What I Learned This Month: Lessons from Running out of Capacity

Scott Chapman
American Electric Power

Failures are always more instructive than successes, and I recently had some significant learning opportunities! We, like many shops, have a long history of running our mainframe systems very "hot": fully utilized for significant periods of time.

The mainframe's capability to classify different workloads by importance and manage the resources to protect the most important workloads is key to being able to run the systems fully utilized. When the system is busy, less important workloads such as development and test work run degraded, while important work, such as production online users, is protected such that it continues to meet its performance goals. This leads to some difficulty in discussing the amount of available capacity because it's different for every workload.

The hope and expectation is that the capacity situation will be monitored and additional capacity added before the situation turns critical for important workloads. However, sometimes things don't go according to plan. Recently we flat ran out of capacity. The most important production workload was being significantly delayed for CPU and we missed an online performance SLA metric for the first time in a very long time. And of course, if the most important workload is suffering significantly, the less important workloads are all but stopped. By any definition, we were out of capacity.

The high level measurements made it fairly clear why we were out of capacity. Our largest production CICS application went from somewhere around 200 transactions per second (tps) to around 300 tps—that sort of increase was not something I was expecting. Part of this was due to normal business cycle peaks: Mondays are busier than Fridays for this application, and summer is its normal peak time. Some was due to unexpected work: the business unit was running automated 3270 macros that were responsible for about 20% of the transactions. They've used automated macros in the past, but not at that level of intensity. Finally, some of my surprise at the peak came from a false sense of security from last year's summer peak, which was not as painful as it should have been because the active user count was reduced due to business changes. I didn't fully appreciate the significance of that fact last year.

Here are a few lessons I took from the experience.

The first, and somewhat obvious one, is to avoid adding workloads to your peak times if you can avoid it. Apparently the business unit and application team didn't have a good understanding of how much work their macros really were doing. They have promised to be more careful about scheduling such work in the future. We've worked together to plan and schedule such work in the past. It appears that this time their assumption was that the new workload wasn't that significant. So part of this lesson is not to assume that your new workload will be insignificant; measure it to make sure!

Fortunately, I have a fairly good database of business metrics[1] so I was able to tell pretty quickly that the increased utilization was related to an increase in business work, not a sudden technical problem. Included in my business metrics was the clue to why last summer was better than I should have expected: the logged-in user count this summer was higher than last, but similar to what it was last spring. The likely reason for this is the company-wide staff reduction that took place last June. While the goal was to reduce staff, some customer-facing positions were expected to be re-filled, so my theory is that staffing levels for the call centers last summer were artificially low. Certainly the logged on user count and known staff reduction actions support that theory.

Another interesting lesson is that while my BMDB tracks the daily peak and average values for a number of metrics, those two measures are sometimes inadequate for fully capturing the application's utilization profile. For example, I found certain values that had peaks that were even higher than what we were experiencing now, but the difference was that those peaks were more transient—we were now sustaining higher peak levels. I plan to add new measures to my database to better capture this issue, but I haven't fully decided how to do that: whether to average *n* peak measurements for the day or to record the 80th or 90th percentile or to do something else.

On/Off Capacity on Demand (OOCoD) makes it quick and easy to add hardware capacity, but the biggest issue is always the software license costs. The software costs tend to be the limiting factor, at least in our shop. By way of example, one day, over the course of a few hours, I tested a workload running on our smallest machine at about a dozen different capacities by varying the processor speed and number of defined processors. It's truly quick and easy to add/remove capacity or alter the engine speed as long as you plan for that when you buy the machine. But if we decide to do a capacity upgrade due to this latest incident, it probably won't happen for several months, in large part because that's how long it will take us to work through the budgetary work for the potential software cost impact.

So I guess the overall lesson this month is that the unexpected will happen. Being prepared for it is key: collecting historical data and having it at the ready is a very important part of being prepared. That historical data should contain average and peak values and ideally some sort of percentile measures. And the truly prepared would probably have software contracts in place that would let them add capacity via OOCoD at the drop of a hat.

As always, if you think I've got it all wrong, please email me at sachapman@aep.com and let me know!

---

[1] Back in 2006, I wrote a couple of CMG papers about building my business metrics database (BMDB). See http://www.cmg.org/proceedings/2006/6052.pdf and http://www.cmg.org/proceedings/2006/6122.pdf if you're interested. Or a search for "business metrics" on the CMG web site will also turn up some interesting papers.