Applications of Computer Performance Analysis

1. **Dynamic storage allocation:** Suppose subintervals of $[0,1]$ arrive randomly to be placed in $[0,1]$. After placement, each stays for some random time and then departs, freeing up the space it was occupying. This is a mathematical model of a storage unit (the interval $[0,1]$) whose size is normalized to 1 and whose discrete structure has been replaced by a continuous relaxation. The subintervals are files that are stored for random times.

   A file of length x is placed left-justified, if possible at its time of arrival in the leftmost vacant interval (gap) at least x in length. This is called the First Fit (FF) rule. If there is no gap big enough for an arriving file, it is turned away (lost). The question is: For given probability laws describing the arrival and residence times of files, what is the long-run fraction of the arrivals that have to be turned away? (Note: Once a file is placed in storage, it can not be moved prior to departure.)

   To mount a simulation study, you must first decide on a data structure for representing the state of the storage unit $[0,1]$, which at any given time consists of an alternating sequence of gaps and strings of adjacent files. There are two operations that the data structure must accommodate efficiently: adding a new file to the state by the FF rule, and removing a randomly chosen file from the state. Design a tree data structure in which both of these operations take $O(\log N)$ time on average, where N is the known maximum number of files that will ever be stored. Specify the insertion and deletion algorithms (but don't write code).

2. **Expected value arguments and a Poisson process in two dimensions:** Consider a server polling a collection of N stations (terminals, input lines, etc.) in a strict, unidirectional, cyclic sequence

   $$1, 2, \ldots, N, 1, 2, \ldots, N, 1, 2, \ldots$$

   . Assuming N is large, we model the system by a continuous, closed tour that the server traverses endlessly in one direction, stopping to perform a service whenever a request is encountered. When not serving, the server moves at a constant speed $1/\alpha$ cycles per time unit. Service times (when the server is stopped) are i.i.d. and have a mean taken to be 1 time unit for convenience. Arrivals occur by a Poisson process in two dimensions: interarrival times are i.i.d. and exponentially distributed with parameter λ, and request locations are independently and uniformly distributed over the tour. *What is the long-run average number of requests served per cycle, and the long-run average cycle duration?*

   If C is the server cycle time in statistical equilibrium (total service time in a cycle plus the time to move around the tour), then λC is the average number of requests served in a cycle in statistical equilibrium. Each request takes an average of one time unit, and the server motion in a cycle requires α time units, so in statistical equilibrium

   $$E[C] = \alpha + \lambda E[C]$$

   and hence

   $$E[C] = \frac{\alpha}{1-\lambda}$$

   is the average cycle time, and $\frac{\lambda\alpha}{1-\lambda}$ is the average number served per cycle.

   Of course, we have assumed that the system has an equilibrium, that it settles down. This is true if and only if $\lambda < 1$, as one might expect; this inequality says that on average less than one arrival occurs during an average service time (services can be completed at a faster rate than requests for them can arrive).

Harder problems: What is the distribution of cycle times? of times waiting for the server to arrive? We return to these later.

A definitive reference: D. Kroese and V. Schmidt, "Continuous Polling System with General Service Times," *Ann. Appl. Prob.*, **2**(1992), 906.

3. **Fault tolerant computing on two processors:** We have a job of duration τ that must be run on a system of two processors, each independently subject to failures according to an exponential failure law. The failure law is operative only while a processor is actively running a job. To guard against loss of work when a failure occurs, the state of the system can be saved (checkpointed) at any time, but it requires time δ to do so, and the failure law is active during the process.

   The problem is to schedule checkpoints and processors in such a way as to maximize the probability that the job completes before both processors have failed. It is easy to see that it can never be beneficial to run the job on both processors simultaneously, so schedules look like this: For given time periods of durations $t_1, \ldots, t_k$, do the following. Run the job on one of the processors for t_1 time units. If it fails before time t_1 then run the job on the remaining processor, starting from scratch, until the job finishes or the second processor fails, whichever occurs first. If it succeeds, then checkpoint the state at time t_1, and then repeat the operation for the next time interval of duration t_2, except that if a failure occurs, and hence the job has to start running on the second machine, start the job at the point where it was checkpointed at time t_1. The only processing time on the first machine that is lost, is that which transpired since the previous checkpoint. The process continues in this way, and terminates on the first machine only if that machine does not fail during the running time τ of the job or the total checkpointing time $k\delta$.

   Normalize the rate parameter of the failure law to 1, and let $Q_k(\tau)$ denote the probability of completion under an optimal k checkpoint schedule (optimal in the sense that $t_1, \ldots, t_k$ have been chosen so as to minimize $Q_k(\tau)$). We have

$$Q_0(\tau) = 1 - (1 - e^{-\tau})^2 = 2e^{-\tau} - e^{-2\tau}, \ \tau \geq 0.$$

   and

$$Q_1(\tau) = \sup_{0 \leq x \leq \tau} [(1 - e^{-(x+\delta)})e^{-\tau} + e^{-(x+\delta)} Q_0(\tau - x)], \ \tau \geq 0 \tag{1}$$

   We find that

$$Q_1(\tau) = (1 - e^{-(\tau/2+\delta)})e^{-\tau} + e^{-(\tau/2+\delta)} Q_0(\tau/2) \tag{2}$$

   A comparison of the formulas for Q_0 and Q_1 shows that one checkpoint is better than none if and only if

$$\delta < \ln \frac{2}{1 + e^{-\tau/2}}$$

   Exercises:

   (a) For the case of no checkpoints, find expected time to complete the job, given that it does complete. Repeat for the case of exactly one checkpoint.

   (b) Prove that (1) implies (2).

   (c) Work on $Q_2(\tau)$ until you get tired.

   (d) Generalization: Introduce a constant repair time r.

4. **A Linear Stochastic Instruction Pipeline Model:** Instructions execute in at most k stages; a pipeline attempts to achieve k-fold parallelism by having k instructions in the pipeline at once, each in a different stage. In the linear model here, instructions are introduced into the pipeline at a fixed rate, start in stage 1, and, depending on the instruction, continue up to stage i for some $1 \leq i \leq k$, when they complete. For simplicity, assume that each stage takes the same amount of time, and that the instruction input rate is one instruction per stage time. In general, a state of the system just after a stage completion is a k-bit vector $(x_1, \ldots, x_k)$, $x_i \in \{0, 1\}$, $1 \leq i \leq k$, where $x_i = 1$ (or 0) means there is (or is not) an instruction beginning its i-th stage. We can define a homogeneous Markov chain on this state space with the $2^k \times 2^k$, one-step transition probability matrix $\{p_{ij}\}$, where i and j are the numerical values of the corresponding states (bit vectors).

   To model a worst-case performance, assume that all instructions take k stages and let us suppose that occasionally the sequential instruction execution has to be interrupted (by a branch instruction) and restarted from some point outside the subsequence currently in the pipeline. At these times, the pipeline has to drain completely and refill.

   Formally, the initial state is $(0, 0, \ldots, 0)$ and the only possible state transitions are

Type 1: $s \to s'$ where s (respectively s') begins with i (respectively $i+1$) 1's and ends in $k-i$ (respectively $k - i - 1$ 0's, where $0 \leq i \leq k - 1$.

Type 2: $(1, 1, \ldots, 1) \to (0, 0, \ldots, 0)$

Type 3: $(1, 1, \ldots, 1) \to (1, 1, \ldots, 1)$

   The type-1 transitions occur with probability 1. Let p (respectively $1 - p$) be the probability of type-2 (respectively type-3) transitions.

   Problem: Assuming that stages require 1 time unit, find the long-run rate (a function of k and p) at which instructions are completed by the pipeline, i.e., the rate at which transitions are made out of state $(1, 1, \ldots, 1)$. (Note that the analysis can be reduced to a two-state Markov chain, where the time to the next transition depends on the current transition.)

5. **Performance of interleaved memories:** A set of n identical, synchronous, interleaved memory units serves a queue of requests for data at locations $r_1, r_2, r_3, \ldots$. In a memory cycle the queue is scanned in FIFO order, with successive requests being accepted until a request referencing a memory module already accepted for this cycle has been encountered. At that point, the accepted requests are submitted to the memory modules with all requests being processed in parallel. Thus, if $m < n$ is the largest integer such that $r_1, \ldots, r_{m-1}$ are distinct, then the first memory cycle consists of acessing $r_1, \ldots, r_{m-1}$ in parallel. The process then repeats on the sequence $r_m, r_{m+1}, \ldots$. For present purposes, only the memory module being addressed is of interest, so suppose the r_i are integers in the set $\{1, \ldots, n\}$ and denote values of the memory-module index.

   Assume that the queue of requests is effectively infinite (e.g., there are always at least n requests waiting). We want to compute the *memory bandwidth* B_n defined as the average number of requests served in a memory cycle.

   Model 1 (Knuth and Rao): In the simplest probability model, the r_i are i.i.d. random draws from $\{1, \ldots, n\}$.

Problem: Prove that

$$B_n = \sum_{k=1}^{n} \frac{k^2(n-1)!}{(n-k)!n^k}$$

One can verify that $B_n \sim \sqrt{\pi n/2}$ as $n \to \infty$ (see the section entitled "Some asymptotic calculations" in Knuth, Vol I).

Model 2 (Burnett, Coffman, and Stone): A somewhat more realistic model gives more probability to *successor* transitions, i.e., after memory module j is referenced, module $j+1 \bmod n$ is referenced with probability $\alpha > 1/(n-1)$. Let us assume that if the successor is not referenced then the reference is equally likely to be any of the remaining $n-2$ modules.

To obtain B_n in this generalization, we write

$$B_n = \sum_{k=1}^{n} Pr\{W_n \geq k\}$$

where W_n is the number of requests up to the first repeated memory module. Observe that $W_n \geq k$ if and only if there are no duplicates among k consecutive requests. Then if $c_n(j,k)$ denotes the number of k-length sequences of distinct integers in $\{1, \ldots, n\}$ having exactly j successor transitions, we can write

$$B_n = \sum_{k=1}^{n} \sum_{j=0}^{k-1} c_n(j,k)\alpha^j \left(\frac{1-\alpha}{n-1}\right)^{k-j-1}$$

For fixed j there are $\binom{k-1}{j}$ ways of placing the j successor transitions. It is not difficult to show that the number of k-length sequences of distinct integers from $\{1, \ldots, n\}$ under one configuration of the successor transitions is the same as the number under any other. Thus, we can write

$$B_n = \sum_{k=1}^{n} \sum_{j=0}^{k-1} \binom{k-1}{j} c_n^o(j,k)\alpha^j \left(\frac{1-\alpha}{n-1}\right)^{k-j-1}$$

where $c_n^o(j,k)$ counts just those sequences where the *first* j transitions are the successor transitions.

As a final reduction, it is easy to see that $c_n^o(j,k) = c_{n-1}^o(j-1,k-1)$, $1 \leq j \leq k-1$ and hence $c_n^o(j,k) = c_{n-j}^o(0,k-j)$, and so

$$B_n = \sum_{k=1}^{n} \sum_{j=0}^{k-1} \binom{k-1}{j} c_{n-j}^o(0,k-j)\alpha^j \left(\frac{1-\alpha}{n-1}\right)^{k-j-1}$$

It remains to observe that, by the principle of inclusion-exclusion, the following formula can be proved.

$$c_m^o(0,\ell) = \sum_{i=0}^{\ell-1} (-1)^i \binom{\ell-1}{i} (m-i-1)_{\ell-i-1}$$

where $(m)_r$ is the standard notation for the falling factorial $m(m-1)\cdots(m-r+1)$. This result appears in the "problem of rencontres."

References

1. Burnett, G.J. and E. G. Coffman, Jr., "A Combinatorial Problem Related to Interleaved Memory Systems," *J. Assoc. Comp. Mach.*, **20**(1973), 39–45.
2. Stone, H. S., "A Note on a Combinatorial Problem of Burnett and Coffman," *Comm. ACM*, **17**(1974), 165–166.
3. Knuth, D.E. and G. Rao, "Activity in an Interleaved Memory," *IEEE Trans. Comp.*, **24**(1975), 943.

6. **Processor-sharing –birth-and-death equations:** In a processor sharing (PS) system, the server shares its fixed capacity equally among all jobs present in the system; there is no waiting line. However, the service received by individual jobs at any point in time is inversely proportional to the total number of jobs in the system. We assume Poisson arrivals and exponential service with the respective parameters λ and μ. Thus, if at time t, $n(t)$ jobs are in the system, then each is being served at the rate $\mu/n(t)$. A stationary distribution for the number in system exists if $\rho := \lambda/\mu < 1$; it is the same as in the $M/M/1$ system:

$$p_n = (1-\rho)\rho^n.$$

We study the conditional "delay" $X_n(\tau)$ experienced by a job requiring τ time units of service, given that it finds n in the system at its time of arrival. The corresponding total time in system is then $X_n(\tau) + \tau$. Specifically, we compute the Laplace transform

$$w_n(\tau, s) := \mathbf{E}e^{-sX_n(\tau)} \tag{3}$$

Let $\Delta\tau$ be a differential element of (service) time. We want to express the transform with service-time parameter $\tau + \Delta\tau$ in terms of the transform with parameter τ. In the usual way, this will lead to a differential equation in τ. Bear in mind that τ is *service* time, while $X_n(\tau)$ is *real* time (delay). Thus, if a job J requiring $\tau + \Delta\tau$ units of service arrives at time t to find n already in system, then it will take about $(n+1)\Delta\tau$ units of *real* time to give J its first $\Delta\tau$ units of *service* time. We say "about" because there may be small variations that occur in case of arrivals or departures. So let us examine what happens in the interval $I := [t, t + (n+1)\Delta\tau]$ following the arrival of J. By the postulates of the birth-and-death process, we need only consider cases in which at most one event occurs in I; two or more events will have probability $o(\Delta\tau)$.

At the start of I, there are $(n+1)$ jobs in total (including J), but a departure can only be one of the original n (since $\tau > 0$, we can always take $\Delta\tau$ small enough so that even if there is a departure during I, and J's service rate speeds up, J can not depart during I). But the original n receive only a fraction $n/(n+1)$ of the processing rate μ, so the probability of a departure during I is

$$\frac{n}{n+1}\mu(n+1)\Delta\tau + o(\Delta\tau) = \mu n\Delta\tau + o(\Delta\tau)$$

An arrival will take place during I with probability $\lambda(n+1)\Delta\tau + o(\Delta\tau)$, so the probability of no event in I is $1 - [\lambda(n+1) + \mu n]\Delta\tau + o(\Delta\tau)$. If no event occurs in I, then at time $t + (n+1)\Delta\tau$, J has accumulated $n\Delta\tau$ units of delay and $\Delta\tau$ units of service. Thus, in the no-event scenario, which occurs with probability $1 - [\lambda + n\mu]\Delta\tau + o(\Delta\tau)$, we express J's delay at time t as its accumulated delay of $n\Delta\tau$ during I plus the delay remaining at the end of I. That is, we write

$$X_n(\tau + \Delta\tau) = n\Delta\tau + X_n(\tau)$$

In the event of an arrival, J will be sharing the processor with $n+1$ other jobs during part of I, and there will be $n+1$ jobs in the system besides J at the end of I. But J's accumulated delay during I will stay within $O(\Delta\tau)$ of $n\Delta\tau$, and J's accumulated service will stay within $O(\Delta\tau)$ of $\Delta\tau$. (Specifically, it will be slightly less than $\Delta\tau$.) These crude estimates are all that we need in expressing the conditional delay at time t as the sum of the delay accumulated in I plus the delay remaining at time $t+(n+1)\Delta\tau$. Thus, in the case of an arrival, which occurs with probability $\lambda(n+1)\Delta\tau+o(\Delta\tau)$, it is enough to write

$$X_n(\tau+\Delta\tau) = O(\Delta\tau) + X_{n+1}(\tau+O(\Delta\tau))$$

In the case of a departure during I, J's accumulated delay will be slightly less than $n\Delta\tau$ and its accumulated service will be slightly more than $\Delta\tau$. But as above, we only need the following crude estimate. We have, with probability $\mu n\Delta\tau+o(\Delta\tau)$,

$$X_n(\tau+\Delta\tau) = O(\Delta\tau) + X_{n-1}(\tau+O(\Delta\tau))$$

In terms of the Laplace transform, we can now write, to within a $o(\Delta\tau)$ term,

$$\begin{aligned}\mathbf{E}\left[e^{-sX_n(\tau+\Delta\tau)}\right] &= (1-[\lambda(n+1)+\mu n]\Delta\tau)\mathbf{E}\left[e^{-s[X_n(\tau)+n\Delta\tau]}\right] \\ &+\lambda(n+1)\Delta\tau\mathbf{E}\left[e^{-s[X_{n+1}(\tau)+O(\Delta\tau)]}\right]+\mu n\Delta\tau\mathbf{E}\left[e^{-s[X_{n-1}(\tau)+O(\Delta\tau)]}\right]\end{aligned} \tag{4}$$

Now substitute for the second factor of the first term on the right-hand side

$$\mathbf{E}\left[e^{-s[X_n(\tau)+n\Delta\tau]}\right] = \mathbf{E}\left[e^{-sX_n(\tau)}\right]e^{-sn\Delta\tau} = \mathbf{E}\left[e^{-sX_n(\tau)}\right](1-sn\Delta\tau+o(\Delta\tau))$$

and then rearrange to obtain, to within a $o(\Delta\tau)$ term,

$$\begin{aligned}\mathbf{E}\left[e^{-s[X_n(\tau+\Delta\tau)}\right]-\mathbf{E}\left[e^{-s[X_n(\tau)}\right] &= \lambda(n+1)\Delta\tau\mathbf{E}\left[e^{-s[X_{n+1}(\tau)+O(\Delta\tau)]}\right] \\ -[\lambda(n+1)+(s+\mu)n]\Delta\tau)\mathbf{E}\left[e^{-s[X_n(\tau)]}\right] &+\mu n\Delta\tau\mathbf{E}\left[e^{-s[X_{n-1}(\tau)+O(\Delta\tau)]}\right]\end{aligned}$$

Now divide by $\Delta\tau$, take the limit $\Delta\tau\to 0$, and use the notation in (3) to obtain

$$\frac{\partial w_n}{\partial\tau} = \lambda(n+1)w_{n+1}-[\lambda(n+1)+(\mu+s)n]w_n+\mu n w_{n-1}, \tag{5}$$

for $n=0,1,2,\ldots$. There is no delay when $\tau=0$, so we have the boundary condition

$$w_n(0,s)=1$$

In the usual way, we now introduce a generating function

$$W(\tau,s,z) := \sum_{n=0}^{\infty} w_n(\tau,s)z^n$$

and obtain from (5) the first order partial differential equation

$$\frac{\partial W}{\partial\tau}-(\mu z^2-(\lambda+\mu+s)z+\lambda)\frac{\partial W}{\partial z} = (\mu z-\lambda)W \tag{6}$$

with boundary condition

$$W(0,x,z) = \frac{1}{1-z}$$

In the usual approach, this pde can be replaced by a system of ode's. After a lengthy, but routine calculation, we get

$$W = \frac{(1-\rho r^2)e^{-\lambda\tau(1-r)}}{(1-\rho r)(1-rz)-(1-r)(z-\rho r)e^{-\mu\tau(1-\rho r^2)/r}} \tag{7}$$

where r is the smaller root of the quadratic $\lambda z^2 - (\lambda+\mu+s)z + \mu$:

$$r = \frac{\lambda+\mu+s-\sqrt{(\lambda+\mu+s)^2-4\lambda\mu}}{2\lambda} \tag{8}$$

(A similar result can be obtained in terms of the larger root. Indeed, the observation that the product of the two roots is $1/\rho$ makes this result easy to find.) By the form of (6), one easily expands W in powers of z to obtain

$$w_n(\tau,s) = \frac{(1-\rho r^2)e^{-\lambda\tau(1-r)}}{1-\rho r+\rho r(1-r)e^{\mu\tau(1-\rho r^2)/r}}\beta^n, \tag{9}$$

$n = 0,1,2,\ldots$, where

$$\beta = \frac{r(1-\rho r)+(1-r)e^{-\mu\tau(1-\rho r^2)/r}}{1-\rho r+\rho r(1-r)e^{-\mu\tau(1-\rho r^2)/r}}$$

Now assume that $\rho < 1$ and hence that a stationary regime exists. Averaging over the stationary distribution of the number in system, $p_n = (1-\rho)\rho^n$, is quite easy. Indeed, by the definition of the generating function, it has effectively been done for us; we need only substitute ρ for z in (7) and multiply by $(1-\rho)$ to get

$$w(\tau,s) = \frac{(1-\rho)(1-\rho r^2)e^{-\lambda\tau(1-r)}}{(1-\rho r)^2-\rho(1-r)^2e^{-\mu\tau(1-\rho r^2)/r}} \tag{10}$$

We can make a couple of checks on this result. By the definition of the Laplace transform, we must have $w(\tau,0) = 1$, which can be verified by inspection in (10), once we see that $r \to 1$ as $s \to 0$ from (8). In the limit $s \to \infty$, we must obtain the probability mass at the origin, $Pr(X=0)$, which is the product of the probability of finding an empty system and the probability that there are no arrivals in τ time units. From (8) one sees that $r \to 0$ as $s \to \infty$, and so one indeed finds, again by inspection of (10), that

$$\lim_{s\to\infty} = (1-\rho)e^{-\lambda\tau}$$

The first two derivatives give the mean and variance of the stationary delay distribution:

$$\mathbf{E}X = -\lim_{s\to 0}\frac{\partial W}{\partial s} = \frac{\rho\tau}{1-\rho}$$

and

$$\begin{aligned}\mathrm{Var}(X) &= \lim_{s\to 0}\frac{\partial^2 W}{\partial s^2} - (\mathbf{E}X)^2 \\ &= \frac{2\rho\tau}{\mu(1-\rho)^3} - \frac{2\rho}{\mu(1-\rho)^4}\left(1-e^{-(1-\rho)\mu\tau}\right)\end{aligned}$$

We see that the expected total time in system is $\tau + \mathbf{E}X = \tau/(1-\rho)$, which checks with expected-value arguments based on the round-robin scheduling rule. Similar, but much more complicated results can be obtained in the same way from the Laplace transform (9) for the conditional distribution given the number in system at arrival. For this case, the expressions for the moments will be analytic for *all* $\rho \geq 0$.

7. **Disk head motion - a Markov chain with a continuous state space:** Let $X_1, X_2, \ldots, X_i, \ldots$ be a sequence of disk requests, each specifying a cylinder (track). We assume that the number of cylinders is large enough to justify the following continuous relaxation. The unit interval represents the set of cylinder addresses, and the X_i are i.i.d. continuous random variables with the common density function $f(x)$ defined on $[0,1]$. The i-th disk access consists of moving the disk head from the point of the $(i-1)$-st request to the point of the i-th request, i.e., from X_{i-1} to X_i. The requested read or write operation is then performed. We are interested here only in describing the head motion between successive requests. Let Z_i be the distance traveled between the $(i-1)$-st and i-th requests. Clearly, $\{Z_i\}$ is not a Markov chain, but $\{X_i, Z_i\}$ is; in fact, $Z_{i+1} = |X_{i+1} - X_i|$. We have

$$g_i(z)dz = \int_{x=0}^{1} g_i(z|x)f(x)dx \cdot dz$$

for the probability that Z_i falls in $[z, z+dz]$. The X_i are i.i.d. so the stationary regime is reached after the first step. We will drop the subscripts and focus on the density $g(z)$ for the head motion following the first request.

   (a) Verify the relation

$$g(z)dz = \int_{x=0}^{1-z} f(x+z)f(x)dx \cdot dz + \int_{x=z}^{1} f(x-z)f(x)dx \cdot dz$$

   Then specialize to the uniform law $f(x) = 1,\ 0 \leq x \leq 1$, and find both $g(z)$ and its first moment.

   (b) Generalize to the triangular density on $[0,1]$ symmetric about $1/2$ with $f(1/2) = 2$. Compare with the previous result.

   (c) Generalize to a lookahead model that allows you to process the requests two at a time; in each pair the one closer to the current head position is processed first. Assume the uniform law for $f(x)$.

   (d) Revert to a discrete formulation and generalize to a "locality" model. That is, devise a model that reflects the fact that a cylinder reference tends to be close to the preceding reference.

8. **Continuous polling revisited:** Return to application 2 and ask: What is the distribution of the number served in a cycle? Here, a cycle can be defined as the time between successive visits to some fixed but arbitrary point on the tour. *Recall that the tour has unit length and that λ is the intensity of the two-dimensional Poisson arrival process. We restrict ourselves to the case of unit service times.*

   Let $\omega = (x_1, \ldots, x_n),\ 0 < x_1 < \ldots < x_n \leq 1$, be a state at the end of a cycle, where x_1 is the distance traveled since the last customer service, x_2 is the distance traveled since the second to the last customer service, $\ldots$ during the previous cycle. Let $p_n(x_1, \ldots, x_n)dx_1 \ldots dx_n$ be the probability that n were served in the preceding cycle and the services were performed at distances in $x_i, x_i + dx_i,\ 1 \leq i \leq n$

behind the server. Note that $p_n(x_1, \ldots, x_n)$ is a mass function in n and a continuous density in $(x_1, \ldots, x_n)$.

Consider the possible state transitions while the server is moving over a small distance δ of the tour, say over the interval $(y, y+\delta)$. If we ignore transitions having probability $o(\delta)$, we see that a transition to state $(x_1+\delta, \ldots, x_n+\delta)$ can be from two possible states.

(a) The original state is $(x_1, \ldots, x_n)$ with probability density $p_n(x_1, \ldots, x_n)$, if there has been no arrival in $(y, y+\delta)$ during the last cycle, an event of probability $1 - \lambda(n+\alpha)\delta$.

(b) The original state is $(x_1, \ldots, x_n, x_{n+1})$ with $1-\delta \le x_{n+1} \le 1$ and event with probability density $p_{n+1}(x_1, \ldots, x_n, 1)\delta$. In this case, after the server moves a distance δ, the customer originally at x_{n+1} no longer belongs to the present cycle.

Then the densities $p_n(\omega)$ must satisfy

$$p_n(x_1+\delta, \ldots, x_n+\delta) = [1-\lambda(n+\alpha)\delta]p_n(x_1, \ldots, x_n) + \delta p_{n+1}(x_1, \ldots, x_n, 1)$$

and hence, after rearrangement and passage to the limit $\delta \to 0$,

$$\sum_{i=1}^{n} \frac{\partial}{\partial x_i} p_n(x_1, \ldots, x_n) = -\lambda(n+\alpha)p_n(x_1, \ldots, x_n) + p_{n+1}(x_1, \ldots, x_n, 1) \tag{11}$$

A boundary condition arises out of a third transition

$$(x_2, \ldots, x_n) \to (\xi, x_2+\delta, \ldots, x_n+\delta)$$

with $\xi \le \delta$ which occurs when a customer arrives in $(y, y+\delta)$, during the cycle just ended. This event has probability density $p_{n-1}(x_2, \ldots, x_n)\lambda(\alpha+n-1)\delta$. The boundary condition is thus

$$p_n(0, x_2, \ldots, x_n) = \lambda(\alpha+n-1)p_{n-1}(x_2, \ldots, x_n)$$

The solution to (11) with this boundary condition is

$$p_n(x_1, \ldots, x_n) = \alpha(\alpha+1)\cdots(\alpha+n-1)\lambda^n(1-\lambda)^\alpha \tag{12}$$

(In fact, this can be shown to be the unique stationary density.) The number served in a cycle thus has the distribution

$$\begin{aligned} p_n &= \int_{0<x_1<\cdots<x_n\le 1} p_n(x_1, \ldots, x_n)dx_1\cdots dx_n \\ &= \frac{\alpha(\alpha+1)\cdots(\alpha+n-1)\lambda^n(1-\lambda)^\alpha}{n!} = \binom{-\alpha}{n}(-\lambda)^n(1-\lambda)^\alpha. \end{aligned} \tag{13}$$

Note that the conditional density of the state given the number served is $p_n(x_1, \ldots, x_n)/p_n = n!$, which is the density of the order statistics of n i.i.d. uniform random draws from $[0,1]$. *Thus, in a cycle that served n customers, the customer locations were distributed as n i.i.d. uniform random draws from the length of the tour.*

There are various other interesting ways of defining cycles. For example, we could ask for the number served in cycles ending with a service (this last customer served being included in the count). The conditional state density given a service at $x_1 = 0$ is

$$\frac{p_n(0, x_2, \ldots x_n)}{\sum_{n\geq 1} \int_{0<x_2<\cdots<x_n\leq 1} p_n(0, x_2, \ldots x_n) dx_2 \cdots dx_n}$$

The denominator evaluates to $\mathbf{E}n$, so the conditional densities are

$$\frac{p_n(0, x_2, \ldots x_n)}{\mathbf{E}n} = \frac{n! p_n}{\mathbf{E}n}$$

Integrating over $0 < x_2 < \cdots < x_n \leq 1$, a region of volume $1/(n-1)!$, gives the desired result

$$\tilde{p}_n = \frac{n p_n}{\mathbf{E}n}$$

The cycle containing a random time t in statistical equilibrium is also of interest. Using the same arguments underpinning the formula for expected residual lifetimes, one obtains

$$\hat{p}_n = \frac{(n+\alpha) p_n}{\mathbf{E}(n+\alpha)}$$

Problems:
1. Prove that the mean of $\{p_n\}$ as computed from (13) has the same formula as found earlier by expected value arguments.

2. Give an argument supporting the result for $\hat{p}_n$; the argument may be informal, but it must be convincing.

3. Prove that the time spent waiting for the server to arrive in statistical equilibrium has the distribution

$$F_w(t) = \frac{1-\lambda}{\alpha} \left[t - \sum_{n=0}^{[t-\alpha]} (t - \alpha - n) p_n \right]$$

4. The analysis fails when we assume a general service time distribution G with mean 1. The question is why? Answer this question by considering just the transition from state (x) (only one customer was served in the preceding cycle) to state $(x+\delta)$. Note that such a transition can take place only if there is no arrival in $[y, y+\delta]$ during the preceding cycle. This means that there was no arrival during the cycle's single service time, say Z, which took place when the server still had distance x to travel before the end of the cycle. Does Z have the distribution G?

9. **The feedback (FB) running-time priority queue:** Let jobs arrive by a rate-λ Poisson process and let service times be exponentially distributed with parameter μ. As usual, the interarrival and service times are independent i.i.d. sequences, and we assume $\mu > \lambda$ so that a stationary regime is sure to exist. Let q be some fixed positive constant (quantum size). The queue is organized into levels numbered $1, 2, \ldots$, each being served FCFS. Arriving jobs are placed at the end of the level-1 queue. Jobs are allocated service q secs at a time by a single server: The next job, say J, selected for service is taken from the head of the lowest numbered nonempty queue level. If J finishes within q secs, it

departs from the system; otherwise, after receiving its q secs, it is placed at (fed back to) the end of the next higher queue level.

We wish to calculate W_k, the expected waiting time in queue (service time not included) of a job requiring between $(k-1)q$ and kq secs of service. For such a job, say J, write

$$W_k = \mathbf{E}T_1 + \mathbf{E}T_2, \tag{14}$$

where T_1 is the time spent waiting for jobs in system at J's time of arrival, and T_2 is the time spent waiting for jobs arriving after J. To calculate $\mathbf{E}T_1$, we transform the FB queue into a conventional nonpreemptive priority queue as follows. The first k queues of the FB system are replaced by a single FCFS $M/G/1$ queue with arrival rate λ and service times S_{kq} distributed as $\min\{kq, S\}$ where $S = S_\infty$ has the exponential distribution with parameter μ. The remaining queues in the FB system are replaced by standard priority queues with levels $2, 3, \ldots$ of the new system corresponding to levels $k+1, k+2, \ldots$ of the old system; they have the same respective arrival rates $\lambda e^{-\mu kq}, \lambda e^{\mu(k+1)q}, \ldots$ and service times distributed as S_q. Call the new system the FB$_k$ system. The key observation is that $\mathbf{E}T_1$ is the expected waiting time in the first queue of the FB$_k$ system, i.e., the latter is equal to the expected time J spends waiting for jobs in the FB system at J's arrival. To see this, we note first that, as far as J is concerned, it is immaterial whether the first k quanta are meted out to the jobs ahead of J all at once, or distributed as in the FB algorithm. Second, lower priority arrivals are Poisson in the FB$_k$ system but not in the FB system. However, it is only important that the rates are the same, insofar as higher priority waiting times are concerned. Any discernible effect of non-Poisson arrivals could only exist in the calculation of residual service times, but as an analysis like that in the notes will show, there can be no such effect so long as rates are held the same.

By these observations and Equations (79) and (84) of the notes, we can write

$$\mathbf{E}T_1 = \frac{(\lambda/2)[\mathbf{E}S_{kq}^2 + \sum_{j\geq k+1} e^{-\mu jq}\mathbf{E}S_q^2]}{1-\rho_k} \tag{15}$$

where $\rho_k = \lambda \mathbf{E}S_{kq} = \rho(1-e^{-\mu kq})$

Coming back to T_2, we return to the FB system and note that the arrivals during $W_k + (k-1)q$ must be given up to $k-1$ quanta before J departs, so

$$\mathbf{E}T_2 = \lambda[W_k + (k-1)q]\mathbf{E}S_{(k-1)q}$$

Substituting (15) and the above into (14) and solving for W_k gives

$$W_k = \frac{(\lambda/2)[\mathbf{E}S_{kq}^2 + \gamma_k \mathbf{E}S_q^2]}{[1-\rho_k][1-\rho_{k-1}]} + \frac{\rho_{k-1}}{1-\rho_{k-1}}(k-1)q \tag{16}$$

where

$$\gamma_k := \sum_{j\geq k} e^{-\mu jq} = \frac{e^{-\mu kq}}{1-e^{-\mu q}}$$

Exercise: Verify that, as $k \to \infty$, $q \to 0$ with $t = kq$ held fixed, we obtain for the mean time in system

$$W_t^* = \frac{\lambda \mathbf{E}S_t^2}{2[1-\rho(1-e^{-\mu t})]^2} + \frac{t}{1-\rho(1-e^{-\mu t})}$$

Describe in words the dynamics of this limiting FB queue.

10. **Parallel streaming and processing**: Typically, the data processed by a program are loaded (streamed) into storage before processing begins. One can save time if the processing begins when, or shortly after, the data begin to stream in. Whenever reference to a datum not in storage in made (i.e., a fault occurs), the processing is suspended until that datum has been loaded, at which point processing continues.

    To model this parallel streaming and processing, we adopt the continuous approximation to storage references, and we take the time to stream in the entire data set as the time unit. Assume for simplicity that the total processing time exceeds the total streaming time. Define $s(t) = \alpha t$, where α is the constant streaming rate and where $[0, s(t)]$ is the initial portion of the data in storage at time t. A sample path of the parallel streaming and processing is a partition of $[0, 1]$ into an alternating sequence $W_1, X_1, W_2, X_2, \ldots$ with $\sum(W_i + X_i) = 1$, where X_i is the i-th interval of processing and W_{i+1} is the time spent waiting for the datum that created the i-th fault.

    Assume that data references are made by a rate-λ Poisson process and let $H(t)$ be the stationary probability that a datum in $[0, s(t)]$ is referenced. Suppose a corresponding density $h(t) = H'(t)$ exists. Then a fault takes place in $[t, t + \Delta t]$ with probability $\lambda_t \Delta t + o(\Delta t)$, where

$$\lambda_t = \lambda[1 - H(t)] \tag{17}$$

    is the *fault rate*. The *resumption rate* is given by

$$\mu_t = \frac{h(t)}{1 - H(t)} \tag{18}$$

    so that $\mu_t \Delta t + o(\Delta t)$ is the conditional probability that at time t in a wait interval the next reference is to a datum in $[s(t), s(t) + \Delta t]$, given that the reference is for a datum in $[s(t), s(1)]$. Note that μ_t is a hazard rate.

    The expectation of the amount T of parallelism (simultaneous streaming and processing) is $\mathbf{E}T = \mathbf{E}\sum X_i$. To find this quantity, define $P(t)$ as the probability that t falls in a processing interval. Since $P(t) = \mathbf{E}\mathbf{1}_{\{B(t)=1\}}$ where $B(t)$ is 1 or 0 according as processing is or is not in progress at time t, respectively, we have

$$\mathbf{E}T = \int_0^1 P(t)dt \tag{19}$$

    An analysis, which should be standard by now, gives

$$P(t + \Delta t) = (1 - \lambda_t \Delta t)P(t) + \mu_t \Delta t[1 - P(t)] + o(\Delta t)$$

    and hence the differential equation

$$P'(t) + (\lambda_t + \mu_t)P(t) = \mu_t$$

    The solution satisfying $P(0) = 1$ is

$$P(t) = e^{-\int_0^t (\lambda_\tau + \mu_\tau)d\tau} \left[1 + \int_0^t \mu_\tau e^{\int_0^\tau (\lambda_x + \mu_x)dx} d\tau\right] \tag{20}$$

    with λ_t and μ_t given in terms of $H(t)$ by (17) and (18).

    To illustrate the possible effects of the reference distribution $H(t)$, define the class of distributions

$$H(t) = 1 - (1 - t)^a, \quad a > 0,$$

with

$$h(t) = a(1-t)^{a-1}$$

Substituting into (17) and (18), we get

$$\begin{aligned}\lambda_t &= \lambda(1-t)^a \\ \mu_t &= \frac{a}{1-t}\end{aligned}$$

in which case, (20) becomes

$$P(t) = (1-t)^a e^{\lambda(1-t)^{a+1}/(a+1)} \left[e^{-\lambda/(a+1)} + \int_0^t \frac{a(a+1)}{1-\tau} e^{-\lambda(1-\tau)^{a+1}/(a+1)} d\tau \right]$$

whereupon substitution into (17) and considerable manipulation gives the relatively simple expression

$$\mathbf{E}T = \int_0^1 e^{-\lambda x^{a+1}/(a+1)} dx$$