Jon's Performance Musings: Inside Out

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.

Backwards Performance Management?

Do you ever have the suspicion that the state of the practice of performance management is backwards, inside-out, or upside-down? I do. As an industry we seem to have a fixation on a couple of basic system utilization numbers, which, to me, are totally inadequate to tell us what's happening with our systems. Performance management should be about how well our systems are handling their workload, not how hard they are working. It's as though we're driving our car by looking at the tachometer (RPMs), not at the road or the speedometer. A couple of recent conversations come to mind.

Green, But Empty

A client and I were chatting about his Wintel systems. We consult on his NonStop, but not the Windows systems. It seems that he was looking around for the cause of a slowdown, apparently not NonStop related, and was reviewing these other servers.

His management consoles showed nothing wrong, but clearly something was wrong. So he started to dig deeper. Then he found it. There were two servers in the path intended to load-share. However, as he dug, it became evident that only one of the servers was handling work. The other server hadn't done any work for months. Yet, that server had been "green" for all that time. The application was up, the server responded to monitoring requests, but did nothing productive.

Numbers For The Sake Of Numbers

I was having a conversation with a young colleague in Asia recently, and I was struck by his fascination with the measurements and data that are available on computer systems. Actually, I am also fascinated by that data, otherwise I wouldn't be in this business. But numbers are not the end game. Results are. The business is.

We had a long conversation about why a client should be interested in those numbers. It took me a while, but I eventually got him to understand that it is really the business that counts, not the usage numbers.

0% CPU Used Is Bad, 100% Is Good

Sometimes. Let's think about this. In an earlier discussion, a server wasn't being used. Its CPU utilization was very low, and that resulted in a "green" for that management console. But that was wrong. The server was effectively down because it wasn't handling any workload.

How can 100% CPU utilization be good? Remember that there are two different types of workload: Interactive and Batch.

Interactive workload uses the resources to provide excellent response. It requires that those resources be available, with short wait times to acquire the resource. Usually this requires lower rates of utilization than 100% for each resource in the transaction path.

Contrast that to batch which uses the resources to their fullest in order to process the workload in the shortest time possible. A well-tuned batch system will drive everything to 100% busy, or at least drive the resources until a bottleneck is reached on the first limiting resource.

The challenge comes from a mixed environment, where batch is competing with interactive. Here, tuning process priorities (CPU) and disk and communications balancing are critical. Even while batch is eating all available resource, the interactive application must still have quick access to the resources it needs.

The System Is Busy: Why?

Utilization numbers occasionally peak. Is this a problem? It depends. I simply can't understand sites and toolsets that don't capture process and workload data. If a CPU or other resource is too busy, the first question should be "What process(es) are using CPU?" And the next question should be "Why are they working harder than usual? Has the workload changed?" Increased workload is usually a good thing. It should mean increased business, with the resultant increase in revenue.

Meaningful Status

Green on a performance management screen needs to be more meaningful. It is useless if it simply indicates that CPU cycles are available. The definition must take into account whether the system is doing meaningful work, and whether that work is being handled properly. Let me suggest the following criteria for a "green" status:

a) The system is processing workload.
b) The interactive workload is giving/getting acceptable response.
c) Queuing for resources for the interactive processes are within acceptable limits.
d) The batch workload will finish within its window.

Application Intelligence

The criteria above will require some intelligence built into either the applications or into the monitoring software.

It shouldn't be too hard to design monitoring software to have a "learn phase" where it looks at a system and identifies the processes that are handling the interactive traffic and the pattern of that traffic. It can create a per-transaction or per-message profile of each process or process class, and then use that profile for actively monitoring the system. If that profile gets out of whack, then the monitoring software can change the green status to raise an alarm. Some of the stats it can review are CPU utilization and disk I/O per transaction; queue depth or message age, or a "wait for resource" metric (if available from the OS).

The application itself can report to the monitoring software if it thinks things are slowing down, or if it isn't getting workload when it should. For that matter, the application or networking stack should report if there isn't any traffic coming in from the network when it should.

Intelligent Monitoring

The tools used for monitoring today's application tiers can and should be more intelligent and more business oriented. A simple-minded reliance on the "usual suspects" of utilization metrics such as CPU, disk I/O, etc. isn't sufficient to provide managers insight into what's actually happening in those tiers. Applications need intelligent reporting which can be fed into the status stream for the management consoles. Only this way will we minimize the false greens and meaningless reds.