CS 547 Lecture 19: M/G/1 Variations

Daniel Myers

This lectures deals with some variations to the basic M/G/1 model, and their applications to real systems.

Approximating a G/G/1 Queue

The G/G/1 queue has both general service times and a general arrival process. Typically, the arrivals are characterized in terms of the mean interarrival time, $\overline{a}$, and the coefficient of variation in the interarrival times, c_a^2.

You might expect G/G/1 to be an important model, because it could allow us to model a greater variety of systems than M/G/1. While G/G/1 does have its important applications, it's significantly more difficult to analyze than the M/G/1 system, so there are fewer good results available, and the results that do exist are typically only approximations. Therefore, we prefer to use the M/G/1 model whenever it's reasonable, even if the arrivals are not perfectly Poisson.

Nonetheless, it's sometimes necessary for us to work with systems that have arrival processes that are burstier than a Poisson process. In these cases, we can use a simple modification of the basic M/G/1 equations to gain some increased accuracy without going to full-scale G/G/1 approximations.

The G/G/1 approximation incorporates the interarrival coefficient of variation, c_a^2, into the M/G/1 residence time calculation.

$$\overline{R} \approx \overline{s} + \frac{\overline{s}U(c_a^2 + c_s^2)}{2(1-U)}$$

If we have Poisson arrivals, then the interarrival times are exponentially distributed and $c_a^2 = 1$, which yields the same equation we've already seen for M/G/1.

This approximation is not guaranteed to work, but it's worth trying if you feel that the arrival process is meaningfully non-Poisson. It's also good to show others that you're aware of the existence of non-Poisson arrivals and can bend your models to account for them.

PowerNap

This section is based on the paper "PowerNap: Eliminating Server Idle Power" by Meisner, Gold, and Wenisch.

The PowerNap authors are concerned with reducing energy consumption in a large-scale datacenter. There have been many proposed strategies for reducing power consumption, but the PowerNap proposal is one of the simplest: when a server has no more work to do, turn it off, then turn it back on when a new job arrives.

A consider a server with three modes: ON, IDLE, and OFF. When a server is ON, it draws full power and can perform useful work. When a server is IDLE, it consumes about 60% of its full power. Servers can transition between IDLE and ON instantly. When a server is OFF, it draws no power, but transitioning from OFF to ON requires a startup period where the server draws full power, but cannot actually serve any jobs.

There are two basic strategies.

- ON/IDLE: when a server's queue is empty, switch to IDLE mode; switch back to ON when a new job arrives
- ON/OFF: turn OFF when a server's queue is empty; when a new job arrives turn back ON, wait for the setup period, then serve the new job

The goal of the PowerNap paper is to compare the performance of these two strategies. To capture the effect of both power consumption and residence time, they use

$$\eta = \frac{1}{\overline{R} \cdot E[Power]}$$

as their preferred statistic. The key problem is to determine the values of $\overline{R}$ and $E[Power]$ for both the ON/IDLE and ON/OFF strategies.

ON/IDLE

We can analyze the ON/IDLE strategy using a basic M/G/1 queue.

The server can transition from IDLE to ON instantly, so jobs never suffer a waiting time penalty. The average residence time is simply

$$\overline{R} = \overline{s} + \frac{\overline{s}U(1 + c_s^2)}{2(1 - U)}$$

The server draws full power whenever it is serving a job and IDLE power whenever it isn't serving. The fraction of time the server is busy with a job is simply the utilization U, so

$$E[Power] = UP_{ON} + (1 - U)P_{IDLE}$$

ON/OFF

Analyzing the ON/OFF strategy is more complicated, because we must account for the time spent in setup when the server transitions from OFF to ON. Suppose that the setup time is governed by a random variable I (for *initialization*). We can then model the behavior of the ON/OFF strategy using an M/G/1 queue "with exceptional first service". In this model, the first customer arriving to an empty queue has its service time increased by the random intialization time I.

Analyzing this queue is non-trivial, because there is a complicated interaction between the setup period and residence time. Jobs can continue to arrive during the setup period, which means they must wait for the setup period to end. Once setup ends and the server begins processing jobs, there may be a backlog of jobs built up, so new jobs that arrive shortly after the setup period ends experience an increased residence time. The key result was derived by Welch in 1964.

$$\overline{R} = \overline{s} + \frac{\overline{s}U(1 + c_s^2)}{2(1 - U)} + \frac{2E[I] + \lambda E[I^2]}{2(1 + \lambda E[I])}$$

Note that the first part of this result is simply the expected residence time in a normal M/G/1 queue. $E[I]$ and $E[I^2]$ are the first and second moments of the random setup time.

We still need to find an expression for the average power consumption in the ON/OFF strategy. Let U_{setup} denote the fraction of time the server is busy with setup or serving jobs.

$$E[Power] = U_{setup}P_{ON} + (1 - U_{setup})0$$

Therefore, if we can find an expression for U_{setup}, we have the solution.[1]

Unfortunately, we can't simply calculate U_{setup} as $\lambda\overline{s}$, as we'd normally do with the Utilization Law, but we can still apply Little's result to derive the expression we need.

Let $\overline{s}_{setup}$ denote the average time a customer spends in service, including the extra setup time for customers that arrive to an empty queue. With probability $1 - U_{setup}$ the queue is empty at an arrival instant, so

$$\overline{s}_{setup} = \overline{s} + (1 - U_{setup})E[I]$$

Now, use Little's result on the server.

$$\begin{aligned} U_{setup} &= \lambda\overline{s}_{setup} \\ &= \lambda(\overline{s} + (1 - U_{setup})E[I]) \\ &= \frac{\lambda E[I] + \lambda\overline{s}}{1 + \lambda E[I]} \end{aligned}$$

Now, the expected power consumption of the ON/OFF strategy is

$$E[Power] = \frac{\lambda E[I] + \lambda\overline{s}}{1 + \lambda E[I]} P_{ON}$$

The PowerNap uses these equations to compare the performance of ON/IDLE and ON/OFF for different loads and average setup times.

Busy Periods

This is extra material that was not covered in class.

A *busy period* starts when a customer arrives to an empty queue and ends when a departing customer leaves behind an empty queue. We can also derive the expression for U_{setup} by reasoning about the length of the busy period, denoted $\overline{B}$

Busy periods are complex, because they're recursive. When a job arrives to an empty queue, the busy period includes the service time for that job. It also includes the service times of any jobs that arrive during the first job, and any jobs that arrive during any of *those* jobs, and so forth.

Suppose the first job has a service time of w. On average, we'd expect λw jobs to arrive during the first job's service time. The expected time required to serve all of those jobs is $\lambda w\overline{s}$. Because jobs continue to arrive at rate λ, we'd expect $\lambda(\lambda w\overline{s})$ more jobs to arrive while we're serving the jobs that arrived during the

[1] The method shown in this section is not exactly what they did in the paper. Their model is assumes a mandatory shutdown and powerup time indepednent of the setup period, so the result is a little more complex. The result here is my own derivation for the utilization of the M/G/1 queue with exceptional first service.

first job. Extending this idea into an infinite sum,

$$\begin{aligned}
\overline{B} &= w + \lambda\overline{s}w + (\lambda\overline{s})^2 w + (\lambda\overline{s})^3 w + \dots \\
&= \sum_{i=0}^{\infty} (\lambda\overline{s})^i w \\
&= \frac{w}{1-U}
\end{aligned}$$

Therefore, if a busy period starts with a job of length w, we expect it to last for $\frac{w}{1-U}$. If we consider a job of average length, we get the average busy period in the M/G/1 queue.[2]

$$\overline{B} = \frac{\overline{s}}{1-U}$$

Now consider the M/G/1 queue with setup. The busy period starts with a job of length $\overline{s} + E[I]$. Therefore, the expected busy period length is

$$\overline{B}_{setup} = \frac{\overline{s} + E[I]}{1 - \lambda\overline{s}}$$

When the queue finally becomes empty, we expect to wait $\frac{1}{\lambda}$ until the next arrival. The fraction of time the queue is busy is

$$U_{setup} = \frac{\overline{B}_{setup}}{\overline{B}_{setup} + \frac{1}{\lambda}}$$

Simplifying this expression yields the same result we derived in the previous section.

[2]Funny how this equation keeps coming up, isn't it?