Computer Systems Modelling

R. J. Gibbens
Computer Laboratory

Computer Science Tripos, Part II
University of Cambridge

Michaelmas Term 2002

Twelve lectures covering:

- *Introduction to modelling*: what is it, why is it useful?
- *Simulation techniques*: random number generation, analysis of results from simulation and measurements;
- *Operational analysis*: performance bounds, balanced systems;
- *Queueing theory*: Markov chains, single/multiple servers, bounded queues, queueing networks.

Recommended books

- Jain, A.R. *The Art of Computer Systems Performance Analysis*, Wiley, 1991

  — Simulation, random number generation, operational analysis, some basic queueing theory, substantial sections on designing and analysing experiments

- Kleinrock, L. *Queueing Systems — Volume 1: Theory*, Wiley, 1975

  — A classic on queueing theory, much more emphasis on mathematical derivations

Recommended books (2)

- Leung, C. *Quantitative Analysis of Computer Systems*, Wiley, 1988

  — Operational analysis, basic queueing theory, small section on simulation

- Ross, S.M. *Probability Models for Computer Science*, Academic Press, 2002

  — Excellent coverage of most material (but no Operational Analysis)

Introduction to modelling

Why model?

- A manufacturer may have a range of compatible systems with different characteristics — which configuration would be best for a particular application?
- A system is performing poorly — what should be done to improve it? Which problems should be tackled first?
- Fundamental design decisions may affect performance of a system. A model can be used as part of the design process to avoid bad decisions and to help quantify a cost/benefit analysis.

A toy problem

system	CPU time	disk time	total
A	4.6	4.0	8.6
B1	5.1	1.9	7.0
B2	3.1	1.9	5.0

- A database running on Type *A* system is too slow
- Type *B1* system available immediately, and a type *B2* system in the future
- Which option is best:
  - Stick with *A*?
  - Change to *B1* immediately?
  - Wait and change to *B2*?
- What factors affect the correct choice?

How can modelling help?

Typical performance questions we might ask include:

- How long will a database request wait before receiving CPU service?
- What is the utilization of the resource (CPU, disk, ...)? (Utilization is the proportion of time that the resource is busy.)
- What is the distribution of the number of requests queued at some time t? What is its mean, standard deviation, ...

Techniques

Many different approaches:

- *Measurement* — if the system already exists then maybe it can be changed and the effects analysed
- *Simulation* — construct a computer program that emulates the system under study and study that instead
- *Operational analysis* — analysis based on directly measured quantities and relationships between them: makes few assumptions about the system
- *Queueing theory* — stochastic processes, analytical models of queueing systems

Techniques (2)

Choice of technique depends on ...

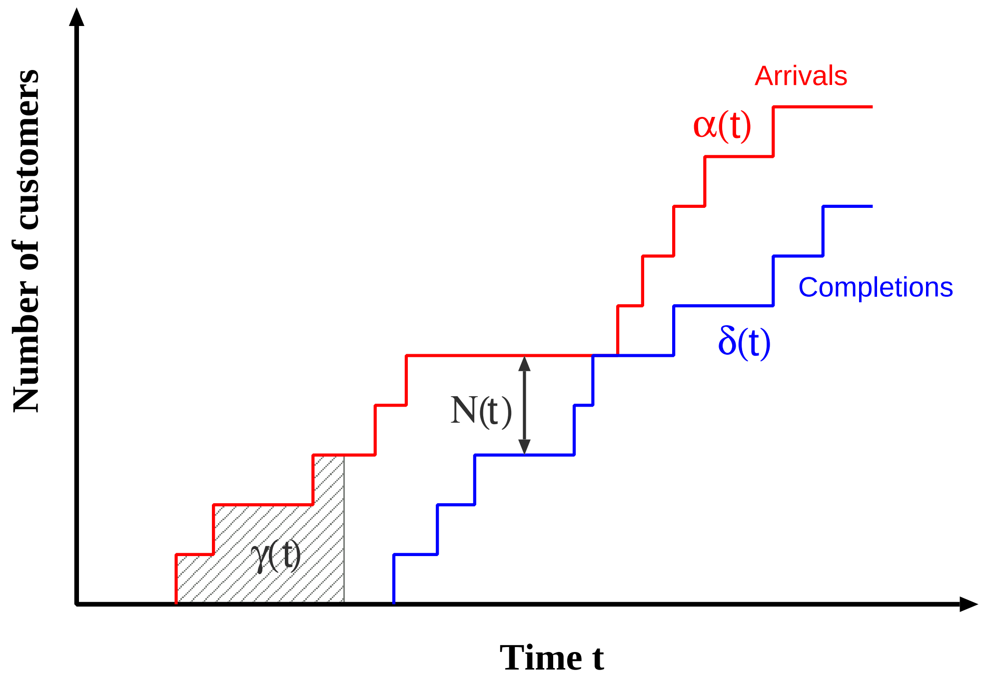
- *Stage of development*: can only measure an existing system
- *Time available*: measurements or simulations can take a long time to complete. How easily can different trade-offs be evaluated?
- *Resources*: systems with which to experiment, people with the relevant skills, cost
- *Desired accuracy*: how do the assumptions made in analytic techniques effect the result? Are appropriate parameters and workloads used during experimental work?
- *Creditable*: will people believe (act on) the results?

Little's result

Begin with a simple derivation of *Little's Result* — a very general theorem relating the number of jobs in a system with the time they spend there.

> *For example*: A disk server takes, on average, 10ms to satisfy an I/O request. If the request rate is 100 per second, then how many requests are queued at the server?

Little's result (2)



$\alpha(t)$ = number of arrivals in $(0, t)$

$\delta(t)$ = number of departures in $(0, t)$

$N(t) = \alpha(t) - \delta(t)$ is the number in the system at t

The area $\gamma(t)$ between the curves $\alpha(t)$ and $\delta(t)$ represents the total time all customers have spent in system in $(0, t)$.

Little's result (3)

Let

$$\lambda_t = \alpha(t)/t$$
$$T_t = \gamma(t)/\alpha(t)$$
$$N_t = \gamma(t)/t\,.$$

- λ_t — the average arrival rate during $(0, t)$;
- T_t — system time per customer averaged over all customers in $(0, t)$;
- N_t — average number of customers in system during $(0, t)$.

Combining these:

$$N_t = \lambda_t T_t\,.$$

Little's result (4)

Assume the following limits exist

$$\lambda = \lim_{t\to\infty} \lambda_t$$
$$T = \lim_{t\to\infty} T_t \,.$$

Then we have

$$N = \lambda T \,.$$

That is, the average number in the system, N, equals the average arrival rate $\times$ average time in system.

This is *Little's result*. The proof makes no assumptions about the way that arrivals or departures are distributed, the queueing discipline or how many servers service the queue.

First proved by Little in 1961.

Applications of Little's result

We can re-state this for any boundary of our queueing system.

We can split T (average time in the system) into W (average time spent waiting) and X (average time spent being served).

Similarly, we can split N (average number in the system) into N_q (average number in the queue) and N_s (average number being served).

Applying Little's result separately to the queue and to the server:

$$N_q = \lambda W$$
$$N_s = \lambda X$$

Probability theory refresher

We write the probability that an event occurs as: $\mathbb{P}(\text{event})$

— e.g. $\mathbb{P}(X < 0.5)$: "the random variable X is less than 0.5"

$\mathbb{P}(a, b)$ is the *joint* probability that *both* events a and b occur

— e.g. $\mathbb{P}(X < 0.5, Y < 0.5)$: "both X and Y are less than 0.5"

$\mathbb{P}(a \mid b)$ is the *conditional* probability that event a occurs given that event b has occured.

— e.g. $\mathbb{P}(X < 0.5 \mid Y < 0.5)$ "X is less than 0.5 given that Y is less than 0.5"

$$\mathbb{P}(a \mid b) := \frac{\mathbb{P}(a, b)}{\mathbb{P}(b)}$$

Continuous distributions

We can describe the distribution of a *continuous* random variable, X, in two ways using:

- either the *cumulative distribution function* (cdf),

$$F_X(x) := \mathbb{P}(X \le x)$$

- or the *probability density function* (pdf)

$$f_X(x) := \frac{dF_X(x)}{dx}.$$

Note that

$$0 \le F_X(x) = \int_{-\infty}^{x} f_X(y)\,dy$$

and that $F_X(x)$ increases with x to

$$F_X(\infty) = \int_{-\infty}^{\infty} f_X(y)\,dy = 1.$$

Expected value and moments

The *expected value* of X, written $\mathbb{E}(X)$, is given by

$$\mathbb{E}(X) := \int_{-\infty}^{\infty} x f_X(x)\, dx\,.$$

Also called the average, mean or first moment of the distribution and sometimes written as $\overline{X}$.

The n^{th} *moment* is defined as

$$\mathbb{E}(X^n) := \int_{-\infty}^{\infty} x^n f_X(x)\, dx\,.$$

The n^{th} *central moment* is defined as

$$\mathbb{E}((X - \mathbb{E}(X))^n) := \int_{-\infty}^{\infty} (x - \mathbb{E}(X))^n f_X(x)\, dx\,.$$

Variance and standard deviation

The 2^{nd} central moment, σ_X^2, is known as the *variance*,

$$\sigma_X^2 := \mathbb{E}((X - \mathbb{E}(X))^2) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$$

which may also be written as $\mathrm{Var}(X)$.

From this, we define the *standard deviation* by

$$\sigma_X := \sqrt{\sigma_X^2}$$

and the *coefficient of variation* by

$$C_X := \frac{\sigma_X}{\mathbb{E}(X)} .$$

Numerically larger values of C_X signify "more variable" data. For example, a coefficient of variation of 5 might be considered large, while 0.2 might be considered small.

Uniform distribution $U(a, b)$

Given the two parameters a and b ($a < b$) the random variable X has the uniform distribution $U(a, b)$ defined by the cdf

$$F_X(x) = \begin{cases} 0 & \text{if } x < a \\ \dfrac{x-a}{b-a} & \text{if } a \leq x < b \\ 1 & \text{if } x \geq b \end{cases}$$

so that the pdf is

$$f_X(x) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise}\,. \end{cases}$$

Exercise: show that the mean of X is $\frac{1}{2}(a + b)$ and the variance is $\frac{1}{12}(b - a)^2$.

Exponential distribution $\mathrm{Exp}(\lambda)$

Given a scale parameter, $\lambda > 0$, the (positive) random variable X has the exponential distribution $\mathrm{Exp}(\lambda)$ defined by the pdf and cdf

$$f_X(x) = \lambda\, e^{-\lambda\, x}$$

$$F_X(x) = 1 - e^{-\lambda\, x}\,.$$

Exercise: show that the mean of X is $\frac{1}{\lambda}$ and the variance is $\frac{1}{\lambda^2}$.

Hence for this distribution the mean and standard deviation are equal and so

$$C_X = \frac{\sigma_X}{\mathbb{E}(X)} = 1\,.$$

Memoryless property

The exponential distribution is the only continuous distribution with the *Memoryless Property*, namely, that

$$\mathbb{P}(X > t + s \mid X > t) = \mathbb{P}(X > s)$$

Intuitively, it may be used to model the distribution of inter-event times in which the time until the next event does not depend on the time that has already elapsed.

If the inter-event times are independent identically distributed random variables with the $\mathrm{Exp}(\lambda)$ distribution then λ is viewed as the mean event rate.

Gamma distribution, $\Gamma(n, \lambda)$

A random variable, X, with pdf

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x} \frac{(\lambda x)^{n-1}}{(n-1)!} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $n = 1, 2, \ldots$ and $\lambda > 0$ has a Gamma distribution with parameters (n, λ). The cdf is can computed from

$$F_X(x) = \int_0^x f_X(y)\, dy\,.$$

The mean and variance of X can be shown to be n/λ and n/λ^2, respectively.

It can be shown that the sum of n independent $\text{Exp}(\lambda)$ random variables has a Gamma $\Gamma(n, \lambda)$ distribution — see more on this latter.

Normal distribution $N(\mu, \sigma^2)$

A random variable, X, has a normal distribution if its pdf is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{(x-\mu)^2/2\sigma^2}$$

where μ is any real number and $\sigma > 0$. Recall that the mean of X is μ and the variance is σ^2.

Thus the standardized random variable

$$Z = \frac{X - \mu}{\sigma}$$

has a normal distribution with mean 0 and variance 1. The cdf of Z is usually written

$$F_Z(x) = \Phi(x) .$$

Notice that then

$$F_X(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) .$$

Central Limit Theorem (CLT)

Suppose that $X_1, X_2, \ldots$ is a sequence of independent, identically distributed random variables (with finite mean μ and finite variance σ^2) then the CLT says that

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{X_1 + X_2 + \cdots + X_n - n\mu}{\sqrt{n}\sigma} < x\right) = \Phi(x)\,.$$

Notice that the mean and standard deviation of $(X_1 + X_2 + \cdots + X_n)$ are $n\mu$ and $\sqrt{n}\sigma$, respectively.

Furthermore, notice that the individual random variables $X_1, X_2, \ldots$ are not assumed to have normal distributions.

Discrete distributions

The previous examples have concerned *continuous* random variables whose distributions have been defined by their *cdf* or, equivalently, their *pdf*.

Similar definitions apply to the case of *discrete* random variables, X, taking values x_i $(i \in I)$, where the distribution is specified by the probability distribution function (pdf)

$$0 \leq \mathbb{P}(X = x_i) \leq 1 \qquad \forall i \in I$$

and where

$$\sum_{i \in I} \mathbb{P}(X = x_i) = 1 \,.$$

Expected value and variance

The *expected value* of X is

$$\mathbb{E}(X) := \sum_{i \in I} x_i \mathbb{P}(X = x_i) .$$

Similarly, for the other moments, where the integration for continuous random variables becomes a summation over the set of possible values.

So, for example, we have that

$$\mathbb{E}(X^2) = \sum_{i \in I} x_i^2 \mathbb{P}(X = x_i)$$

and

$$\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$$

just as with continuous random variables.

Bernoulli(p) distribution

The random variable X has a *Bernoulli distribution* if it takes two values $X = 0$ or $X = 1$ with probabilities

$$\mathbb{P}(X = x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases}$$

We say that $p = \mathbb{P}(X = 1)$ is the probability of success $(0 \leq p \leq 1)$.

The mean and variance of X are p and $p(1 - p)$, respectively.

Binomial(n, p) distribution

The (random) number of successes in a fixed length sequence of independent Bernoulli trials has a distribution known as the *Binomial distribution*. The pdf is given by

$$\mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x} \quad x = 0, 1, \ldots, n$$

where p is the probability of success of an individual Bernoulli trial and n is the fixed number of trials. Thus the parameters satisfy $0 \le p \le 1$ and $n = 1, 2, 3, \ldots$.

The mean and variance of X are np and $np(1-p)$, respectively.

Note that $\mathbb{P}(X = x)$ is the product of the number of ways that x successes can occur in n trials and the probability that exactly that pattern of successes and failures occurs.

Poisson(λ) distribution

The random variable X has a Poisson distribution if it takes values $0, 1, 2, \ldots$ with probabilities

$$\mathbb{P}(X = i) = e^{-\lambda} \frac{\lambda^i}{i!} \quad i = 0, 1, 2, \ldots$$

where $\lambda > 0$ is a scale parameter.

Exercise: show that both the mean and variance of X are equal to λ.

The Poisson(λ) distribution is a good approximation to a Binomial(n, p) distribution when the number of trials, n, is large and the probability of success, p, is small (and where we take $\lambda = np$).

Geometric(p) distribution

Given a sequence of independent Bernoulli trials, each with probability of success p how long do we wait till the first successful trial?

The number of trials X up to and including the first successful trial has a distribution called the Geometric distribution given by

$$\mathbb{P}(X = n) = p(1 - p)^{n-1} \quad n = 1, 2, \ldots .$$

The mean of X is given by $1/p$ and the variance by $(1 - p)/p^2$.

The Poisson process

Consider a process of events occuring at random points of time and let $N(t)$ be the number of events that occur in the interval $[0, t]$. A Poisson process at rate λ ($\lambda > 0$) is defined by the following conditions:

- $N(0) = 0$;
- The Nos of events in disjoint time intervals are independent and the distribution of the No of events in a interval depends only on its length (and not its location);
-

$$\mathbb{P}(N(h) = i) = \begin{cases} 1 - \lambda h + o(h) & i = 0 \\ \lambda h + o(h) & i = 1 \\ o(h) & \text{otherwise}\,. \end{cases}$$

A quantity $g(h) = o(h)$ if $\lim_{h \to 0} f(h) = 0$.

The Poisson process (2)

Consider the No of events, $N(t)$, occuring in an interval of length t. Divide the interval into n nonoverlapping subintervals each of length $h = t/n$.

A subinterval contains a single event with probability approximately $\lambda(t/n)$ and so it follows that the number of such subintervals is approximately a Binomial random variable with parameters n and $p = \lambda t/n$.

> Letting $n \to \infty$, shows that $N(t)$, the number of events in $[0, t]$, is a Poisson random variable with parameter λt.

The Poisson process (3)

Given a Poisson process of rate λ let X_1 be the time of the first event and for $n > 1$ let X_n denote the time between the $(n-1)st$ and nth events.

The sequence $X_1, X_2, \ldots$ gives us the sequence of inter-arrival times between events in a Poisson process.

The Poisson process (4)

To determine the distribution of X_1 note that

$$\mathbb{P}(X_1 > t) = \mathbb{P}(N(t) = 0) = e^{-\lambda t}$$

since $N(t)$ is a Poisson random variable with parameter λt. Thus, X_1 has an $\text{Exp}(\lambda)$ distribution.

Now consider, X_2 then

$$\begin{aligned}
\mathbb{P}(X_2 > t \,|\, X_1 = s) &= \mathbb{P}(0 \text{ events in } (s, s+t] \,|\, X_1 = s) \\
&= \mathbb{P}(0 \text{ events in } (s, s+t]) \\
&= e^{-\lambda t}\,.
\end{aligned}$$

The inter-arrival times $X_1, X_2, \ldots$ are independent, identically distributed random variables with distribution Exp (λ).

Time for first n events

Let $S_n = \sum_{i=1}^{n} X_i$ be the (random) time for the first n events in a Poisson process. Then

$$
\begin{aligned}
\mathbb{P}(S_n \le t) &= \mathbb{P}(N(t) \ge n) \\
&= \sum_{j=n}^{\infty} e^{-\lambda t} \frac{(\lambda t)^j}{j!}
\end{aligned}
$$

So, the pdf of S_n is given by differentiating

$$
\begin{aligned}
f_{S_n}(t) &= \sum_{j=n}^{\infty} j\lambda e^{-\lambda t} \frac{(\lambda t)^{j-1}}{j!} - \sum_{j=n}^{\infty} \lambda e^{-\lambda t} \frac{(\lambda t)^j}{j!} \\
&= \sum_{j=n}^{\infty} \lambda e^{-\lambda t} \frac{(\lambda t)^{j-1}}{(j-1)!} - \sum_{j=n}^{\infty} \lambda e^{-\lambda t} \frac{(\lambda t)^j}{j!} \\
&= \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!} \,.
\end{aligned}
$$

Thus, $S_n = \sum_{j=1}^{n} X_i$ has a Gamma $\Gamma(n, \lambda)$ distribution.

Non-homogeneous Poisson processes

The Poisson process has a constant rate of events, λ, but we can relax this assumption to use a time-dependent rate function $\lambda(t)$ to produce a non-homogeneous Poisson process as follows.

- $N(0) = 0$;
- The No of events in disjoint time intervals are independent;
- $$\mathbb{P}(\text{exactly } i \text{ events occur in } (t, t+h]) = \begin{cases} 1 - \lambda(t)h + o(h) & i = 0 \\ \lambda(t)h + o(h) & i = 1 \\ o(h) & \text{otherwise}\,. \end{cases}$$

Simulation techniques

Introduction

The main building block of a simulation study is a source of random variables.

We will begin by looking at the generation of sources of $U(0, 1)$ random variables and then show how this leads to the generation of random variables with arbitrary distributions, both continuous and discrete.

Random number generation

It is important *not* to generate random numbers with an ad hoc method — complex algorithms do not necessarily generate random outputs.

Some operating systems include support for generating random numbers based on (e.g.) key strokes or network inter-arrival times — however, these mechanisms are not usually suited to generating large volumes of random data.

How can we generate a large *random* sequence in a *deterministic manner*?

The answer is that the sequence is not random, but appears random as far as can be determined from statistical tests. The sequence is termed *pseudo-random*.

Random number generation (2)

Important requirements include

- The algorithm should be fast;
- The storage requirements should be low;
- The random sequence should only repeat after a very long period;
- The sequence generated should possess two important statistical properties: uniformity and independence.

For ease of implementation and speed, most random number generators use integer representation over an interval m

Given a value in this range, a desired value in the range $(0, 1)$ can be obtained by dividing by m

Multiplicative congruential method

A popular method, known as the multiplicative congruential method, starts with a seed value, X_0, and then recursively constructs the successive values X_n by the equation

$$X_n = (aX_{n-1}) \quad \text{modulo} \quad m$$

where a and m are positive integers. The values X_n lie in the range $0, 1, 2, \ldots, m-1$.

The sequence of values X_n has period at most m and so we should wish to choose a and m such that the period remains large whatever the seed value X_0.

A common choice is $m = 2^{31} - 1$ and $a = 7^5 = 16,807$.

Mixed congruential method

An extension to the multiplicative congruential method is the mixed congruential method which adds an additive term to the recursion

$$X_n = (aX_{n-1} + c) \quad \text{modulo} \quad m$$

We will not investigate such methods further here. Instead we will assume that we have an efficiently generated supply of random numbers distributed as independent $U(0, 1)$ random variables.

Random variable generation

We now have a sequence of pseudo-random uniform variables. How do we generate variables from other distributions?

We will find that with a suitable transformation random behaviour can be programmed so that the resulting random variables appear to have been drawn from any desired probability distribution.

Discrete distributions

Suppose we are given a distribution p_i with $0 \leq p_i \leq 1$ for $i = 0, 1, \ldots$ and with $\sum_i p_i = 1$ and wish to generate a discrete random variable whose probability distribution function is

$$\mathbb{P}(X = x_i) = p_i \qquad i = 0, 1, \ldots$$

This may be done by generating a (pseudo-)random variable U with distribution $U(0, 1)$ and setting

$$X = \begin{cases} x_0 & \text{if } U < p_0 \\ x_1 & \text{if } p_0 \leq U < p_0 + p_1 \\ \vdots & \\ x_i & \text{if } \sum_{j=0}^{i-1} p_j \leq U < \sum_{j=0}^{i} p_j \\ \vdots & \end{cases}$$

The inverse transform method

For now since U is $U(0,1)$

$$\mathbb{P}(X = x_i) = \mathbb{P}\left(\sum_{j=0}^{i-1} p_j \leq U < \sum_{j=0}^{i} p_j\right) = p_i$$

If we write $F(x_k) = \sum_{i=0}^{k} p_i$ then the process of generating X is given by

- Generate a random number U
- If $U < F(x_0)$ set $X = x_0$ and stop
- If $U < F(x_1)$ set $X = x_1$ and stop
- $\vdots$

If the x_i are ordered so that $x_0 < x_1 < \cdots$ then this is equivalent to choosing $X = x_i$
if $F(x_{i-1}) \leq U < F(x_i)$ and for this reason the method is known as the inverse transform method.

Geometric random variables

Here

$$p_i = \mathbb{P}(X = i) = p(1-p)^{i-1} \quad i = 1, 2, \ldots$$

and so

$$\sum_{j=1}^{i-1} p_j = 1 - \mathbb{P}(X > i-1)$$
$$= 1 - (1-p)^{i-1}\,.$$

Thus, we can use the inverse transform method by setting X to the value of i such that

$$1 - (1-p)^{i-1} \leq U < 1 - (1-p)^{i}\,.$$

A little algebra shows that we can write this as

$$X = \left\lfloor \frac{\log(U)}{\log(1-p)} \right\rfloor + 1\,.$$

Poisson random variables

Here we have for $\lambda > 0$

$$p_i = \mathbb{P}(X = i) = e^{-\lambda}\frac{\lambda^i}{i!} \quad i = 0, 1, \ldots .$$

Hence, it follows that

$$p_{i+1} = \frac{\lambda}{i+1} p_i \quad i = 0, 1, \ldots$$

and an algorithm to generate a Poisson(λ) random variable is as follows.

1 Generate a random number U

2 Set $i = 0, p = e^{-\lambda}, F = p$

3 If $U < F$, set $X = i$ and stop

4 Set $p = \lambda p/(i+1), F = F + p, i = i + 1$

5 Go to step 3

Clearly, similar algorithms can be formulated for other discrete distributions.

Generating continuous random variables

Let U be a random variable with distribution $U(0,1)$ then

$$X = F_X^{-1}(U)$$

is a random variable with cdf $F_X(x)$.

Proof:

$$\begin{aligned}
\mathbb{P}(X \le x) &= \mathbb{P}(F_X^{-1}(U) \le x) \\
&= \mathbb{P}(F_X(F_X^{-1}(U)) \le F_X(x)) \\
&= \mathbb{P}(U \le F_X(x)) \\
&= F_X(x)\,.
\end{aligned}$$

So, the inverse transform method for continuous random variables with cdf $F_X(x)$ generates $X = F_X^{-1}(U)$ where U is $U(0,1)$.

Uniform distribution (a, b)

Consider the *uniform* random variable on the interval (a, b) with distribution function

$$F_X(x) = (x - a)/(b - a)$$

for x in the interval (a, b).

Given U in interval $(0, 1)$ we set

$$X = F_X^{-1}(U)$$

and so

$$U = F_X(X) = (X - a)/(b - a)$$

so that

$$X = (b - a)U + a\,.$$

Exponential distribution

For the exponential distribution with parameter λ we have

$$F_X(x) = 1 - e^{-\lambda x}, \quad x > 0$$

and so given U, a random variable with distribution $U(0,1)$, we set

$$X = F_X^{-1}(U) .$$

Thus

$$U = F_X(X) = 1 - e^{-\lambda X}$$

so that

$$X = -\frac{1}{\lambda} \log(1 - U) .$$

Note that both U and $1 - U$ are distributed as $U(0,1)$ so we might as well save a little arithmetic and set

$$X = -\frac{1}{\lambda} \log(U) .$$

Alternative method for Poisson random variables

Recall that for a Poisson process of rate λ, the number of events in $[0, 1]$, $N(1)$, is Poisson(λ). Moreover, the inter-arrival times of events X_i are Exp(λ). Hence,

$$N(1) = \max\left\{ n : \sum_{i=1}^{n} X_i \le 1 \right\} .$$

Thus $N = N(1)$ is a Poisson(λ) random variable where putting $X_i = -\log(U)/\lambda$

$$\begin{aligned}
N &= \max\left\{ n : \sum_{i=1}^{n} -\frac{1}{\lambda} \log U_i \le 1 \right\} \\
&= \max\{ n : \log(U_1 U_2 \cdots U_n) \ge -\lambda \} \\
&= \max\left\{ n : U_1 U_2 \cdots U_n \ge e^{-\lambda} \right\} \\
&= \min\left\{ n : U_1 U_2 \cdots U_n < e^{-\lambda} \right\} - 1
\end{aligned}$$

Simulating a Poisson process

Consider the problem of generating the first n event times of a Poisson process of rate λ. One way is to generate $U_1, U_2, \ldots, U_n$ random numbers each from a $U(0,1)$ distribution and then set $X_i = -\frac{1}{\lambda}\log(U_i)$. Then the first n event times are $\sum_{i=1}^{j} X_i$ for $j = 1, 2, \ldots, n$.

To generate the first T time units we could proceed as above and stop when the sum first exceeds T. Algorithmically,

1 Set $t = 0, I = 0$

2 Generate a random number U

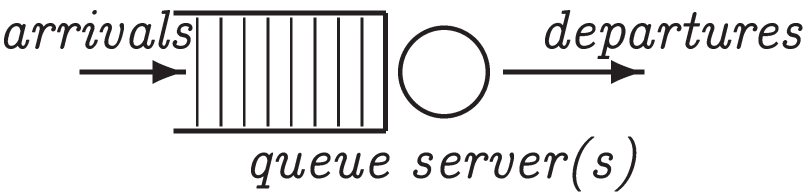
3 Set $t = t - \frac{1}{\lambda}\log(U)$. If $t > T$ stop

4 Set $I = I + 1, S(I) = t$

5 Go to step 2

will build the sequence of event times in $S(\cdot)$.

A simple queueing system



We characterise queueing systems by:

- Arrival process
  $A(t) = \mathbb{P}(\text{ inter-arrival time} \leq t)$
- Service process $B(x) = \mathbb{P}(\text{ service time} \leq x)$
- Storage capacity available for waiting customers
- The number of servers/customers available
- The different kinds of arriving customers (big jobs, small jobs,. . .)
- Queueing discipline used: FCFS, LCFS, priority, . . .
- Defections, balking, bribing, . . .

Queueing systems notation

The Kendall notation describes a single queueing system using the notation $A/B/m/k/l$ where:

- A is the inter-arrival time distribution of customers
- B is the service time distribution
- m is the number of parallel servers
- k is the limit on the customers in this system
- l is the population size

If the population size or the limit on the queue length are not specified then they are assumed to be infinite.

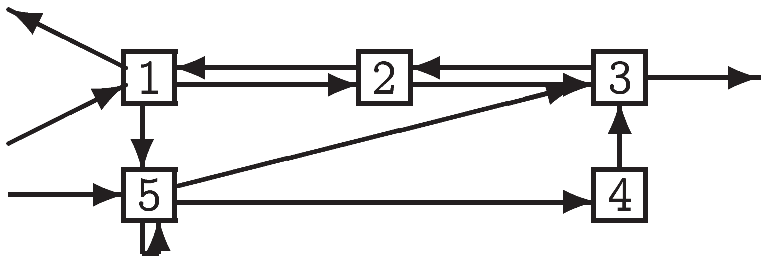
Queueing notation (2)

- M – exponential distribution
- E_r – r-stage Erlangian distribution
- D – Deterministic
- G – General

Examples:

- M/M/1: exponential inter-arrival, exponential service, single server
- M/E_r/1: exponential inter-arrival, r-stage Erlang service, single server
- M/G/1: exponential inter-arrival, general service time, single server
- M/M/K/K: exponential inter-arrival, exponential service, K servers and at most K customers present

Queueing networks



More generally, consider systems comprising multiple inter-connected service centres, forming a *queueing network*. Consider:

- the properties of each node

  — e.g. using Kendall notation

- the way in which jobs move between the nodes

  — e.g. the links between nodes and the ways jobs move between them

- the workload being analyzed

  — e.g. disk-server workload comprises a $20:1$ mix of small/large requests

Queueing networks (2)

We can classify queueing networks as either

- *closed* networks in which a fixed set of jobs circulate between nodes, but no new jobs are introduced and no jobs leave the system

  — e.g. a computer system with a fixed number of terminals attached to it.

- *open* networks in which jobs may enter and leave the system

  — e.g. the network on the previous slide: jobs arrive at 1 or 5 and leave from 1 or 3.

Open networks may be further classified as *feed-forward* if a single job visits each node *at most* once.

Simulation

Simulation allows arbitrarily complex systems to be evaluated

- Able to capture the dynamic behaviour of systems
- Captures the dynamics of complex systems by imitation
- Tracks the evolution of the system over time
- Examples include communication network design, road traffic modelling, studying chemical reactions, fluid flow, etc.

Simulation (2)

Execution of a simulation model consists of a series of state space changes.

Consider simulation as a ‘set of equations’ describing evolution in time of system under study.

The ‘equations’ are ‘solved’ by following their evolution in time.

We *always* follow the *dynamic* evolution of the system, even if we only want a mean value — therefore, as well as techniques for implementing simulators, it is necessary to know how to analyse their results — more later.

Simulation is of particular use when we are studying systems that are not in steady state.

Types of simulation

- *Discrete state/event* simulation in which the state of the system is described by discrete variables

  — e.g. the number of jobs at different stages on a production line

- *Continuous state/event* simulation in which the state is described by continuous variables

  — e.g. the quantities of various chemical reagents in a vat

A similar distinction may be drawn between *discrete time* and *continous time* simulations depending on whether the system state is defined at all times.

— e.g. a simulation of the number of students attending these lectures would be discrete time.

Types of simulation (2)

We will be concerned with *discrete event* simulation because it applies most naturally to computer systems in which state variables are generally discrete, e.g.

- the state of jobs in the system;
- the number of jobs of different kinds;
- the number or availability of devices.

Pros and cons

The principal advantage of simulation is its extreme generality. However, . . .

- The design, coding and debugging of a simulation program is often time consuming and difficult to understand — it may even approach that of implementing the system and measuring it directly!
- Generality can lead to complexity which can obscure understanding of the model — fine details may be irrelevant if the simulated workload is a poor estimate.
- Execution of the simulation can be computationally expensive.
- Statistical analysis of the output can be problematic — e.g. how long to run the simulation before averaging the results?

Events

Each event contains a time stamp identifying 'when it occurs' and denotes some change in the state of the system to be simulated e.g. 'packet arrived'

Events are ordered in time in an *event list*

- Initialize the clock to 0
- Initialise the event list
- WHILE termination criterion is not met
  - remove a smallest tuple (t,m) from the event list
  - update the clock to t
  - simulate the effect of transmitting m at time t

It is *crucial* that the simulator always selects the event with the *least* time stamp

Frequently most of the simulation time is spent maintaining the chronological order of the event list.

Simulation variables

As we have seen, discrete event simulations involve both events and variables and where we continually keep track of variables as they change during a simulation.

In general, there are three types of variables.

- Time variable, t, to record the passage of time during the simulation;
- Counter variables which keep count of the number of times that certain events have occurred by time t;
- System state variables which define the state of the system at time t.

As simulation events occur we change these variables and gather any output of interest.

Physical systems

Concerned with the problem of representing so-called 'physical systems'

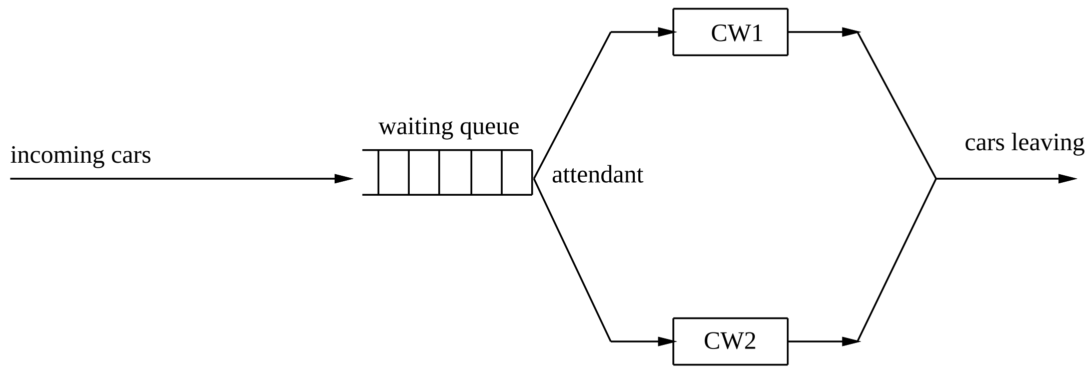
A physical system consists of one or more physical processes

Each physical process operates independently, aside from interaction via messages (events)

All state changes are summarized in the event exchanges

Note the similarity with object-oriented styles of programming in which objects communicate solely by method invocations. Older object-oriented languages tended to refer to these as message-send operations.

Example



An automatic car wash which is able to service one car at a time — is it viable to install a second car wash?

Model: attendant (**att**), car washes, (**cw1**, **cw2**), entrance (**source**) and exit (**sink**).

- The inter-arrival time of cars that need to be washed is randomly distributed
- If both car washes are busy, an arriving car joins the queue
- When a car wash becomes idle then the car at the head of the queue is sent to it.

Example (2)

Events:

- **source**$\rightarrow$**att**: car arrives in the system
- **att**$\rightarrow$**cw**: car sent from the queue to the car wash
- **cw**$\rightarrow$**att**: car wash becomes idle
- **cw**$\rightarrow$**sink**: car departs the system

Note how the departure of a car is modelled by *two* events: one representing the car leaving the system and the other that signals to the attendant that the car wash is now idle.

Given that it takes **cw1** 8 minutes and **cw2** 10 minutes to wash a car, a *possible* sequence of events is . . .

Sequence of events

message	event	time	sender	receiver	content
1	-	0	cw1	att	idle
2	-	0	cw2	att	idle
3	1	6	source	att	car 1
4	2	6	att	cw1	car 1
5	3	11	source	att	car 2
6	4	11	att	cw2	car 2
7	5	12	source	att	car 3
8	6	14	cw1	sink	car 1
9	-	14	cw1	att	idle
10	7	14	att	cw1	car 3
11	8	17	source	att	car 4
12	9	19	source	att	car 5
13	10	21	cw2	sink	car 2
14	-	21	cw2	att	idle

Modelling stochastic systems

To represent the stochastic nature of the systems being modelled, simulation requires the use of random variables.

An input to the simulation may have a stochastic distribution e.g. arrivals at a queue.

Simulation model needs to sample random variables from the given distribution to "re-create" the input process.

Random variables can be generated for a wide range of theoretical and empirical distributions.

This process requires only a sequence of independent random variables with a uniform distribution!

Simulating a single server queue

As a more detailed example consider the simulation of a single server queue to which customers arrive according to a (homogeneous) Poisson process of rate λ.

On arrival a customer either enters service immediately (if the server is free) or waits in a queue of customers (if the server is busy). When a customer departs the server the customer who has been waiting longest (FIFO discipline) enters service or, if the queue is empty, the server remains idle until the next customer arrives.

The times taken to serve each customer are independent, identically distributed random variables with probability distribution function G.

Simulating a single server queue (2)

Let T be a fixed time beyond which customers are nolonger allowed to enter the system.

Beyond time T the server continues to serve all remaining customers until the system is empty.

We could use the simulation to estimate

- the average time a customer spends in the system;
- the average time beyond T that the last customer departs.

Simulating a single server queue (3)

Variable	Description
Time	t
Counter	N_A, the No of arrivals
	N_D, the No of departures
System state	n, the No of customers

The event list will consist of two elements: t_A, t_D, the times of the next arrival and next departure, respectively. If there is no customer in service then put $t_D = \infty$.

We will output $A(i)$ and $D(i)$ the times of arrival and departure of customer i together with T_f the time past T that the last customer departs the system.

Simulating a single server queue (4)

Within the simulation we use the random variable T_s as the time of the next arrival after time s and Y as a random service time chosen from the given distribution G.

The initialization of the variables is as follows.

- Set $t = N_A = N_D = 0$;
- Set $n = 0$;
- Generate T_0 and set $t_A = T_0$ and $t_D = \infty$.

The simulation advances event by event updating the variables according to one of the following cases.

Simulating a single server queue (5)

Case I: $t_A \leq t_D$, $t_A \leq T$

- Reset $t = t_A$, $N_A = N_A + 1$, $n = n + 1$;
- Generate T_t, reset $t_A = T_t$;
- If $n = 1$, generate Y and reset $t_D = t + Y$;
- Collect output data $A(N_A) = t$.

Case II: $t_D \leq t_A$, $t_D \leq T$

- Reset $t = t_D$, $n = n - 1$, $N_D = N_D + 1$;
- If $n = 0$ reset $t_D = \infty$ else generate Y and reset $t_D = t + Y$;
- Collect output data $D(N_D) = t$.

Simulating a single server queue (6)

Case III: $\min(t_A, t_D) > T$, $n > 0$

- Reset $t = t_D$, $n = n - 1$, $N_D = N_D + 1$;
- If $n > 0$ generate Y and reset $t_D = t + Y$;
- Collect output data $D(N_D) = t$.

Case IV: $\min(t_A, t_D) > T$, $n = 0$

- Collect output data $T_f = \max(t - T, 0)$.
- Stop.

Each simulation run will produce the quantities $D(i) - A(i)$ giving the amounts of time that customers spent in the system and the value T_f.

We then use the numerical average value of these quantities as our estimates.

Simulation performance measures

As the simulation itself is stochastic, so too are the observed outputs.

It is critical to realize that a simulation can only yield *estimates* for performance measures

There will *always* be some statistical error in the estimates.

We can attempt to reduce the error by

- running the simulation for longer until sufficient samples have been taken;
- running the same simulation with a different pseudo-random number sequence, and combining the results from multiple runs. We will say more on this later.

Utilization

The utilization is the proportion of time that a server is busy.

An estimate can therefore be obtained by taking the sum of the busy times of the server and dividing by T, the simulation length.

In the case of a k-server, the busy times can be estimated together and divided by kT.

There are two obvious ways to aggregate the busy times:

- Sample the system and observe whether servers are busy or idle
- When the server becomes busy the time is saved, and when it becomes idle again the difference in the two times is the busy period.

Throughput, queue length

A simple count is kept of the number of customers receiving service. The *throughput* is this value divided by the total time.

There are at least two ways to estimate *queue length*:

- estimate the queue length distribution, then use that to obtain the mean;
- view the queue length as a function of time: the mean queue length is 1/T of integral of this function.

Queue length

- Each time the queue length changes the time t_{i+1} is stored;
- subtract previous recorded time t_i from current time;
- multiply by previous queue length n_i.

Sum these areas to give mean queue length, N, (averaged over M observations) by

$$N = \frac{1}{T} \sum_{i=1}^{M} n_i (t_{i+1} - t_i)$$

Queueing time

Two obvious ways to obtain *queueing time*

- observe the queueing times and take their average;
- simply use Little's law.

Statistical analysis of results

Suppose that $X_1, X_2, \ldots, X_n$ are independent, identically distributed random variables and let μ and σ^2 be their common mean and variance, respectively.

Hence,

$$\mu = \mathbb{E}(X_i)$$

and

$$\sigma^2 = \mathrm{Var}(X_i) = \mathbb{E}((X_i - \mathbb{E}(X_i))^2) \, .$$

Given a sample of values of the random variables $X_1, X_2, \ldots, X_n$ how might we estimate the values of the parameters μ and σ^2 and how does the sample size, n, affect the accuracy of our estimates?

Sample mean

The quantity

$$\overline{X} := \frac{1}{n} \sum_{i=1}^{n} X_i$$

is called the sample mean. Note that the sample mean is another random variable with mean

$$\begin{aligned}
\mathbb{E}(\overline{X}) &= \mathbb{E}\left(\sum_{i=1}^{n} \frac{X_i}{n}\right) \\
&= \sum_{i=1}^{n} \frac{\mathbb{E}(X_i)}{n} \\
&= \frac{n\mu}{n} \\
&= \mu \,.
\end{aligned}$$

Sample mean (2)

The variance of $\overline{X}$, which we can then think of as the mean squared error, is given by

$$\begin{aligned}\mathbb{E}((\overline{X}-\mu)^2) &= \mathrm{Var}(\overline{X}) \\ &= \mathrm{Var}\left(\sum_{i=1}^{n} \frac{X_i}{n}\right) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}(X_i) \\ &= \frac{\sigma^2}{n}\,.\end{aligned}$$

Thus, $\overline{X}$ can be used to estimate μ from the sample. It is a good estimator of μ when $\sigma/\sqrt{n}$ is small.

Sample variance

Call the random variable

$$S^2 := \frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n-1}$$

the sample variance.

It is possible to show that

$$\mathbb{E}(S^2) = \sigma^2$$

and so S is a suitable estimator for the true standard deviation σ.

It is important to have an estimator for σ since we have seen that the accuracy of our estimator $(\overline{X})$ for the true mean μ depends on this variance.

Confidence intervals

We can use the Central Limit Theorem to see that for large sample sizes, n, the random variable

$$\frac{(\overline{X} - \mu)}{\sigma/\sqrt{n}} = \sqrt{n}\frac{(\overline{X} - \mu)}{\sigma}$$

is approximately distributed $N(0, 1)$.

Additionally, we may not know the true variance σ^2 but instead need to estimate it by S^2 so that

$$\sqrt{n}\frac{(\overline{X} - \mu)}{S}$$

is approximately distributed $N(0, 1)$.

Confidence intervals (2)

Write z_α for the value such that $\mathbb{P}(Z > z_\alpha) = \alpha$ where Z is a standard normal random variable $N(0, 1)$ then it follows that

$$\mathbb{P}\left(-z_{\alpha/2} < Z < z_{\alpha/2}\right) = 1 - \alpha$$

and so by the CLT

$$\mathbb{P}\left(-z_{\alpha/2} < \sqrt{n}\frac{(\overline{X} - \mu)}{S} < z_{\alpha/2}\right) \approx 1 - \alpha$$

or, equivalently,

$$\mathbb{P}\left(\overline{X} - z_{\alpha/2}\frac{S}{\sqrt{n}} < \mu < \overline{X} + z_{\alpha/2}\frac{S}{\sqrt{n}}\right) \approx 1 - \alpha\,.$$

The interval $\overline{X} \pm z_{\alpha/2}S/\sqrt{n}$ is an (approximate) $100(1 - \alpha)$ percent confidence interval for μ.

Student's t-distribution

If it is known that the common distribution of the variables X_i is also Normal then

$$\sqrt{n}\frac{(\overline{X} - \mu)}{S}$$

has (exactly) a distribution called the Student's t-distribution with $n - 1$ degrees of freedom.

Thus, an alternative confidence interval for μ is to take t_α in place of z_α where t_α is defined analogously as

$$\mathbb{P}(T > t_\alpha) = \alpha$$

where T is a random variable with the Student t-distribution with $n - 1$ degrees of freedom.

Stopping rules

How do we know when a system has been run 'long enough' for performance measures to be accurate?

We can repeat the simulation several times to obtain many samples. These multiple runs are called *replications*.

Having repeated the experiment n times, we can construct a confidence interval on our measure L, say.

Although large numbers of replications reduce the variance in the sample, each replication requires re-stabilizing the simulation.

Can we avoid this?

We may be able to use a single, long run, and break up our samples into n blocks, each of these can form a sample L_i.

Stopping rules (2)

What could go wrong with this?

Correlation between successive blocks could mean that we have biased samples.

If the block size is large then the correlation should be small.

Explicit techniques exist to estimate the correlation and obtain the block size.

The simulation can be stopped once the estimate of L becomes stable. For example, once the confidence interval around L becomes sufficiently narrow.

Stopping rules (3)

Suppose we wish to run the simulation until our $100(1-\alpha)$ confidence interval for the true mean value is of at most of width ℓ, say.

We can guarantee this by means of the following algorithm

- Generate at least 100 data values;
- Repeatedly, generate additional data values, stopping when the number of values generated, n, is such that $2z_{\alpha/2}S/\sqrt{n} < \ell$

The initial 100 data values is for illustration, a suitable value will depend on the precise simulation experiment under consideration.

The intention is to suppress the effects of the initial transient phase.

Goodness of fit tests

How might we validate our simulation model?

One area of concern is the assumption of various probabilistic distributions in the model. For example, how sure are we in the use of a Poisson distribution for the numbers of events in a given interval or the choice of the distribution G for the service times of customers in a queue.

Statistical procedures have been developed to help with these questions. The hypothesis of a particular distribution can be tested by observing the system and then asking whether the assumption of a particular distribution is 'consistent' with the data. Such procedures are known as statistical goodness of fit tests.

Chi-Square test for discrete data

Suppose we have n independent random variables $Y_1, Y_2, \ldots, Y_n$ and we wish to test the null hypothesis that they have the distribution

$$\mathbb{P}(Y_j = i) = p_i \qquad i = 1, 2, \ldots, k\,.$$

If we let N_i be the number of Y_j equal to i then we expect under the null hypothesis that

$$\mathbb{E}(N_i) = np_i \qquad i = 1, 2, \ldots, k\,.$$

Thus we should consider rejecting the null hypothesis when

$$T := \sum_{i=1}^{k} \frac{(N_i - np_i)^2}{np_i}$$

is large. How large is too large?

Chi-Square test (2)

It can be shown that under the null hypothesis and when n is large that the distribution of T is approximately a chi-squared random variable with $k-1$ degrees of freedom.

Thus, we can assess the value of

$$\mathbb{P}(T > t)$$

where t is the observed value of T from standard tables of the chi-squared distribution.

Typically, we would reject the null hypothesis when

$$\mathbb{P}(T > t)$$

has a value less than 0.05 or, more conservatively, as low as 0.01. Otherwise, we say that the observed data appears consistent with the null hypothesis.

Kolmogorov-Smirnov test for continuous data

Suppose now that we wish to test whether n independent random variables $Y_1, Y_2, \ldots, Y_n$ arise from a common continuous distribution $F(x)$.

First we observe the n random variables and construct the empirical ditribution defined by

$$F_e(x) := \frac{\text{No of } i \text{ such that } Y_i \leq x}{n} \, .$$

This will measure the proportion of observed values less than or equal to x and so should be 'close' to the function $F(x)$ under the null hypothesis.

Kolmogorov-Smirnov test for continuous data (2)

Consequently, we would expect the quantity

$$D = \max_x |F_e(x) - F(x)|$$

to be small and we should reject the null hypothesis if D is too large.

The quantity D is called the Kolomogorov-Smirnov test statistic.

The distribution will depend on the sample size n and has been tabulated.

Other tests for randomness

Various statistical tests are available.

Runs tests, which examine the arrangement of numbers in a sequence (a run) to test the hypothesis of independence.

These tests frequently check for the number of "up runs", the number of "down runs", and the runs above and below the mean.

Autocorrelation tests check the correlation structure of the sequence of observations.

Variance reduction techniques

So far we have used the sample mean, $\overline{X}$, as our estimator for μ, the mean value of our distribution. We know that

$$\mathbb{E}(\overline{X}) = \mu \qquad \text{and} \quad \mathrm{Var}(\overline{X}) = \frac{\sigma^2}{n}$$

where n is the sample size.

Might we be able to find an alternative estimator for μ which has smaller variance?

Such variance reduction techniques can sometimes produce significant speed-ups in the simulation.

Antithetic variables

Suppose that X_1 and X_2 are two identically distributed random variables with common mean $\mathbb{E}(X_1) = \mathbb{E}(X_2) = \mu$. Then

$$\mathrm{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4}\left(\mathrm{Var}(X_1) + \mathrm{Var}(X_2) + 2\mathrm{Cov}(X_1, X_2)\right) .$$

Hence, we would get a reduced variance by using $(X_1 + X_2)/2$ when X_1 and X_2 are negatively correlated.

Recall that in the inverse transform method we generate (pseudo) random numbers by first generating pseudo random numbers $U_1, U_2, \ldots$ with a $U(0,1)$ distribution. But then the numbers $1 - U_1, 1 - U_2, \ldots$ are also random numbers with a $U(0,1)$ distribution and these two series of numbers are negatively correlated.

Antithetic variables (2)

It often happens in practice that dividing the simulation runs into two groups and using $1 - U$ for the second group in place of U in the first group yields two random variables X_1 and X_2 which are negatively correlated.

In this case we say that X_1 and X_2 are antithetic variables.

Example

Consider a queueing system and let D_i be the delay in the queue of the ith customer and suppose we wish to estimate

$$\mathbb{E}(D_1 + D_2 + \cdots + D_n)$$

the sum of the delays of the first n customers.

We should expect to require a collection of $2n$ random variables U_j (one for each arrival and departure event per customer).

Repeating the simulation using the $2n$ random numbers given by $1 - U_j$ will then give an improved estimator compared to using a 'fresh' set of $2n$ random numbers.

Control variates

Suppose we run a simulation and gather from the output a random variable X for estimating μ with $\mathbb{E}(X) = \mu$.

Now, suppose that we also gather another random variable, Y, from the same output with a known mean value $\mathbb{E}(Y) = \mu_Y$.

Hence, for any number c

$$Z = X + c(Y - \mu_Y)$$

is also an estimator for μ since clearly $\mathbb{E}(Z) = \mu$.

What is the best choice of c?

Control variates (2)

Note that

$$\begin{aligned} \mathrm{Var}(Z) &= \mathrm{Var}(X + c(Y - \mu_Y)) = \mathrm{Var}(X + cY) \\ &= \mathrm{Var}(X) + c^2\mathrm{Var}(Y) + 2c\mathrm{Cov}(X, Y) \end{aligned}$$

and so, using calculus, the variance is minimized by taking $c = c^*$ where

$$c^* = -\frac{\mathrm{Cov}(X, Y)}{\mathrm{Var}(Y)}$$

and then

$$\mathrm{Var}(X + c^*(Y - \mu_Y)) = \mathrm{Var}(X) - \frac{(\mathrm{Cov}(X, Y))^2}{\mathrm{Var}(Y)}\,.$$

The variable, Y, is called the control variate for X.

Example

Suppose we are simulating a queueing system and we are interested in estimating the total time spent in the system by all customers arriving before time t. If W_i is the amount of time spent in the system by the ith customer then we wish to estimate $\mu = \mathbb{E}(X)$ where

$$X = \sum_{i=1}^{N(t)} W_i$$

where $N(t)$ is the number of arrivals by time t.

Example (2)

If the service time of the ith customer is S_i then

$$Y = \sum_{i=1}^{N(t)} S_i$$

can act as a control variate and we have from known quantities that

$$\mathbb{E}(Y) = \mu_Y = \mathbb{E}(S)\mathbb{E}(N(t))\,.$$

Note that in order to compute the optimal choice c^* for c we would have to estimate the variance $\mathrm{Var}(Y)$ and covariance $\mathrm{Cov}(X, Y)$ from the simulated data.

Operational analysis

Operational Analysis

Based on the idea of *observation* rather than a probabilistic description of system behaviour.

It is also concerned with quantities 'directly related' to these observed quantities.

Operational analysis makes very weak assumptions about the system being modelled

— unlike simulation which requires detailed system knowledge, or the techniques from queuing theory (in the next section) which depend extensively on the probability distributions involved

We begin by examining some *fundamental quantities* and *operational laws*.

Operational analysis (2)

We examine a system for some time recording the customer arrivals and departures, and define the quantities of interest:

- T, the length of time we observe the system
- A, the number of arrivals observed
- C, the number of departures (or completions of service) observed
- W, the job-time product: the sum of the durations of all customers over the observation period

If the system is a single resource, then we can also measure:

- B, the time for which the resource was busy.

Operational analysis (3)

From these we can define the following quantities:

- *Arrival rate*, $\lambda := \frac{A}{T}$

  — the mean number of arrivals per unit time

- *Throughput*, $X := \frac{C}{T}$.

  — the mean number of departures per unit time

- *Mean number of customers*, $N := \frac{W}{T}$

  — the job-time product in terms of N and T

- *Mean residence time*, $R := \frac{W}{C}$

  — the job-time product in terms of R and C

Operational analysis (4)

For a single resource, we can also define:

- *Utilization*, $U := \frac{B}{T}$

  — the proportion of the time that the resource is busy

- *Average service requirement*, $S := \frac{B}{C}$

  — the mean time that the resource spends for each departure

The utilization law

Our first "law" is just an algebraic identity

$$U := \frac{B}{T} = \frac{C}{T}\frac{B}{C} = XS$$

This is termed the *utilization law*.

For example, if the throughput (X) is 5 departures/sec and the service demand (S) is 0.1 sec/departure then the utilization (U) is 50%.

Little's law

Similarly, we can derive the familiar *Little's Law*

$$N := \frac{W}{T} = \frac{C}{T}\frac{W}{C} = XR$$

For example, if the throughput (X) is 5 customers/sec and the mean residence time is 1 sec then the average number in the system is 5.

- Very weak assumptions about the system
- Applicable to a wide range of systems
- Can be applied recursively to subsystems and to individual resources

— but take care that mutually-consistent values are used for X and R; in particular, whether they apply to the queue, the server or the entire system

An example

Observe system for $T = 10$ sec

4 customers spend 10 s in the system

One customer spends 5 s in the system

Then we have $W = 4 \times 10 + 1 \times 5 = 45$ s.

If also $A = C = 5$ then

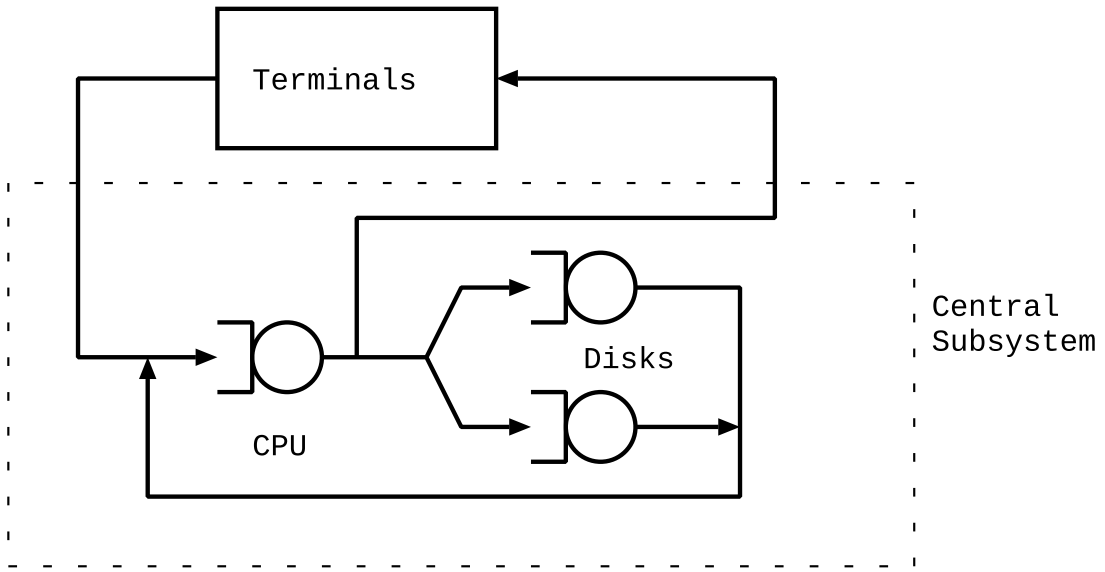
$X = C/T = 5/10 = 0.5$ customers per second

$\lambda = 5/10 = 0.5$ customers per second

$N = 45/10 = 4.5$ customers

$R = 45/5 = 9.0$ s per customer

A simple interactive system



Fixed number, M, of users logged on.

Customer is at the terminal whilst thinking.

The *think time*, Z, is the average time a user spends between receiving a prompt and responding.

A customer not thinking is somewhere inside the central subsystem.

Simple interactive system (2)

We can use Little's Law to relate some observable quantities in the central subsystem:

- N is the number of customers in the central subsystem $0 \leq N \leq M$
- X is the rate at which customers complete in the central subsystem
- R is the average time a customer spends in the central subsystem (intuitively equivalent to "response time")

If we observe that system throughput is 0.5 interactions per second and we find on average 7.5 users in the subsystem then

$$R = \frac{N}{X} = \frac{7.5}{0.5} = 15 \text{ s}$$

from Little's Law.

Simple interactive system (3)

We can also apply Little's law to the entire system.

This is a closed system so the number of customers is fixed as M.

We can split the time spent during an interaction into the response time (R) and the *think time* (Z). The *residence time* is $R + Z$.

We consequently derive the *interactive system* version of Little's Law:

$$M = X(R + Z)$$

Simple interactive system (4)

With 10 users logged on, 5 s average think time and an average response time of 15 s,

$$X = \frac{M}{R + Z} = \frac{10}{15 + 5} = 0.5 \quad \text{interactions/sec}$$

Under heavy load (M large or Z small)

$$U \approx 1$$

Using the Utilization Law the throughput $X \approx \frac{1}{S}$ and hence

$$R = \frac{M}{X} - Z \approx MS - Z$$

Thus the response time grows approximately linearly with the number of users M.

Visit counts and forced flow

We now extend our notation to allow the modelling of multiple devices. Use subscripts $i = 1, 2, \ldots, K$ to identify each device, e.g. X_i is the throughput at device i. Assume that the service required by a customer is an inherent property of the *customer* not of the state of the system.

A *visit count* for a device is the number of completions at that device for every completion from the system

$$V_i := \frac{C_i}{C}$$

Where C_i is the number of completions at device i. Recall that in a *feed-forward* queueing network, $0 \leq V_i \leq 1$ because each customer visits a given device at most once.

The forced flow law

Since $X = \frac{C}{T}$, we have that

$$X_i = \frac{C_i}{T} = \frac{C_i}{C}\frac{C}{T} = V_i X$$

the *Forced Flow Law*.

For example, if the throughput from the entire system is 20 customers per second and each customer visits a given device 3 times then the throughput of that device must be 60 completions per second.

If devices are load independent, then define the *service demand* a customer makes on a device i by

$$D_i := V_i S_i$$

— be careful to distinguish the *service requirement* (S_i) and the *service demand* (D_i)

Queue lengths at a server

Applying the utilization law at each device:

$$U_i = X_i S_i = (X V_i) S_i = X(V_i S_i) = X D_i$$

Similarly, applying Little's law at each device:

$$N_i = X_i R_i \, .$$

R_i is the residence time at device i and can be decomposed into the time spent queuing and the time spent in service, approximated by R_i^*:

$$\begin{aligned} R_i^* &= N_i S_i + S_i \\ &= R_i^* X_i S_i + S_i \\ &= R_i^* U_i + S_i \, . \end{aligned}$$

Queue lengths at a server (2)

Hence

$$R_i^* = \frac{S_i}{1 - U_i} .$$

So that

$$\begin{aligned} N_i &= X_i R_i^* \\ &= \frac{X_i S_i}{1 - U_i} \\ &= \frac{U_i}{1 - U_i} . \end{aligned}$$

Observe that

- N_i is zero when U_i is zero;
- N_i grows rapidly without bound as U_i approaches one.

Bottleneck analysis

A bottleneck in a system is a hindrance to movement or progress.

Given the forced flow assumption, at high loads system performance is determined by the device with the highest utilization: *the bottleneck*.

The ratio of the completion rates of any two devices is

$$\frac{X_i}{X_j} = \frac{V_i X}{V_j X} = \frac{V_i}{V_j} \, .$$

Since $U_i = X_i S_i$, we have a similar property for utilizations

$$\frac{U_i}{U_j} = \frac{X_i S_i}{X_j S_j} = \frac{V_i S_i}{V_j S_j} \, .$$

Bottleneck analysis (2)

A system is *load independent* if

- V_i are intrinsic properties of customers,
- S_i are independent of the queue length at i.

In such cases, the throughput and utilization ratios are the same for all loads.

This can be used to determine asymptotes for X and R.

In general, $U_i \leq 1$ and $X_i \leq \frac{1}{S_i}$.

A device i becomes *saturated* as $U_i \to 1$

Thus, as $U_i \to 1$, we have that $X_i \to \frac{1}{S_i}$: device i is working as fast as it can and consequently serves one customer every S_i.

Bottleneck analysis (3)

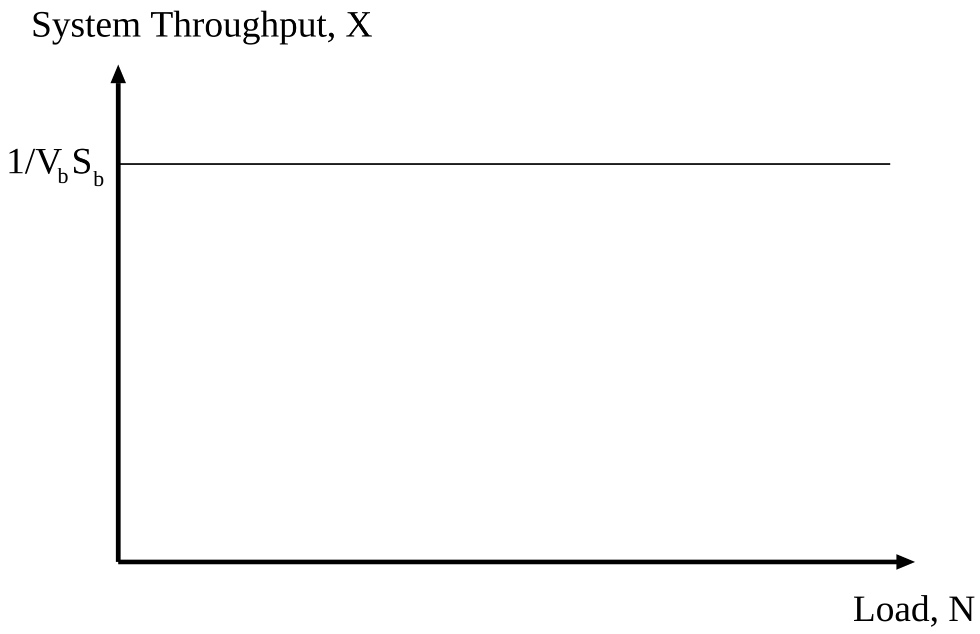
We use the subscript b to denote a device capable of saturating.

Since the utilization ratios are fixed, the device i with the largest $V_i S_i$ product will be the first to achieve 100% utilization as N increases:

$$V_b S_b = \max\{V_1 S_1, ..., V_K S_K\}$$

so the bottleneck is determined by both the device and workload (the V_i and S_i) properties.

Maximum throughput



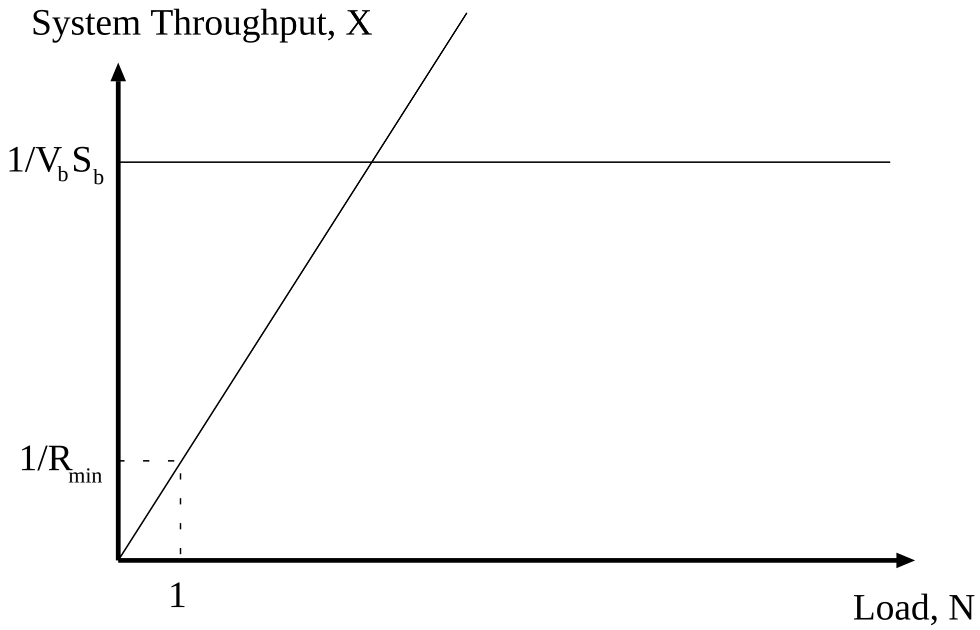
By the forced flow law

$$X = \frac{X_b}{V_b}$$

So, as $U_b \to 1$ and $X_b \to 1/S_b$

$$X_{max} = \frac{X_b}{V_b} \to \frac{1}{V_b S_b}$$

Maximum throughput (2)



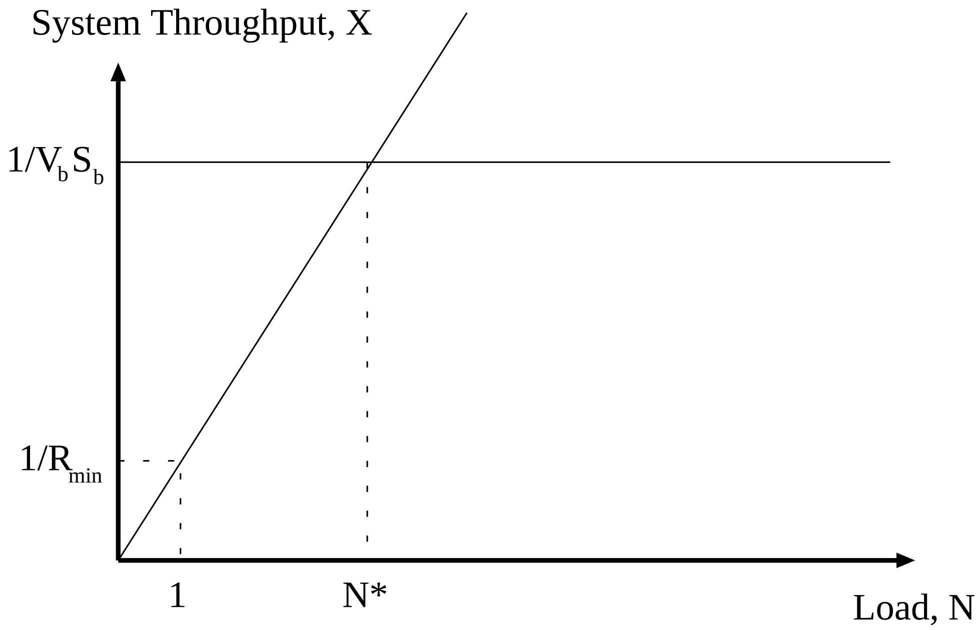
The total per-customer service required is

$$R_{min} = \sum_{i=1}^{K} V_i S_i$$

$$\Rightarrow \qquad X \leq \frac{N}{R_{min}}$$

R_{min} denotes the smallest possible value of mean response time, occurring when $N = 1$.

Maximum throughput (3)



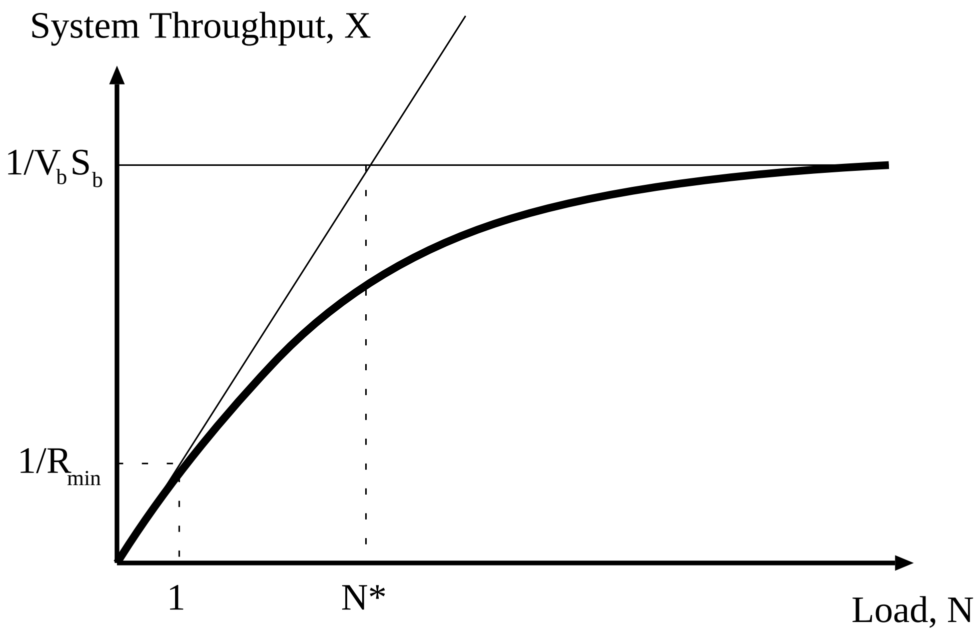
If $k \leq K$ jobs always avoid each other then

$$X = \frac{k}{R_{min}} \leq \frac{1}{V_b S_b}$$

$$k \leq \frac{R_{min}}{V_b S_b} = \frac{\sum_{i=1}^{K} V_i S_i}{V_b S_b} = N^*, \quad \text{say}$$

So, beyond N^* queueing is *certain*.

Maximum throughput (4)



- It stays below $1/(V_b S_b)$ because, at that point, a bottleneck will be operating at maximum utilization;

- It stays below the straight line $X = N/R_{min}$ because the throughput is limited by the number of customers in service.

Interactive response time

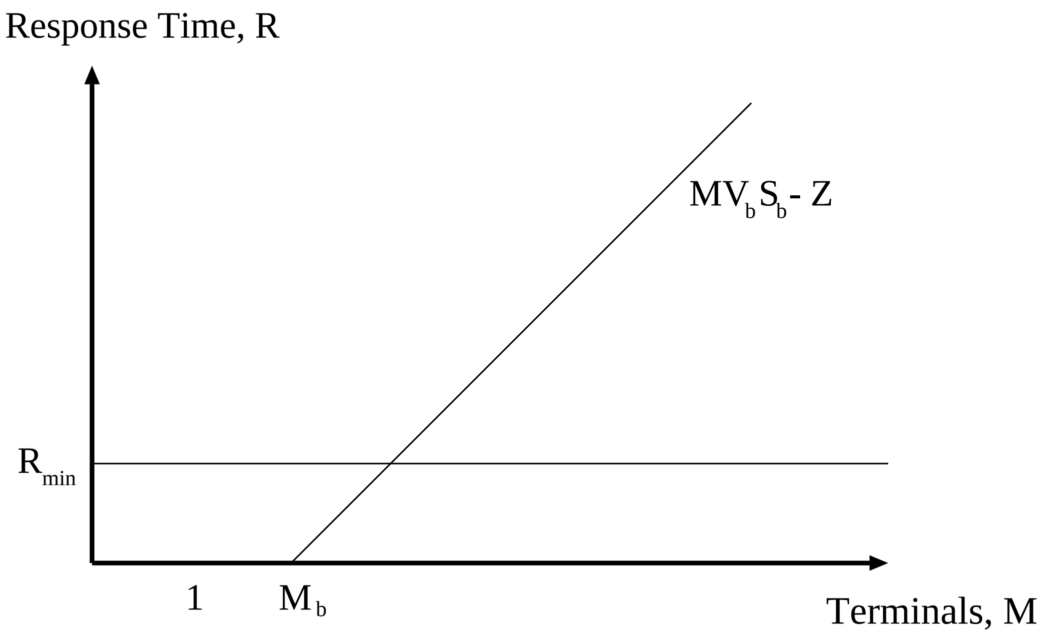
- X throughput;
- M terminals;
- Average think time Z;
- Recall the interactive system version of Little's law:

$$R = \frac{M}{X} - Z\,.$$

By considering a bottleneck device b:

$$\begin{aligned} & X \le \frac{1}{V_b S_b} \\ \Rightarrow \quad & R \ge M V_b S_b - Z \\ \Rightarrow \quad & R \ge M V_i S_i - Z \qquad \forall i \in \{1..K\} \end{aligned}$$

Interactive response time (2)



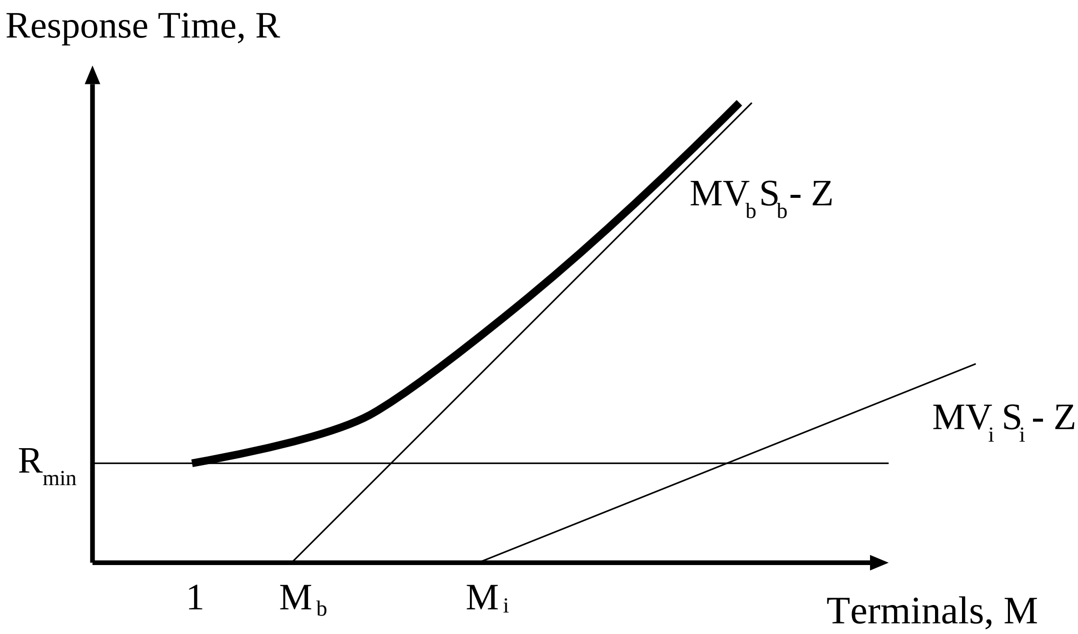
The response time asymptote meets the horizontal axis at

$$M_b = \frac{Z}{V_b S_b}$$

It intersects the minimum response time R_{min} at M_b^* (say) where

$$M_b^* V_b S_b - Z = R_{min} \, .$$

Interactive response time (3)



Thus

$$M_b^* = \frac{R_{min} + Z}{V_b S_b} = N^* + M_b$$

When there are more than M_b^* terminals, queueing is *certain* to exist.

Summary

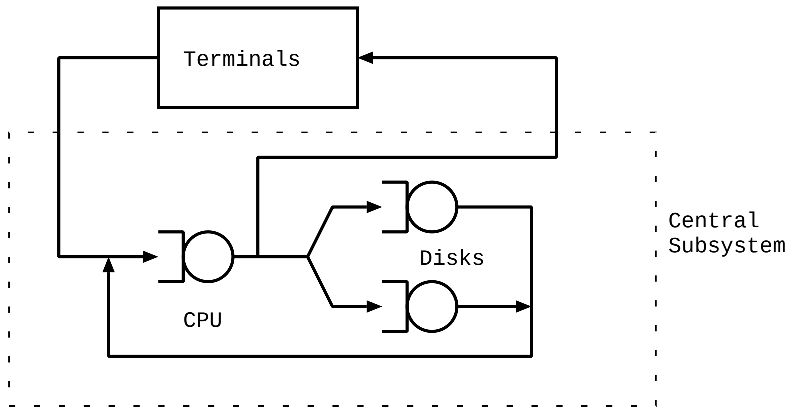
- The largest of the products $V_i S_i$ determines the bottleneck b.
- The sum of these products determines the smallest possible response time R_{min}.
- Queueing cannot be avoided when N exceeds

$$N^* = \frac{R_{min}}{V_b S_b}$$

- Queueing cannot be avoided in an interactive system when the number of logged-on terminals exceeds

$$M_b^* = N^* + \frac{Z}{V_b S_b}.$$

Example: interactive system

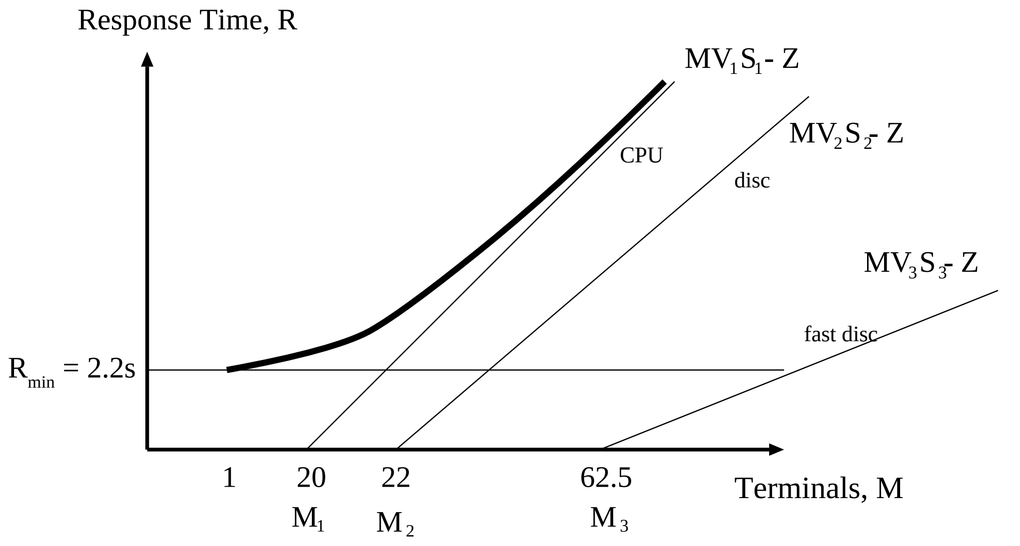


Suppose that $Z = 20$ s.

No.	device	S_i (s)	V_i	$D_i = V_i S_i$
1	CPU	0.05	20	1.00
2	disk	0.08	11	0.88
3	fast disk	0.04	8	0.32
			R_{min}	2.20

Question: Is a 8 second response time feasible with 30 users logged on? If not, what changes are required?

Example: interactive system (2)



$$V_1S_1 = 1\text{s} \quad \textit{(bottleneck)}$$

$$V_2S_2 = 0.88\text{s}$$

$$V_3S_3 = 0.32\text{s}$$

$$\Rightarrow R_{min} = \sum_{i=1}^{3} V_iS_i = 2.2\text{s}$$

For $M = 30$, the response time asymptote requires $R \geq 30 \times 1 - 20 = 10$ s.

So the answer is *no*.

Example: interactive system (3)

To make a 8 second response time feasible, we need to speed up the CPU, so that the new service time obeys the condition

$$MV_1S_1' - Z \leq 8$$

or

$$S_1' \leq \frac{20+8}{30 \times 20} = .047 \ \text{ s}$$

which is a 7% speed up in the CPU.

Then $V_1S_1' = 0.93$ is still the largest product so the CPU is still the bottleneck.

Example: interactive system (4)

Question: Is a 10s response time feasible when 50 users are logged on? If not, how much CPU speedup is required?

If $S_1 \to 0$, the disk will become bottleneck

$$R \geq MV_2S_2 - Z$$

For $M = 50$, this is

$$R \geq 50 \times 0.88 - 20 = 24 \quad \text{s}$$

so a 10s response time is not feasible with $M = 50$ and no amount of CPU speedup is capable of achieving it.

Balanced system bounds

Balanced system bounds provide *tighter* bounds at mid-range loads than bottleneck analysis

A system is *balanced* if for any load the utilizations of all devices are equal.

Balanced systems exhibit the following important property

$$U_i(N) = \frac{N}{N + K - 1} .$$

So the system throughput is given by

$$X(N) = \frac{U_i}{D_i} = \frac{N}{N + K - 1} \times \frac{1}{D_i} .$$

Balanced system bounds (2)

For example,

$$
\begin{array}{lll}
N = 1 & K = 2 & U_1 = U_2 = \frac{1}{2} \\
N = 1 & K = 100 & U_1 = \ldots = U_{100} = \frac{1}{100} \\
N = 100 & K = 2 & U_1 = U_2 = \frac{100}{101} \\
N = 2 & K = 2 & U_1 = U_2 = \frac{2}{3}
\end{array}
$$

We observe the system to determine

- D_{max} — maximum demand at any device;
- D_{min} — minimum device demand;
- D_{av} — average demand at each device;
- $D = D_{\mathrm{tot}}$ — total demand across all devices.

So, we have $D_{\mathrm{av}} = D/K$.

Balanced system bounds (3)

Consider imaginary balanced systems related to our system

pess1: balanced system with K devices each with a demand of D_{max};

opt1: balanced system with K devices each with a demand of D_{min}.

The throughput of system **pess1** is

$$\frac{N}{N+K-1} \times \frac{1}{D_{max}} \,.$$

The throughput of system **opt1** is

$$\frac{N}{N+K-1} \times \frac{1}{D_{min}} \,.$$

Balanced system bounds (4)

So for the system being modelled we have

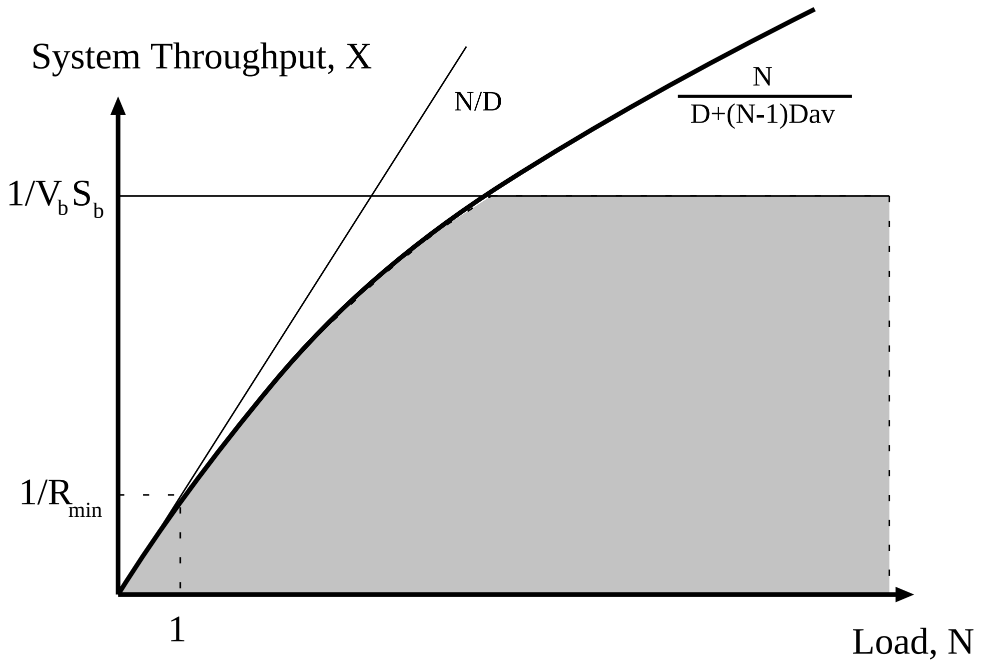
$$\frac{N}{N+K-1} \times \frac{1}{D_{\text{max}}} \leq X(N) \leq \frac{N}{N+K-1} \times \frac{1}{D_{\text{min}}} \, .$$

Since $N = XR$ we have,

$$(N+K-1)D_{\text{max}} \geq R(N) \geq (N+K-1)D_{\text{min}} \, .$$

This will give tighter bounds on the mid-range performance than the bottleneck bounds.

Balanced system bounds (5)

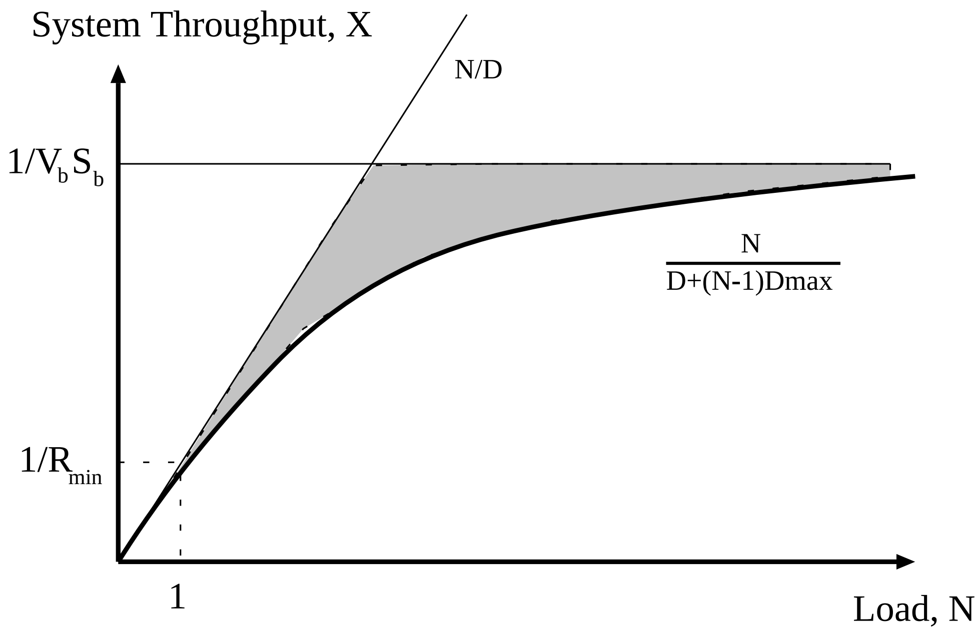


We can do even *better* by considering the *best* performance that the system can achieve which occurs when the load is spread out evenly among all the devices.

opt2: D_{av} at each of the K devices

$$X(N) = \frac{N}{N+K-1} \times \frac{1}{D_{av}} = \frac{N}{D+(N-1)D_{av}} \, .$$

Balanced system bounds (6)



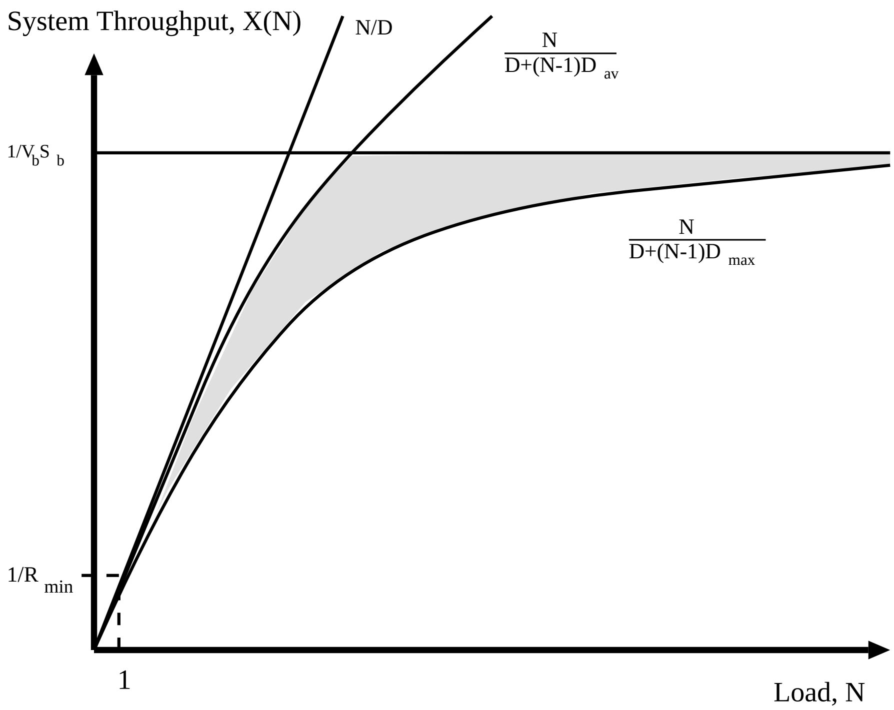
Now, what is the *worst* system subject to the constraints that D and D_{max} remain fixed?

Answer: place D_{max} at as many devices as possible and 0 at the rest

pess2: D_{max} at each of the $\frac{D}{D_{max}}$ devices

$$X(N) = \frac{N}{N+K-1} \times \frac{1}{D_{max}} = \frac{N}{D+(N-1)D_{max}} .$$

Balanced system bounds (7)



$$\frac{N}{D + (N-1)D_{max}} \leq X(N) \leq \frac{N}{D + (N-1)D_{av}}$$

$$D + (N-1)D_{max} \geq R(N) \geq D + (N-1)D_{av}$$

Balanced system bounds (8)

As $N \to \infty$,

$$\frac{N}{D + (N-1)D_{\max}} \to \frac{1}{D_{\max}}$$

Note that at high loads the bottleneck bounds are the limiting high bound.

The asymptotic bottleneck bound

$$\frac{1}{D_{\max}}$$

and the optimistic balanced bound intersect at $N^\dagger$ where

$$\frac{1}{D_{\max}} = \frac{N^\dagger}{D + (N^\dagger - 1)D_{\mathrm{av}}}\,.$$

So that

$$N^\dagger = \frac{D - D_{\mathrm{av}}}{D_{\max} - D_{\mathrm{av}}}\,.$$

Queueing theory

Markov processes

A *Markov process* is a family of random variables $X(t)$ indexed by a time parameter t in which

- $X(t)$ denotes the *state* at time t
- The next state depends only on the current state

We are concerned with the particular case of Markov processes in which the state space is discrete, termed *Markov chains*

If we denote successive states as x_1, x_2,... then the *Markov property* says that

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_1 = x_1, \ldots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$$

Other types of stochastic process

A Markov process is a particular kind of *stochastic process*

- *Birth death process* (BDP): a Markov process in which transitions occur only between neighbouring states
- *Random Walks*: refer to Probability IA notes
- *Renewal Process*: $X(t)$ counts state transitions in $(0, t)$

Birth death processes

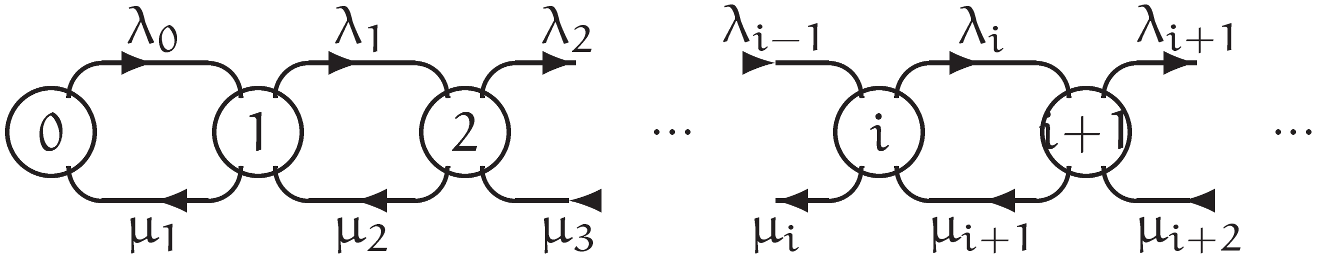
A *birth death process* is a special case of a continuous-time Markov chain in which we allow transitions only between neighbouring states.

The state space is the set of non-negative integers and we allow only a single birth or death per transition. Thus given $X_n = i$

$$X_{n+1} = \begin{cases} i+1 & \text{birth (or arrival)} \\ i-1 & \text{death (or departure)} \end{cases}$$

State space diagram

We use λ_i to represent the birth rate in state i, and μ_i to represent the death rate in state i.



Time dependent solution of BDP

We denote by $P_i(t)$ the probability of being in state i at time t.

The probability of a birth in an interval Δt when the system starts in state i is assumed to be $\lambda_i \Delta t + o(\Delta t)$.

The probability of a death in Δt when the system starts in state i is $\mu_i \Delta t + o(\Delta t)$.

The probability of more than one event in Δt is $o(\Delta t)$.

[Recall: $o(\Delta t)$ denotes a quantity which becomes negligible when compared with Δt as $\Delta t \to 0$.]

To solve for $P_i(t)$ we write a set of difference equations called the *Chapman Kolmogorov* equations.

Chapman Kolmogorov equations

For $i \geq 1$:

$$\begin{aligned} P_i(t + \Delta t) = P_i(t)&(1 - \lambda_i \Delta t)(1 - \mu_i \Delta t) \\ &+ P_{i+1}(t)(\mu_{i+1}\Delta t)(1 - \lambda_{i+1}\Delta t) \\ &+ P_{i-1}(t)(\lambda_{i-1}\Delta t)(1 - \mu_{i-1}\Delta t) \\ &+ o(\Delta t)\,. \end{aligned}$$

For $i = 0$:

$$\begin{aligned} P_0(t + \Delta t) = &P_0(t)(1 - \lambda_0 \Delta t) \\ &+ P_1(t)(\mu_1 \Delta t)(1 - \lambda_1 \Delta t) \\ &+ o(\Delta t)\,. \end{aligned}$$

Chapman Kolmogorov (2)

We derive differential-difference equations from these by dividing through by Δt and taking the limit as $\Delta t \to 0$ to get for $i \geq 1$

$$\begin{aligned} \frac{dP_i(t)}{dt} &= -(\lambda_i + \mu_i)P_i(t) \\ &+ \mu_{i+1}P_{i+1}(t) \\ &+ \lambda_{i-1}P_{i-1}(t) \end{aligned}$$

and for $i = 0$

$$\frac{dP_0(t)}{dt} = -\lambda_0 P_0(t) + \mu_1 P_1(t) \,.$$

The time dependent solution is *difficult* for many systems of interest, so we will study the stationary solution.

Stationary solution

We are interested in the long term probabilities after the system has reached an *equilibrium*.

These probabilities are independent of the initial conditions.

System reaches equilibrium if, for all i,

$$\lim_{t\to\infty} P_i(t) = p_i \quad \text{exists}\,.$$

Stationary solution (2)

The quantities p_i solve the Chapman Kolmogorov equations with $dP_i(t)/dt = 0$ so that

$$0 = -(\lambda_i + \mu_i)p_i + \mu_{i+1}p_{i+1} + \lambda_{i-1}p_{i-1}$$
$$0 = -\lambda_0 p_0 + \mu_1 p_1 \,.$$

Rewriting gives

$$p_{i+1} = \frac{\lambda_i + \mu_i}{\mu_{i+1}}p_i - \frac{\lambda_{i-1}}{\mu_{i+1}}p_{i-1} \quad (i \geq 1)$$
$$p_1 = \frac{\lambda_0}{\mu_1}p_0 \,.$$

Stochastic balance

Under steady-state conditions we require total flow into a state to equal total flow out of a state.

The total flow is the product of the steady state probabilities and the flow rates.

We enter state i at rate $p_{i-1}\lambda_{i-1} + p_{i+1}\mu_{i+1}$.

We exit state i at rate $p_i\lambda_i + p_i\mu_i$.

The equation

$$p_{i-1}\lambda_{i-1} + p_{i+1}\mu_{i+1} = p_i\lambda_i + p_i\mu_i \quad (i \geq 1)$$

equates the flow into and out of state i.

This is called the *global balance* equation.

Stochastic balance (2)

The two equations

$$p_i \lambda_i = p_{i+1} \mu_{i+1} \quad (i \geq 0)$$

describing flow from state i to state $i+1$, and

$$p_i \mu_i = p_{i-1} \lambda_{i-1} \quad (i \geq 1)$$

describing flow from state i to state $i-1$ are the *detailed balance* equations.

Rewriting gives

$$p_{i+1} = \frac{\lambda_i}{\mu_{i+1}} p_i$$

which gives us the product solution

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \quad \text{for } k \geq 1$$

for p_k $(k \geq 1)$ in terms of p_0.

Stochastic balance (3)

Since the sum of state probabilities is unity,

$$p_0 + \sum_{k=1}^{\infty} p_k = 1$$

$$p_0 + \sum_{k=1}^{\infty} p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} = 1$$

$$p_0 \left[1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right] = 1$$

so that

$$p_0 = \left[1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1}$$

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}$$

which are known as the *general flow balance equations*.

The M/M/1 queue

The BDP maps well onto our domain of study — queueing systems.

Births represent arrivals to queue, deaths represent departures as customers finish service.

The M/M/1 queue is an infinite customer system, with infinite waiting room, and a state independent service rate.

This means that $\lambda_i = \lambda$ and $\mu_i = \mu$ for all i and we can solve the balance equations as

$$\begin{aligned} p_k &= p_0 \prod_{i=0}^{k-1} \frac{\lambda}{\mu} \\ &= p_0 \left(\frac{\lambda}{\mu}\right)^k . \end{aligned}$$

The M/M/1 queue (2)

Writing $\rho = \dfrac{\lambda}{\mu}$

$$\begin{aligned} p_0 &= \frac{1}{1 + \sum_{k=1}^{\infty} \rho^k} \\ &= \frac{1}{1 + \rho \sum_{k=0}^{\infty} \rho^k} \\ &= \frac{1}{1 + \rho \left(\frac{1}{1-\rho} \right)} \\ &= 1 - \rho \end{aligned}$$

Consequently, the number in the system is geometrically distributed

$$p_k = (1 - \rho)\rho^k, \qquad k = 0, 1, 2, \ldots$$

If $\rho > 1$, that is, if $\lambda > \mu$ the system will not reach equilibrium.

The M/M/1 queue (3)

What is the average number of customers, $\mathbb{E}(N)$, in the system?

$$\begin{aligned}
\mathbb{E}(N) &= \sum_{k=0}^{\infty} k p_k \\
&= \sum_{k=0}^{\infty} k(1-\rho)\rho^k \\
&= (1-\rho)\rho \frac{\partial}{\partial \rho}\left(\sum_{k=0}^{\infty} \rho^k\right) \\
&= (1-\rho)\rho \frac{\partial}{\partial \rho}\left(\frac{1}{1-\rho}\right) \\
&= \frac{\rho}{1-\rho}\,.
\end{aligned}$$

The M/M/1 queue (4)

An arriving customer will find, on average $\mathbb{E}(N)$ in the system, and will spend a time, say $\mathbb{E}(T)$, in the system. During $\mathbb{E}(T)$ there will be, on average $\lambda\mathbb{E}(T)$ arrivals, leaving $\mathbb{E}(N)$ customers in the queue. Thus

$$\mathbb{E}(N) = \lambda\mathbb{E}(T)$$

which is Little's result restated. In our case

$$\begin{aligned}\mathbb{E}(T) &= \frac{\mathbb{E}(N)}{\lambda} = \frac{\rho}{\lambda(1-\rho)} \\ &= \frac{1}{\mu(1-\rho)} = \frac{1}{\mu-\lambda}\end{aligned}$$

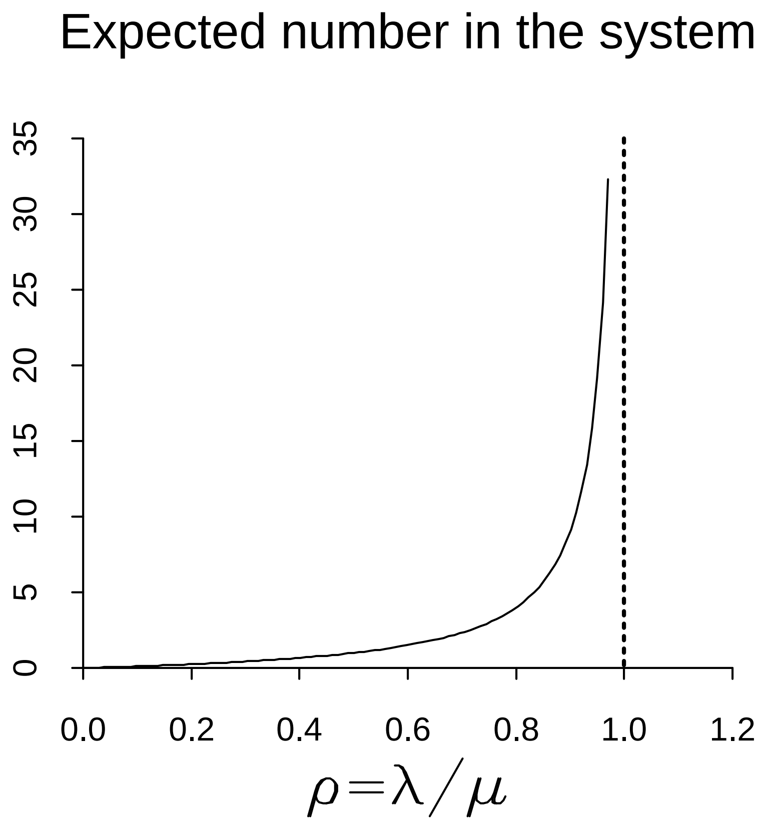
which is the M/M/1 average response time.

Note that

- $\frac{1}{\mu}$ is the average service time;
- ρ is the utilization.

Performance at high load

At high utilizations ρ approaches one and the response time and queue lengths are unbounded.



Exercise: what happens to the variance as $\rho \to 1$?

Response time

Consider the effect on response time by increasing the utilization by a constant load of 0.1.

utilization (ρ)			response time ($\mathbb{E}(T)$)		
old	new	% $\uparrow$	old	new	% $\uparrow$
0.1	0.2	100.0	1.11 $\frac{1}{\mu}$	1.25 $\frac{1}{\mu}$	13
0.5	0.6	20.0	2 $\frac{1}{\mu}$	2.5 $\frac{1}{\mu}$	25
0.8	0.9	12.5	5 $\frac{1}{\mu}$	10 $\frac{1}{\mu}$	100

Predicting response times is very difficult at high loads.

Running systems at maximum utilization may please the *providers*, but it doesn't please the *users*.

Busy period of a single server system

Viewed from the server's perspective activity alternates between busy and idle periods. If customers arrive according to a Poisson process at rate λ then the idle periods have the $\mathrm{Exp}(\lambda)$ distribution.

We have seen that the equilibrium probability that the server is idle, p_0, is given by $1 - \rho$ so that over a long period of time T the server is idle for a total duration of $T(1 - \rho)$ on average and that this corresponds on average to $T\lambda(1 - \rho)$ distinct idle periods.

Busy period of a single server system (2)

Thus there are on average $T\lambda(1-\rho)$ distinct busy periods taking a total time of $T\rho$ and so the mean length of a single busy period is

$$\frac{T\rho}{T\lambda(1-\rho)} = \frac{\rho}{\lambda(1-\rho)} \, .$$

Each customer requires a service time of μ^{-1} on average so the mean number of customers served during a busy period is

$$\frac{1}{1-\rho} \, .$$

M/M/1 — an example

H.R Cutt barber shop, no appointment needed!

Customers served FCFS.

On Saturday mornings he is very busy and is considering hiring a helper.

Measures the (Poisson) arrival rate of customers to be 5 per hour.

Customers are prepared to wait, and he spends on average 10 min per cut.

What are the average number of customers in the shop, the average number of customers waiting? What percentage of time can a customer receive a cut without waiting?

He has 4 seats in his waiting room. What is the probability that an arriving customer will find no seat and have to stand?

M/M/1 example solution

The average number of customers in the shop:
$\lambda = 5$ per hour, $\mu = 6$ per hour So

$$\rho = 5/6, \quad \text{and hence } \mathbb{E}(N) = 5\,.$$

Since the average number of customers in service is ρ, the utilization, the average number of customers waiting is

$$\mathbb{E}(N_q) = \mathbb{E}(N) - \rho = 4\frac{1}{6}\,.$$

How likely is the barber to be idle?

$$p_0 = 1 - \rho = \frac{1}{6}\,.$$

How often is no seat free?

$$\begin{aligned} \mathbb{P}(\text{no seat}) &= \mathbb{P}(N \geq 5) \\ &= \rho^5 \\ &\approx 0.402 \end{aligned}$$

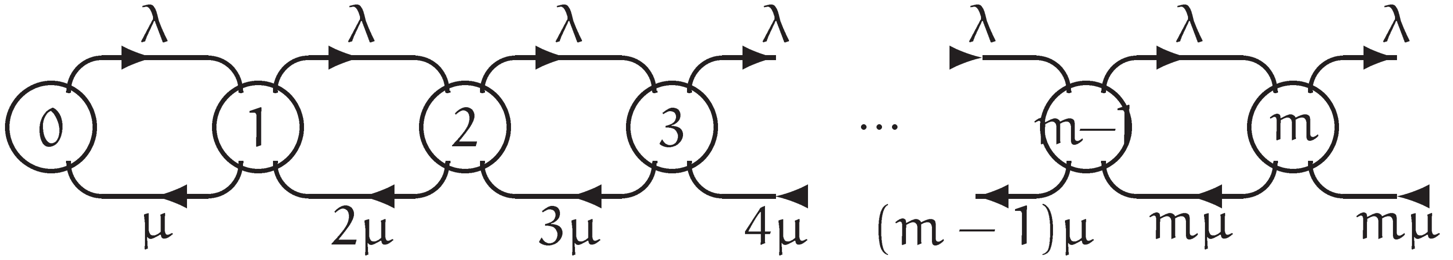
$M/M/m$ — m servers

This is just like the M/M/1 system, except that there are m servers.

For all $k, \lambda_k = \lambda$, but now the service rate is a function of the number of customers in the system

$$\mu_k = \begin{cases} k\mu & \text{if } 0 < k \le m \\ m\mu & \text{if } k \ge m \end{cases}$$

For an equilibrium distribution to exist we require that $\frac{\lambda}{m\mu} < 1$.



$M/M/1/K$ — finite capacity

The system can hold up to K customers.

Now for $k \geq K$, $\lambda_k = 0$ and for $k > K$, $p_k = 0$.

Using the equations from the $M/M/1$ queueing system, but limiting the summation, and again writing $\rho = \frac{\lambda}{\mu}$,

$$p_k = p_0 \rho^k \quad \text{for} \quad k \leq K$$

$$p_0 = \frac{1}{1 + \sum_{k=1}^{K} \rho^k} = \frac{1}{1 + \frac{\rho - \rho^{K+1}}{1-\rho}}$$

$$= \frac{1-\rho}{1-\rho^{K+1}}$$

Note that p_0 is greater than in the $M/M/1$ case.

For this system with a finite state space an equilibrium distribution always exists whatever the arrival and departure rates.

M/M/1//N — finite population

Single server, unbounded queue capacity and a population of N customers. We solve this system by modifying the λ_k to model the arrival rate. Instead of having an arrival rate for the population as a whole, we assign an arrival rate to each customer, say λ.

If there are no customers in the system, then all of them are eligible to be born, so that

$$\lambda_0 = N\lambda\,.$$

As we have more customers in the system, we have less eligible to be born. So that,

$$\lambda_k = (N - k)\lambda, \quad \text{for} \quad 0 \le k \le N\,.$$

With a single server the service rate is constant

$$\mu_k = \mu, \quad \text{for} \quad k \ge 1\,.$$

$M/M/m/m$ — m server loss system

An application of this system is to model a link in a telephone network.

Such a link contains m circuits each of which carries a single call.

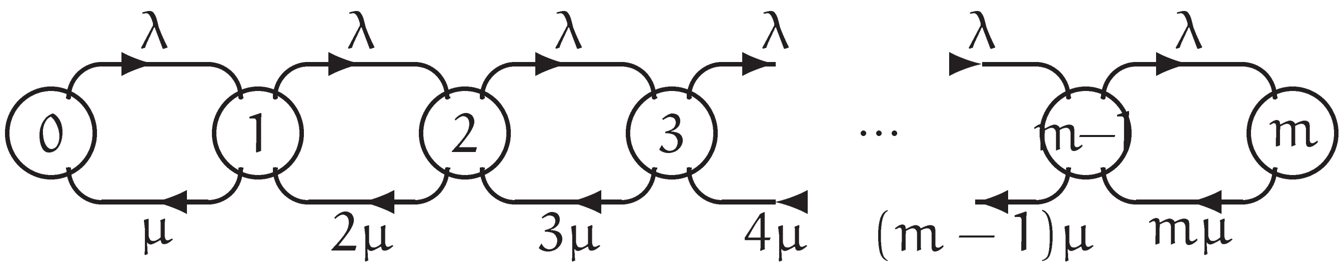
Suppose that calls arrive at the link according to a Poisson process of rate λ.

If there is a free circuit the call is connected and holds the circuit for an exponentially distributed length of time, with mean $\frac{1}{\mu}$.

At the end of this holding period the call terminates and the circuit is freed.

If there are no free circuits then the call is lost.

M/M/m/m



$$\lambda_k = \begin{cases} \lambda & k < m \\ 0 & k \geq m \end{cases}$$

$$\mu_k = k\mu \quad \text{for} \quad 1 \leq k \leq m$$

The flow balance equations give for $k \leq m$

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} = p_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu}$$

$$= p_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}$$

M/M/m/m (2)

Solving for p_0 gives

$$\sum_{k=0}^{m} p_k = p_0 \sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} = 1$$

$$\Rightarrow \quad p_0 = \left[\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}\right]^{-1}$$

The probability that an arriving call finds all circuits occupied, p_m, is called the *loss probability* for the telephone link. Thus,

$$p_m = p_0 \left(\frac{\lambda}{\mu}\right)^m \frac{1}{m!}$$

$$= \left(\frac{\lambda}{\mu}\right)^m \frac{1}{m!} \left[\sum_{k=0}^{m} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}\right]^{-1}$$

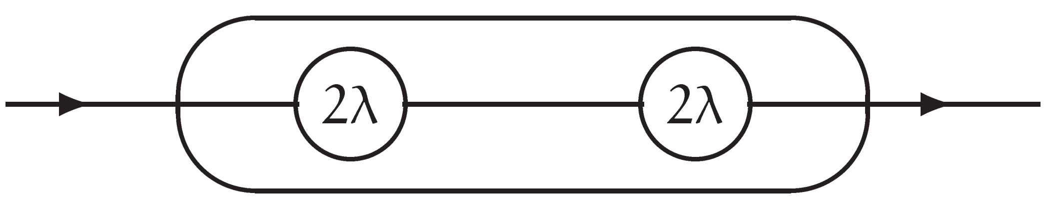
This expression for the loss probability is known as *Erlang's formula*.

BDM extensions

First we relax our constraints on the arrival process distribution.

We want to model systems in which the coefficient of variation of the interarrival time is less than one.

Consider a system in which a birth occurs in two stages.



Each stage has an exponentially distributed residence time.

BDM extensions (2)

If the desired birth rate is λ, then let each stage have a rate 2λ.

The average time to get through the combined birth process will be

$$\mathbb{E}(\tau) = \frac{1}{2\lambda} + \frac{1}{2\lambda} = \frac{1}{\lambda}\,.$$

Since each stage has exponentially distributed residence times, the variance of each stage is

$$\sigma^2_{\text{single}} = \frac{1}{(2\lambda)^2}\,.$$

The two stages are independent, so the variance of τ, the time to get through both stages is

$$\sigma^2_\tau = \frac{1}{(2\lambda)^2} + \frac{1}{(2\lambda)^2} = \frac{1}{2\lambda^2}$$

BDM extensions (3)

So the coefficient of variation is

$$C_\tau = \frac{\sqrt{\frac{1}{2\lambda^2}}}{\frac{1}{\lambda}} = \frac{1}{\sqrt{2}} .$$

In general, if we use r stages each with rate $r\lambda$ we get an average time through all stages of $\frac{1}{\lambda}$ and a coefficient of variation of $\frac{1}{\sqrt{r}}$.

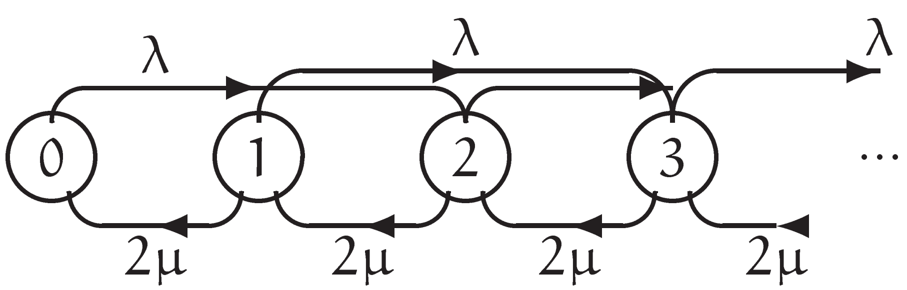
The distribution that describes this r-stage process is the Erlangian distribution, denoted E_r.

Example, $M/E_2/1$

Allow the state of the process to represent the number of stages remaining to be served.

An arrival increases the number of stages remaining to be served by 2 and occurs at rate λ.

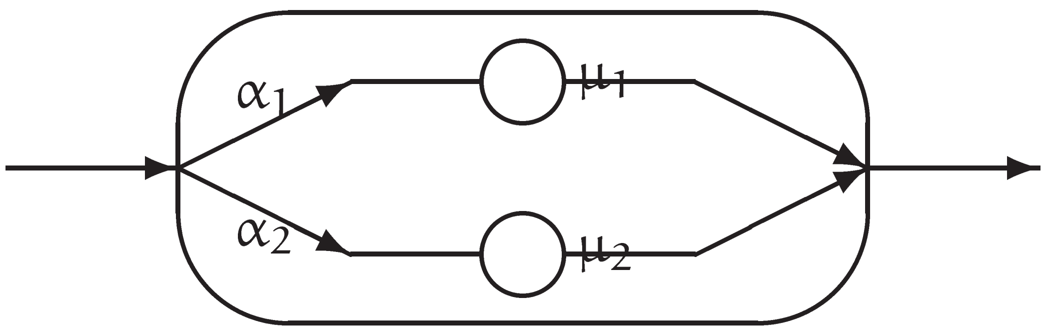
A departure from a stage reduces the number of stages to be served by 1 and occurs at rate 2μ.



Parallel Servers

Combining stages in series reduces the coefficient of variation.

If, instead, we combine them in parallel with a probability α_i of choosing the i^{th} parallel stage we get a service distribution with coefficient of variation larger than 1.



The coefficient of variation is given (see Kleinrock) by

$$C_\tau = \frac{2\sum_{i=1}^{r} \frac{\alpha_i}{\mu_i^2}}{\left(\sum_{i=1}^{r} \frac{\alpha_i}{\mu_i}\right)^2} - 1 \geq 1 .$$

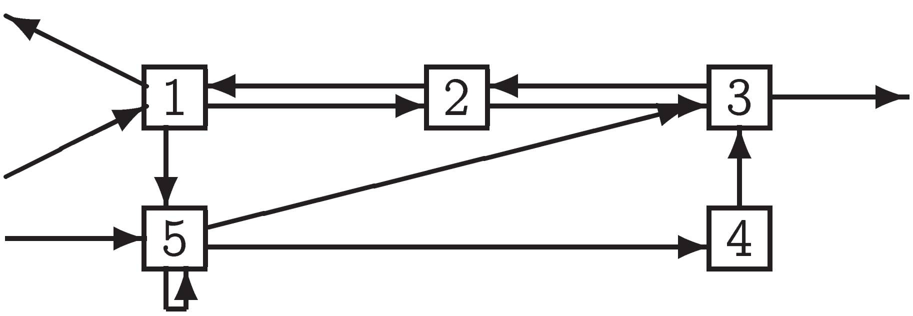
Queueing Networks

We have seen the solution of several queueing systems of interest.

In general we want to solve a system of such queues representing a real world performance problem e.g. a distributed computing system.

We represent the system under study as a network of connected queues.

Customers move (instantaneously) between service centres where they receive service.



Model definition

- *customers*: typically these represent programs or data packets etc
- *service centres*: the resources in the system e.g. disks, CPU, transmission links
- *service time distributions*: may vary according to customer type and visit
- *load dependence*: multi-processor systems have load dependent service rates
- *waiting lines and scheduling*: may have limited capacity and various scheduling algorithms
- *customer types*: multiple customer classes may exist

Open Queueing Networks

Allow customers to arrive from an external source, with rate γ.

Customers may leave the network on completion.

Assume we have N nodes, each a single server queue with infinite waiting room.

Each server i has exponential service time with mean $1/\mu_i$.

Customers arrive as a Poisson stream at node i at rate γ_i.

A customer completing at node i moves to node j with probability q_{ij} for $(i, j = 1, 2, \ldots, N)$.

Note that

$$\sum_{j=1}^{N} q_{ij} \leq 1 \,.$$

Open Queueing Networks (2)

A job leaves the network from node i with probability

$$q_{i0} = 1 - \sum_{j=1}^{N} q_{ij} \, .$$

The probabilities q_{ij} are called the *routing probabilities*.

Written as an $N \times N$ matrix this is called the routing matrix $Q = (q_{ij})$.

An open network with parameters γ_i, μ_i and Q is called a *Jackson network*.

The system state is $(k_1, k_2, \ldots, k_N)$, where k_i is the number of jobs present at node i.

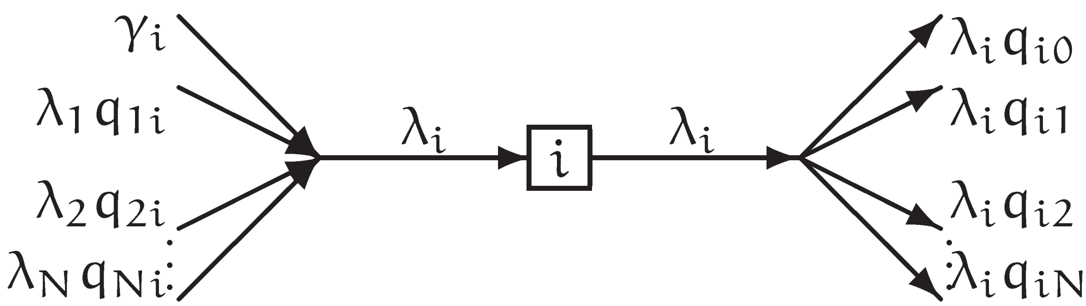
Steady state solution

Let λ_j be the average number of arrivals to node j per unit time.

On average the departure count per unit time is therefore λ_j.

A fraction q_{ji} go to node i.

The rate of traffic from node j to node i is thus $\lambda_j q_{ji}$.



Traffic equations

Adding together all contributions,

$$\lambda_i = \gamma_i + \sum_{j=1}^{N} \lambda_j q_{ji} \quad i = 1, 2, \ldots, N\,.$$

These are known as the *traffic equations*.

A necessary and sufficient condition for the existence of an equilibrium distribution is that

$$\rho_i := \frac{\lambda_i}{\mu_i} < 1$$

where λ_i is the solution of the traffic equations.

Jackson's Theorem

Let $p(k_1, k_2, \ldots, k_N)$ be the equilibrium distribution. *Jackson's Theorem* states that

$$p(k_1, k_2, \ldots, k_N) = p_1(k_1)p_2(k_2) \cdots p_N(k_N)$$

where $p_i(k_i)$ is the equilibrium distribution that there are k_i jobs in an $M/M/1$ queue with traffic intensity ρ_i.

Jackson's theorem has some important implications.

- The numbers of jobs at the various nodes are independent.
- Node i behaves as if subjected to a Poisson stream with rate λ_i.

Jackson's theorem may be generalized to the case where node i has n_i servers and so the nodes behave as independent $M/M/n_i$ queues.

Closed Queueing Networks

Frequently used to model systems at high load or where a limited, constant number of jobs is admitted to service.

No external arrivals or departures.

Now the routing probabilities satisfy

$$\sum_{j=1}^{N} q_{ij} = 1\,, \quad i = 1, 2, \ldots, N$$

The number of jobs in the system is always a constant, denoted by K.

The states of the system, described by the vector $(k_1, k_2, \ldots, k_N)$, thus satisfy the constraint,

$$\sum_{i=1}^{N} k_i = K\,.$$

Closed Queueing Networks (2)

The state space, S, is then finite. The number of states is

$$\binom{K+N-1}{N-1}$$

The traffic equations become

$$\lambda_i = \sum_{j=1}^{N} \lambda_j q_{ji}\,, \quad i = 1, 2, \ldots, N\,.$$

With a finite state space there always exists an equilibrium distribution.

Analogous to Jackson's theorem for the open network case it may be shown that

$$p(k_1, k_2, \ldots, k_N) = \frac{1}{G} r_1(k_1) r_2(k_2) \cdots r_N(k_N)$$

where $r_i(k_i)$ is the probability that there are k_i jobs in an $M/M/1$ queue with traffic intensity given by a solution to the traffic equations.

Open vs closed networks

The normalization constant G has to be determined by

$$G = \sum_{s \in S} r_1(k_1) r_2(k_2) \cdots r_N(k_N)$$

obtained by summing over all states $s = (k_1, k_2, \ldots, k_n)$ in the state space S.

With closed networks need to compute the normalization constant G — a computationally hard problem.

The constraint $\sum_{i=1}^{N} k_i = K$ means that the numbers of jobs in the various queues are no longer independent.

For instance, consider the extreme case where all K jobs are at one node.

Derivation of the size of S

We require to show that

$$p(K, N) := \binom{K + N - 1}{N - 1}$$

is the number of (ordered) partitions of a positive integer K into N integer summands

$$K = \sum_{i=1}^{N} k_i \; .$$

Proof: consider $K + N - 1$ boxes aligned in a row and select $N - 1$ of these boxes (without replacement) which can be done in $p(K, N)$ ways. Place a "/" symbol in each of the boxes and a "1" in each of the other boxes. The boxes now represent an (ordered) partition of K into N groups of "1" which when added together give the k_i summands.

Aside: application to balanced systems

Recall that in a *balanced system* with N jobs and K devices the common utilization at each device is given by

$$U_i(N) = \frac{N}{N+K-1} \qquad \text{for } i = 1, \ldots, K\,.$$

Aside: application to balanced systems (2)

But using the function $p(K, N)$ we can see that, by symmetry, the utilization is given by

$$
\begin{aligned}
U_i(N) &= 1 - \frac{p(N, K-1)}{p(N, K)} \\
&= 1 - \frac{\binom{N+K-2}{K-2}}{\binom{N+K-1}{K-1}} \\
&= 1 - \frac{(N+K-2)!}{(K-2)!N!} \frac{N!(K-1)!}{(N+K-1)!} \\
&= 1 - \frac{K-1}{N+K-1} \\
&= \frac{N}{N+K-1} .
\end{aligned}
$$

The M/G/1 queue

It is usually easier to justify the memoryless property for arrivals than for service times.

For arrivals, we can appeal to asymptotic results on the merging of large numbers of independent streams to help justify the memoryless property for arrivals.

For service times, it is easy to think of examples where the service times have a quite different distribution to the exponential. For example, the service times might be constant corresponding to certain packet lengths in a communication network.

This leads to an interest in the M/G/1 queue with general service times given by CDF $B(x) = \mathbb{P}(\text{service time} \leq x)$.

(Lack of) Markov property

With general service times we no longer find that $X(t)$, the number of customers in the system at time t, has the Markov property.

This follows since the future evolution of $X(t)$ now depends not just on the number present but on the remaining service time of the customer (if any) currently in service.

Recall, that in the $\cdot/M/\cdot$ case the remaining service time always has the same memoryless distribution whenever we observe the queue.

Embedded Markov Chain

It would be possible to formulate a model for the $M/G/1$ queue using a state variable with two components (n, x) where n is the number of present and x is the remaining service time, if any, of the customer in service. This augmented model does have the Markov property and can be analyzed directly.

Instead, it is possible to pick out a discrete set of times where the Markov property holds and build a model on this discrete time Markov Chain. Such a set of times is given by t_i $(i = 1, 2, \ldots)$ where t_i is the time of the i^{th} departure from the queue. There is no remaining service time to worry us at these time instants.

Thus, $X(t_i)$, $i = 1, 2, \ldots$ is a *Markov Chain* embedded in the stochastic process $X(t)$.

Performance measures

The determination of a full description of the M/G/1 model is possible but difficult. Instead, we shall look at some steady state performance measures.

Let $1/\mu$ be the mean service time of a customer in the M/G/1 queue, obtained from the CDF of the service time distribution $B(\cdot)$, say. Then the mean queueing time, $\mathbb{E}(T_q)$, of a customer before it receives service is given by

$$\mathbb{E}(T_q) = \mathbb{E}(N_q)\frac{1}{\mu} + \rho\mathbb{E}(R)$$

where $\mathbb{E}(N_q)$ is the average number of customers waiting in the queue at the time of arrival, $\mathbb{E}(R)$ is the average remaining service time of the customer, if any, in service and $\rho = \lambda/\mu$, the traffic intensity, gives the utilization of the server.

Remaining service time, R

A result from *renewal theory* is that $\mathbb{E}(R) = \mu\mathbb{E}(S^2)/2$.

Notice that this involves the 2nd moment, $\mathbb{E}(S^2)$ of the service time S.

For the exponential case, $\mathbb{E}(S^2) = 2/\mu^2$ so that $\mathbb{E}(R) = 1/\mu$ as might be intuitively expected (recall the Memoryless property).

Performance measures (2)

From Little's law,

$$\mathbb{E}(N_q) = \lambda\mathbb{E}(T_q)$$

and so

$$\begin{aligned}
\mathbb{E}(T_q) &= \lambda\mathbb{E}(T_q)\frac{1}{\mu} + \rho\mathbb{E}(R) \\
&= \frac{\rho\mathbb{E}(R)}{(1-\rho)} \\
&= \frac{\rho\mu\mathbb{E}(S^2)}{2(1-\rho)} \\
&= \frac{\lambda\mathbb{E}(S^2)}{2(1-\rho)} .
\end{aligned}$$

Performance measures (3)

Let C_S be the coefficient of variation of the service time distribution then

$$C_S^2 = \frac{\mathbb{E}(S^2)}{(\mathbb{E}(S))^2} - 1$$

where $\mathbb{E}(S) = 1/\mu$ so

$$\mathbb{E}(S^2) = \frac{(1 + C_S^2)}{\mu^2}$$

Hence,

$$\begin{aligned} \mathbb{E}(T_q) &= \frac{\lambda(1 + C_S^2)}{\mu^2 2(1-\rho)} \\ &= \frac{\rho(1 + C_S^2)}{2\mu(1-\rho)}\,. \end{aligned}$$

Pollaczek-Khintchine formula

Consider now the total time, $\mathbb{E}(T)$, for a customer to on average pass through the system given by their waiting time in the queue and their own service time.

Thus,

$$\mathbb{E}(T) = \mathbb{E}(T_q) + \frac{1}{\mu} = \frac{1}{\mu}\left(1 + \frac{\rho(1 + C_S^2)}{2(1-\rho)}\right) .$$

Using Little's law for the entire system we can now find, $\mathbb{E}(N)$, the mean number of customers in an $M/G/1$ queueing system by

$$\begin{aligned} \mathbb{E}(N) &= \lambda \mathbb{E}(T) \\ &= \rho + \frac{\rho^2(1 + C_S^2)}{2(1-\rho)} \end{aligned}$$

This is known as the *Pollaczek-Khintchine* (P-K) formula.

P-K formula

The Pollaczek-Khintchine formula tells us that the mean number of customers is determined not only by the mean interarrival and mean service times but also by the *coefficient of variation* of the service time distribution, C_S.

There are several cases.

- $C_S = 0$: this is the case of constant service times. For example, in ATM networks where the cells (that is, the packets) are of fixed length (53 bytes).
- $C_S < 1$: this is the case where the variability is less than in the case of exponential service times, thus the $M/M/1$ model will be conservative in its performance estimates.

P-K formula (2)

- $C_S \approx 1$: this is where the M/M/1 model works best and many systems correspond to this model. For example, batch jobs on a mainframe.
- $C_S > 1$: this is the case where the M/G/1 model is required. An example, is the observed packet lengths in Internet traffic. The distribution of packet sizes (and hence service times) is often found to be bimodal with many small packets and many longer packets of length determined by the MTU.