Jon's Performance Musings: Catches

If you're the kind of system manager who enjoys the adrenaline rush when you pager goes off, then stop reading. On the other hand, if you're the kind of system manager who would prefer a stress-free existence, quietly anticipating and heading-off problems, then read on.

We review systems monthly. In that review we see not only what happened in the last month, but also how it compares to the last 24 months. This past period we had a number of nice "catches" that helped our clients anticipate and prevent issues. Here's a sampling.

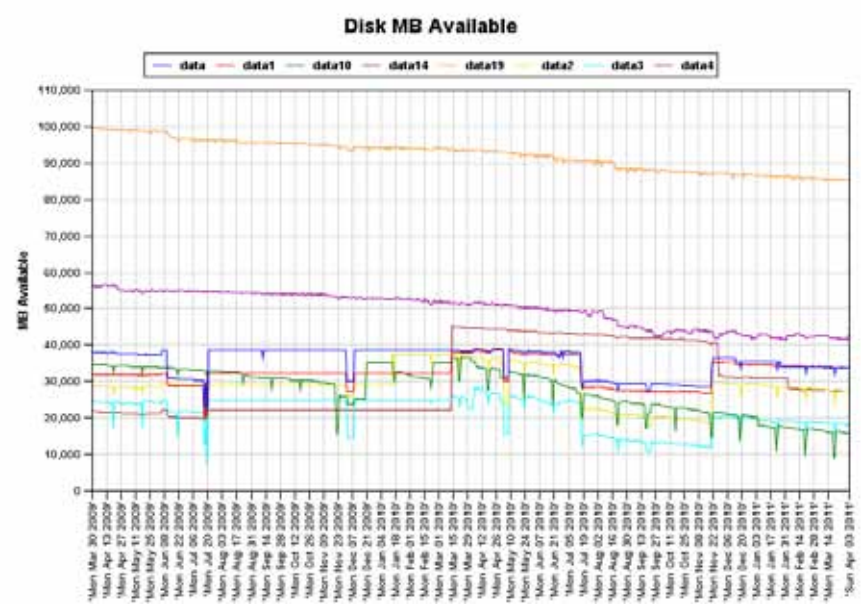
Janitor?

We recommend that our clients run a daily disk cleanup "janitor." These are not difficult to build, especially when the application logs are date-time stamped. If the janitor is run daily via a batch facility then it becomes an automatic way of maintaining clean disks.

Sometimes a janitor fails. The result is an ongoing depletion of disk space. Eventually this will cause an alert, hopefully, and someone will go in and cleanup up manually and restart the janitor. Otherwise the results could be very embarrassing for a high-availability system.

Janitors don't always cleanup everything. Perhaps a new application folder or subvol didn't get added to the janitor list. Perhaps an application has a single log which is not rolled over daily. Or perhaps the business traffic is simply growing and the logs and files are too.

Our catch this month was a disk volume which has been steadily depleting since March, 2010.



The interesting point is that there is a batch job which runs apparently once a month, and when it runs it uses about 8 GB of additional space on the volume. We are always looking for peaks, lows, and problems, and the graph of space available (left) shows the impact of that job. While there is usually lots of space on the volume, it has been dropping, and at the present rate that batch job will fill the volume in a month or two.

Recommendation? 1) Clean the volume, and/or 2) Move the monthly job's files to a volume with more space.

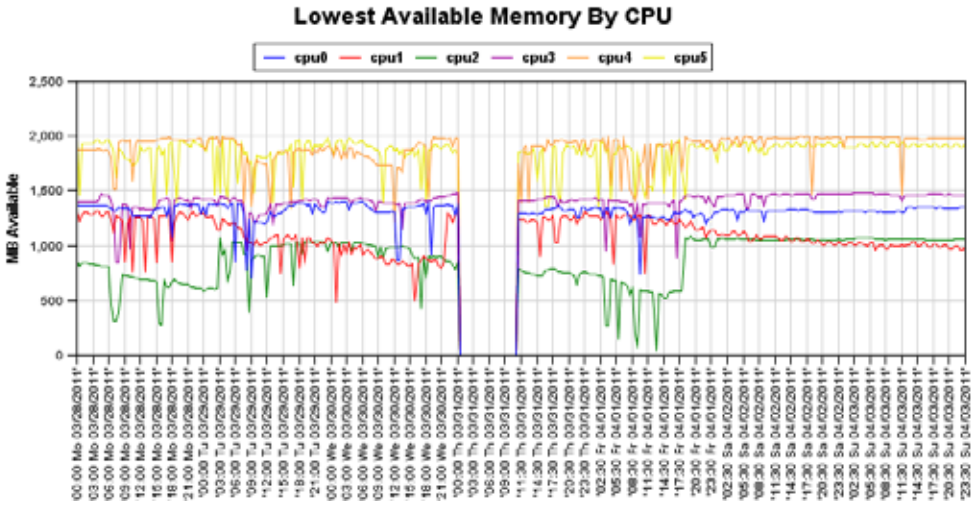
Memory

One of the biggest hits to online performance is caused when a box is out of memory and thrashing. If a process or job requests and tries to use a lot of memory that isn't physically available, every process on the box suffers. The operating system tries to respond to the request for more memory by throwing other process memory out to the swap file.

Interestingly, we see boxes all the time which have memory over-allocated. Most times it's not a problem. So a counter showing zero free memory is not sufficient to prove there is a problem. On the other hand, if you see zero free memory and a high swap (page fault) rate, then you have a problem. I've said before that there is no such thing as a good page fault. Only occasionally necessary ones.

So too, page faults are not necessarily an indication of low memory. Page faults are caused when a process is starting, loading code and allocating memory. A stable online system should have zero page faults. High page faults are an indication of transient processes, or low memory.

Our catch was for a major batch client who started to run a new, big job. This job, when it runs, will use almost a gigabyte of main memory. It runs several times a day. We noted two things: 1) The CPU in which it has been running seems to have a memory leak: Available memory drops consistently over the course of several days. 2) This particular CPU has the least memory available of the 6 CPUs in the system. Recommendation: Move the job to a CPU with the highest available (and stable) memory.

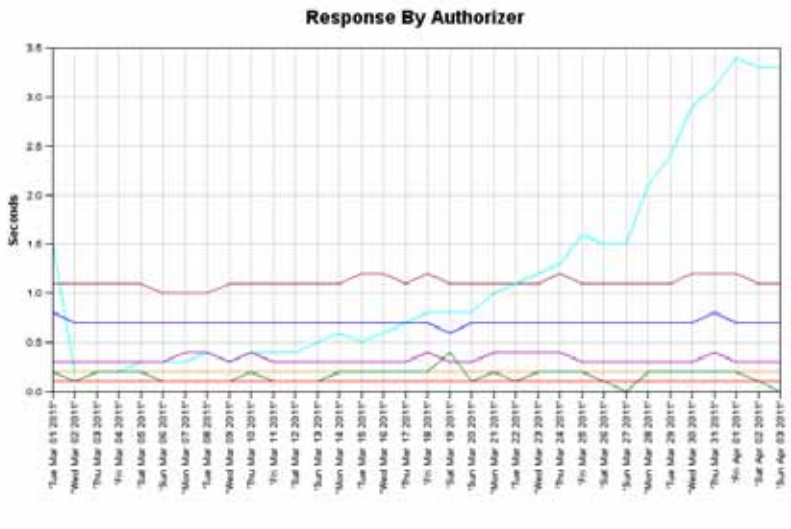


Flaky Partner

A previous article (CYA, Cover Your Aggregation) talked about the response time provided by your application to your clients being dependent on the response times you receive from external services. It's rare that an online application is completely contained in one box or system. Usually there are several external services such as database, SOA services, and external partners for authorization.

We recommend that you application design include measurements of the response of any off-box service. If you do it properly, you will be able to see slowdowns before they negatively affect

the service you are providing. At the very least you will able to assign the blame properly and allocate resource to the target that needs fixing.



Sometimes you don't have control over the external partner. That is the case for this catch. The client is switching out transactions to partners for authorization, and this month we saw that the response time for one particular partner went from less than 0.5 seconds, in a steady climb, to over 3 seconds (left). We brought this to the client's attention, and they are talking to this partner. Happily, transactions are not yet being hurt or reversed.

Home Run

Do it right, and (to mix a metaphor) good catches will result in a home run for your team.

Jon E. Schmidt
Transaction Design, Inc.
San Rafael, CA, 94901, USA
1.415.256.8369
inform@banbottlenecks.com

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.