^{n}\lambda_i}{\prod_{i=0}^{n}\mu_i}\pi_0$$

^think of it like series / parallel, where you add multiple connections out in different directions (parallel) and multiply connections stacked onto each other (series)

OR $\text{jobs}_{in} = \text{jobs}_{out}$

Steady state: n->∞

For discrete: use the sum of the X's, so $E[X] = \Sigma(P(X=i)\cdot X_i)$ and $E[X^2] = \Sigma(P(X=i)\cdot X_i^2)$

Matrices

Rows: equations for nodes going out (add up to 1)

Columns: equations for nodes coming in

CTMC

Poisson Process

Counting Process: a way of determining the time between consecutive occurrences of an event

Poisson Process: a *counting process*, whose time between arrivals uses Exponential Distribution

- $\lambda_{total} = \Sigma\lambda_i$
  - you can also split up λ into multiple λs

- Not only do you see each second as time independent, each stream of probabilities is independent
- $P(x;\lambda) = e^{-\lambda}\lambda^x/x!$
    - [x]: things will happen
    - [λ]: rate; $\lambda = \alpha t$
- [α]: expected number of events during unit interval
- [t]: time interval length
- $P_X(t) = \dfrac{e^{-\alpha t} \cdot (\alpha t)^X}{X!}$

Kendall notation

Job Processing time [μ]: rate of jobs leaving system (jobs/sec)
$\mu = 1/$ processing_time_per_job

M/M/1 Queue
[M]: time between arrivals is Markovian (Memoryless) ~ exp(λ)
[M]: job processing times are Markovian (Memoryless) ~ exp(μ)
[1]: single server

$(\Sigma p_{out}) \times \pi_i = \Sigma p_j \pi_j$, j=0..n, j≠i

π_0: percent of time that the queue is empty

Attributes:

- FIFO
- Infinite buffer

Variations

- M/M/2 Queue: same, except 2 servers
- M/M/C Queue: *C* servers
- $M/E_k/C$: Erlang k, i.e. series of exponential
- H()/M/C: hyperexpontial distribution
- PH/M/C: phase type, i.e. any combination of any number of exponentials with any rate
- M/G/C: General distribution
- G/G/1: has not been solved yet
- M/M/1/1: 1 server, 1 job

[c]: number of servers

Think: one queue goes to multiple servers

$$\xrightarrow{\lambda}] \begin{array}{l} \nearrow M_1 \\ \searrow M_2 \end{array}$$

$\pi_0 = 1 - \lambda/\mu$

e.g.

When you have varying

$$\pi_n = (n+1)\left(\frac{\lambda}{\mu}\right)^n \pi_0$$

$$1 = \sum_{i=0}^{\infty}(i+1)\rho^i \pi_0$$

$$= \pi_0 \sum_{i=1}^{\infty}(i+1)\rho^i$$

$$= \pi_0 \frac{\mathrm{d}}{\mathrm{d}\rho}\left(\sum_{i=0}^{\infty}\rho^{i+1}\right)$$

$$= \pi_0 \frac{\mathrm{d}}{\mathrm{d}\rho}\left(\sum_{i=1}^{\infty}\rho^{i}\right)$$

$$= \pi_0 \frac{\mathrm{d}}{\mathrm{d}\rho}\left(\sum_{i=0}^{\infty}\rho^{i+1}\right)$$

M/M/C Provisioning

Useful if multiple jobs are sharing the same queue

$$\pi_0 = \left[1 + \sum_{i=1}^{c}\frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i \left(\frac{1}{1-\frac{\lambda}{c\mu}}\right)\right]^{-1}$$

$$\pi_i = \begin{Bmatrix} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n \pi_0, \quad n < c \\ \frac{1}{c!c^{n-c}}\left(\frac{\lambda}{\mu}\right)^n \pi_0, n \ge c \end{Bmatrix}$$

Blocking Probability [P_Q]: probability that a process will be blocked when entering the system and be placed in the queue

Erlang-C Equation: $P_Q = \sum_{i=0}^{\infty}\pi_i = \frac{1}{c!}\left(\frac{\lambda}{\mu}\right)^c\left(\frac{1}{1-\rho}\right)\pi_0$

Given λ and μ, what should c be so $P_Q < \rho$

Waiting time in queue [R_Q]: response time of queue

$$E\left[R_Q\right] = \frac{1}{\lambda}P_Q\left(\frac{\rho}{1-\rho}\right)$$

M/M/1: $E\left[R_Q\right] = \frac{1}{\mu-\lambda} - \frac{1}{\mu}$

M/M/C: $E\left[R_Q\right]=\left(\frac{(\lambda/\mu)^c \mu}{(c-1)!(c\mu-\lambda)^2}\right)\pi_0$

[Q]: transition matrix

$$q_{ii} = -\sum_{j=i} q_{ij}$$

$$q_{ij} = \lim_{\Delta t \to 0} \frac{P\{X_{t+\Delta t} = j \mid X_t = i\}}{\Delta t}$$

Replace i <--> j to get q_{jj} and q_{ji}.

Jackson Networks

$P(N_1 = n_1) =$

$$\pi_{\vec{n}} = P_1^{n_1}(1-\rho_1)\rho_2^{n_2}(1-\rho_2)\cdots\rho_k^{n_k}(1-\rho_k)$$

$\pi_{\mathbf{n}\sim}$ = P(state of system n~) $= \prod_{i=1}^{k}$ P(n jobs at node i)

$$= \prod_{i=1}^{k} \rho_i^{n_i}(1-\rho_i)$$

Traffic Equations

For each node, what is the number of jobs entering?

$\lambda_x = R + \Sigma P_{i,entering} \cdot \lambda_{i,entering}$

response rate + probability of each job entering

Questions

- Assignment 5, Q2 states in ready queue??
- Assignment 6, Q3, M/M/C