Seven Insights Into Queueing Theory:

Even if you don't understand the math of queueing theory you can still learn from it.

By: Bob Wescott

Article Summary

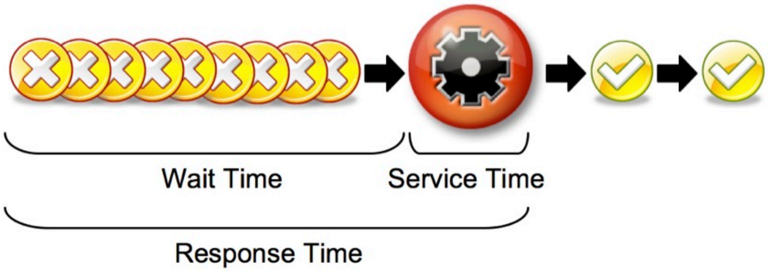
Queuing theory provides a way to predict the average delay at a service center when the arrival rate of work is greater than the throughput of completed work. The calculations are complex, but luckily we can often ignore the math and focus on the seven insights this branch of mathematics can bring to performance work.

This is an excerpt (with modifications) from: ***The Every Computer Performance Book***

Defining a Few Terms First

Service Center, Service and Response Time

A service center is where the work gets done. CPUs, processes, and disks are examples of service centers. To accomplish a given task, it is generally assumed that it takes a service center a fixed amount of time – the **service time**. In reality this assumption is usually false, but still very useful. If work arrives faster than it can be processes a queue builds and the average response time grows.

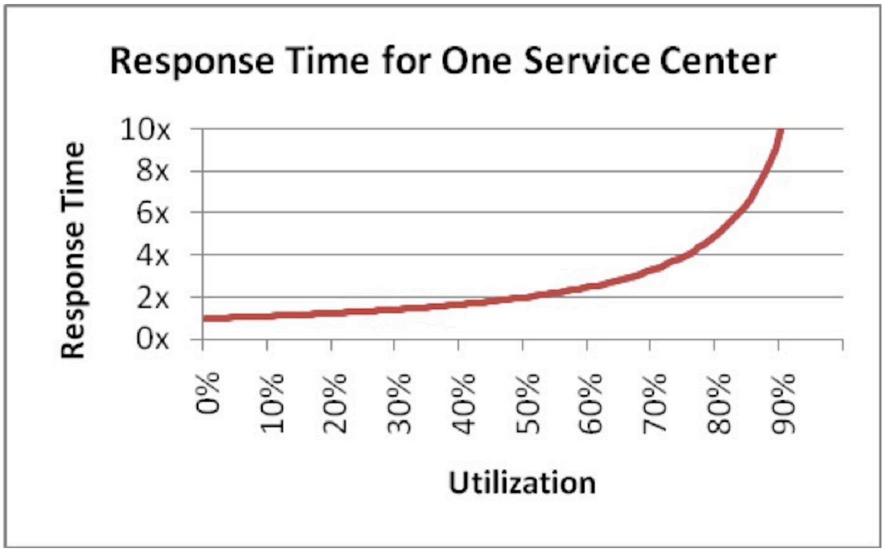


Queuing Theory

As the utilization of a service center grows, it becomes more likely that a newly arriving job will have to wait because there are jobs ahead of it. In general, the response time degradation is more pronounced the busier the resource is. An approximate formula that describes this relationship is:

$$\text{ResponseTime} = \text{ServiceTime} / (1 - \text{Utilization})$$

The real insight comes from looking at the graph of this function below, as the utilization is goes from 0% to 90%.



Notice that response time starts out as 1x at idle. At idle the response time always equals the service time as there is nothing to wait for.

Notice that the response time doubles when the service center gets to 50% utilization. At this point, sometimes the arriving jobs finds the service center idle, sometimes they find it with several jobs already waiting, but the effect on the average job is to double the response time as compared to an idle service center. The response time doubles to 4X when the service center is at 75% utilization and doubles again to 8x at around 87% utilization. Assuming you kept pushing more work at the service center, the response time doublings keep getting closer and closer (16x at 94% utilization, 32x at 97% utilization) as the curve turns skyward. All these doublings are created by the fact that the service center is busy, and thus there will often be many jobs waiting ahead of you in the queue.

Insight #1:

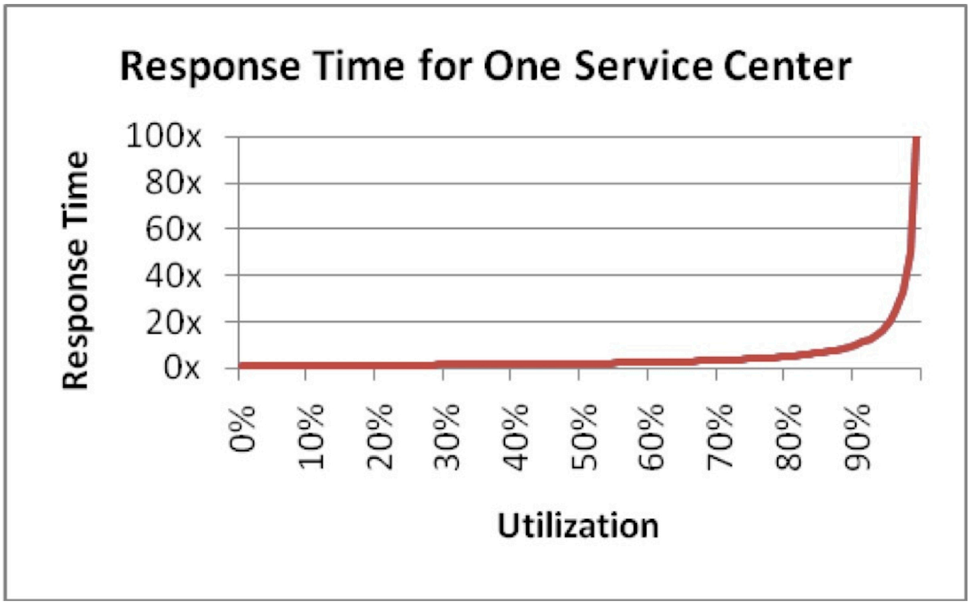
The slower the service center, the lower the maximum utilization you should plan for at peak load. The slowest computer resource is going to contribute the most to overall transaction response time increases. Unless you have a paper-tape reader as part of your transaction path, the slowest part of any computer in the early part of the twenty-first century is the rotating, mechanical magnetic disks. At the time of this writing, on an average machine, fetching a 64 bit word from memory was ~50,000x faster than getting it off disk.

The first doubling of response time comes at 50% busy and that is why conventional wisdom shoots for the spinning magnetic disks to be no more than 50% busy at peak load. Think about it this way, if the boss insists that you run the disk up to 90% busy then the average response time for a disk read will be about 10X slower than if the drive was idle. Ouch!

Insight #2:

It’s very hard to use the last 15% of anything. As the service center gets close to 100%

utilization the response time will get so bad for the average transaction that nobody will be having any fun. The graph below is exactly the same situation as the previous graph except this graph is plotted to 99% utilization. At 85% utilization the response time is about 7x and it just gets worse from there.



Insight #3:

The closer you are to the edge, the higher the price for being wrong. Imagine your plan called for a peak of 90% CPU utilization on the peak hour of your peak day but the users didn't read the plan. They worked the machine 10% harder than anticipated and drove the single CPU to 99% utilization. Your average response time for that service center was planned to be 10x, instead it is 100x. Ouch! This is a key reason that you want to build a safety cushion into any capacity plan.

Insight #4:

Response time increases are limited by the number that can wait. Mathematically, the queuing theory calculations predict that at 100% utilization you will see close to an infinite response time. That is clearly ridiculous in the real world as there are not an infinite number of users to send in work.

The max response time for any service center is limited by the total number of possible incoming requests. If, at worst case, there can only be 20 requests in need of service, then the maximum possible response time is 20x the service time. If you are the only process using a service center, no matter how much work you send it, there will be no increase in response time because you never have to wait for anyone ahead of you in line.

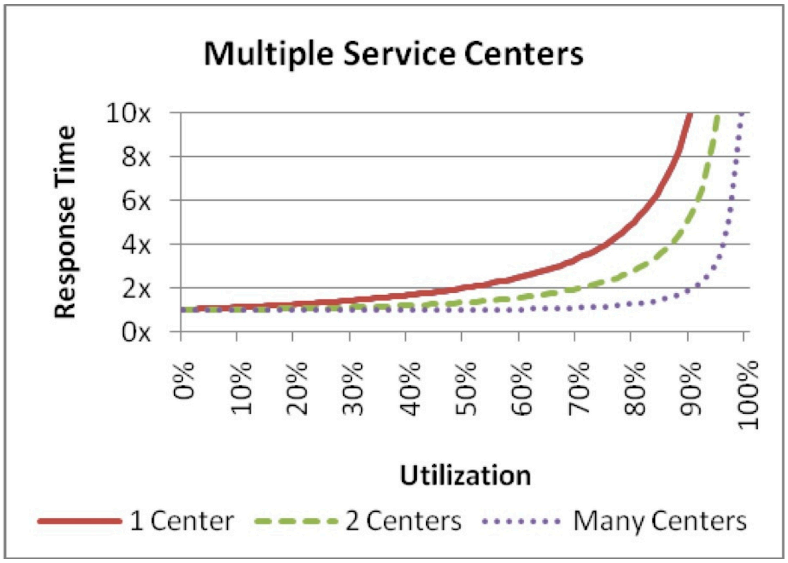
Insight #5:

Remember this is an average, not a maximum. If a single service center is at 75% utilization, then the average response time will be 4x the service time. Now a specific job might arrive when the service center is idle (no wait time) or it might arrive when there are dozens of jobs ahead of it to be processed (huge wait time).

The higher the utilization of the service center the more likely you are to see really ugly wait times and have trouble meeting your service level agreements. This is especially true if your service level agreements are written to specify that no transaction will take longer than X seconds.

Insight #6:

There is a human denial effect in multiple service centers. If there are multiple service centers that can handle the incoming work, then, as you push the utilization higher, the response time stays lower longer. Eventually the curve has to turn and when it does so the turn is sudden and sharp!



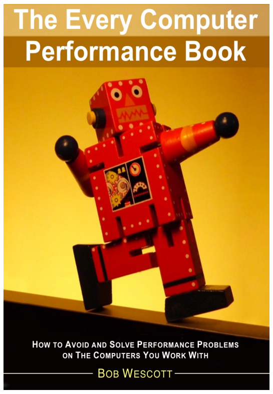
This effect makes sense if you think about buying groceries in a MegaMart. You are ready to check out and you look down the line of 10 cashiers and notice that seven of them are busy, three are free, so you go to the idle cashier. Even though the checkout service center is 70% busy overall, your wait time is zero, and therefore your response time is equal to the service time. Life is good.

On the other hand, if you have a system with eight available CPUs the response time will stay close to the service time as the CPU busy climbs to around 90%. At this point the response time curve turns upward violently. Add enough work to get the CPU to 95% busy, and the system becomes a world of response time pain. So, for resources with multiple service centers, you can run them hotter than single service center resources, but you have to be prepared to add capacity quickly or suffer horrendous jumps in response time. Most companies are much better at understanding real pain they are experiencing now, as opposed to future pain they may experience if they don’t spend lots of money now. ***Most corporations only learn through pain.***

Insight #7:

Show small improvements in their best light. If you see some change that will make the system 10% more efficient, when is the best time to tell your boss about it? If you reveal your idea during the slow season when the system is 20% busy, that small efficiency improvement will hardly be noticeable. If you wait until a busy time of the year when your system is on the ugly part of the response time curve you will be a hero and be a shoo-in for a promotion.

Of course, I'm kidding here. When you do find a silly waste of resources that can be easily fixed, take the time in your presentation to show the effect this small fix will have at the next seasonal peak.


The Every Computer
Performance Book
How to Avoid and Solve Performance Problems on The Computers You Work With
Bob Wescott