Jon's Performance Musings: Roadmap To Availability

Jon E. Schmidt
Transaction Design, Inc.
San Rafael, CA, USA

Jon is the founder of Transaction Design, Inc. (TDI), a consulting firm located in the San Francisco Area which specializes in capacity/ performance studies with clients worldwide. He is the creator of the Ban Bottlenecks® service and has an extensive background in the implementation, testing, and tuning of high-availability systems.

Not A Jolly Holiday

I don't know whether you saw the news, but Internet Retailer magazine listed a surprising number of sites that suffered outages around the Thanksgiving weekend, Black Friday, and Cyber Monday. (http://www.internetretailer.com/2011/11/28/thanksgiving-weekend-brings-multiple-site-headaches)

Their article details the downtime incurred by such sites as PC Mall, Brookstone, Target, and Victoria's Secret. American Eagle Outfitters was down for about 8 hours (!) on Wednesday. Victoria's Secret was down for about 80 minutes on Black Friday. Target has had a series of outages since they brought their site in-house, and had "problems" for more than two hours on Friday. Consequences? Certainly each outage meant lost business. But it also got personal: Target, for example, has a new "president, multichannel" as of November 22.

I suspect that these outages were all capacity-related. If so, they were all avoidable, if one were to follow the proper roadmap.

Getting There From Here: Where's Here?

It is possible to anticipate workload and requirements, and thereby estimate what resources are needed. But surprisingly, a lot of shops don't even know where they are now. What is the baseline? What's the current demand on the system? How much business is currently being processed? Is it growing, staying static, or is it slowing down? How do the messages going through the system relate to the business?

Solidly-managed shops have these numbers. It's not hard for an application to report its workload: Each application can create one data row per day: Records processed for batch applications; Transactions processed for online applications. If it's done right, it shouldn't be hard to collect these rows each day into an Excel sheet or a database, and do trend and peak analysis. Even if the application doesn't report the business numbers, they may be available by implication. Count messages submitted to queues. Count IP packets. Count I/O activity to a disk or to a file. Count the rows in a table, or the size of a file. As long as the count is proportional to the business workload, you have something to work with.

You need to take it one more level, though: Have the application report its peak each day. We recommend the peak half-hour, but you can pick what makes sense. Why do you want to know the peak? Because that's your design point. If you design your systems to handle the upcoming peak, the rest of the time will be covered.

Where's There?

If you know where you are, then you next have to figure out where you are going. Let's say you have all the traffic data. You probably have computed several growth factors: Year-to-date growth for transactions over last year; Month-to-month growth; Month-to-last year's month; Average day each month to last year; Peak day each month to last year; etc. If you're really good you also have similar numbers for the peak half-hour each month.

So, what number do you use to project what's coming? You have choices, so pick what your informed guess says is the most reasonable. Maybe you want to pick the worst case? Maybe you want to pick the average? In any case, there's one immutable rule:

Talk To The Business Partner

Always, always, always talk to the business partner about projections! No matter how good a job you do about collecting statistics and massaging them, unless you understand what is going to happen to the business those statistics are useless.

Hopefully your business partner will help you refine your estimate. S/he will have more information about such things as mergers, acquisitions, and marketing initiatives. Business changes such as these can have a huge impact on the growth of demand and the resultant utilization of the system.

Alternatively, not understanding the business plan can be very embarrassing. We've seen systems that have been hit with outages because they didn't plan on the increased traffic brought on by a successful advertising campaign, for example.

Map The Route

When you have numbers you believe, and when you've added sufficient "fudge factor" to the numbers, then you have to figure out what the impact of the increased workload will be. To all components of the transaction path. *All.*

As I've mentioned in previous articles, the throughput and response of an application is based on the sum of the steps that a transaction takes. If any of those steps is not capable of handling the workload, then you've failed. So, since you know the historical peaks and the historical utilization of the system components during those peaks, you can apply your growth factor to each and every utilization number: CPU; Disks or SAN; IP traffic; server processes; etc.

But the previous list is not the entire story. For most applications there are external calls. Maybe to a crypto box. Probably to a corporate database or data store. Maybe to a back-end bank or switch for authorizing transactions. Probably to one or more in-house SOA services.

So not only do you have to apply the growth factor to

the components you control, you have to apply them to the components you don't control. Any slowdown, anywhere along the path, can cause the system to fail.

How Many Lanes?

Slowdowns don't happen only because you've run out of CPU, disk, memory, or bandwidth. You can have tons of all of those left, and still provide terrible response or failed transactions.

Performance is like designing a road so that cars can drive it at 60 MPH. An engineer needs to think about degree of curvature, superelevation, sign size, signal timing, etc. Capacity is like expanding that road so that cars can drive it at 60 MPH during rush hour. So too with services. Here's the law:

> *Little's law — In queuing theory, "The average number of customers in a stable system (over some time interval) is equal to their average arrival rate, multiplied by their average time in the system."*

So if transactions are arriving, for example, at 5 per second, and are taking 2 seconds to complete, there will be, on average, 10 transactions "in-flight" in the system. Conversely, if a component of your service has 5 processing threads, and that component has a 1 second response, then the maximum throughput of that component is 5 TPS.

However, if the response time of that component slows down to 1.5 seconds, for example, as a result of disks becoming too busy or SAN congestion, the maximum throughput of that component is just 3.3 TPS. An arrival rate of 5 TPS with a bottlenecking service of 3.3 TPS is going to result in queues building up, and, unless something backs off, transactions are going to fail.

If you know that the Holiday peak is going to 10 per second, you need to create enough processing threads for each component to handle the throughput and the in-flight count, plus some.

Team It

One cannot assume that all of the components of a transaction's path are going to provide uniform, predictable response and throughput. As we have seen, any single service used when processing a transaction has the ability to become a bottleneck. Any such slowdown is going to cause an increase in the number of in-flight (simultaneous) transactions. One must assume that during peaks slowdowns will occur.

So, take your projections for throughput and in-flight requirements and publish them to everyone involved. Get them to be just as professional as you are in preparing for the peak.

Extreme retail: 5x growth over the Holidays.