Jon's Performance Musings: Non-Balanced NonStop

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

Unbalanced!

No, I'm not talking about the author of this column. (Although the term may apply.) The topic refers to the need for a NonStop to be as balanced as possible in order to optimize throughput. This is an "opportunity" for those of us who are managers of NonStop systems to show our expertise and earn our exorbitantly high salaries. (Yeah, right!)

NonStop, as an asymmetric multiprocessor (multiple CPUs or blades each with memory and an OS image), with symmetric multiprocessor characteristics (multiple cores sharing the memory and OS image), poses a non-trivial balancing challenge. Kudos to the HP engineers who successfully migrated a single-thread kernel and disk process design into an extremely effective multi/multi architecture!

Intelligent Use

As application managers, however, we need to be aware of how our software processes are using the full throughput available to us. In the beginning of time we discovered that a single monolithic program won't work well on the NonStop. That program had to be segmented or replicated so that there were copies running on each processor on a node. Only in that way would the full throughput available on the NonStop be utilized.

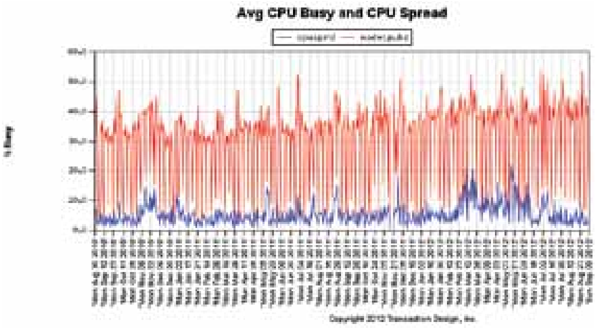
But segmenting the application isn't enough. The workload also needs to be balanced and distributed. Transactions need to be spread equally across processes/processors. With multi-core blades the workload needs to be allocated across the cores as well, although this is handled by the NonStop Kernel.

Saturation

Even on well-managed systems the balancing can drift. When it does, you want to be able to detect the imbalance before it results in saturation of a resource. Saturation means that the work being processed is not using all the resources available to it, and is bottlenecking on something like a CPU, core, or disk. The result is slower than necessary response or throughput.

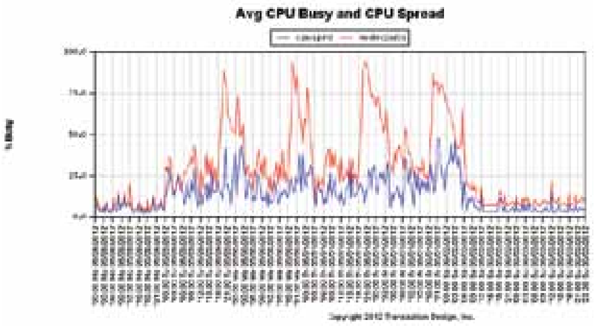
I thought you may be interested in this example we ran across recently. The client is a service bureau with over 1,000 users around the world. Their clients are able to run ad-hoc queries and batch reports basically anytime. After a cold boot we started to see their CPU usage patterns change.

One of the many metrics we track over time, in addition to average CPU usage, is the average CPU spread. In the chart below we saw that beginning in February 2012 the spread (blue line) widened significantly, even though the CPU usage didn't change that much. This chart also shows that the problem was fixed in July 2012. (Scale: daily average for 24 months.)

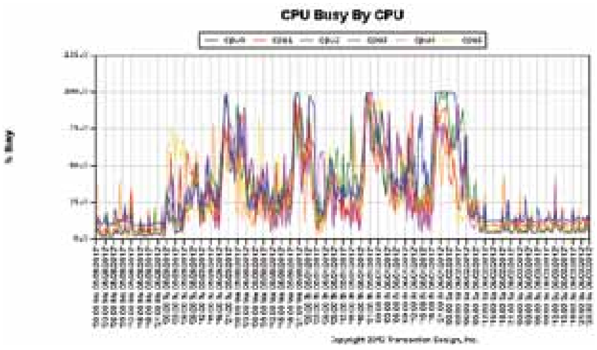


When we saw the spread increasing we raised the question: What changed, and how is it impacting the system? We knew that they had rebooted the node in February, but "That shouldn't have changed anything."

In our next report cycle the imbalance was still there, and we dug a little deeper. We could see that the CPUs weren't as balanced as they had been in previous months. Occasionally the spread among the CPUs approached 50%. Of course, if the CPUs are generally not busy, this isn't a problem. (Scale: 1 week by half-hour.)

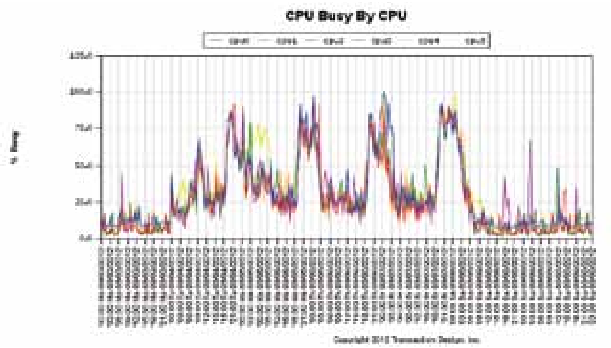


Unfortunately, the chart below shows that the CPUs were occasionally "pegging" at 100% busy. The chart shows that a CPU could be pegged for hours at a time. This is not good. It means that the processes on the pegged CPU were waiting for cycles, and as a result were running more slowly than necessary, especially since other CPUs had cycles available. (Scale: 1 week by half-hour.)

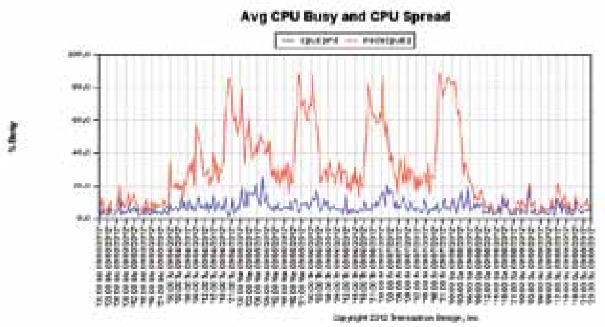


In a situation like this where a CPU is pegged it could be the result of a single process looping, or a very heavy batch processing cycle. We went looking for that and didn't find anything. In fact, this system is known for immense numbers of very short jobs, sometimes over 10,000 in a half-hour. So if things were balanced well there is no reason a CPU should be pegged.

After our continuing to raise the question with the client and showing them the implications of the imbalance, the client's team discovered that a load-balancing process that was supposed to be working wasn't. As noted above, they fixed the problem in mid-July and the charts below show the results.



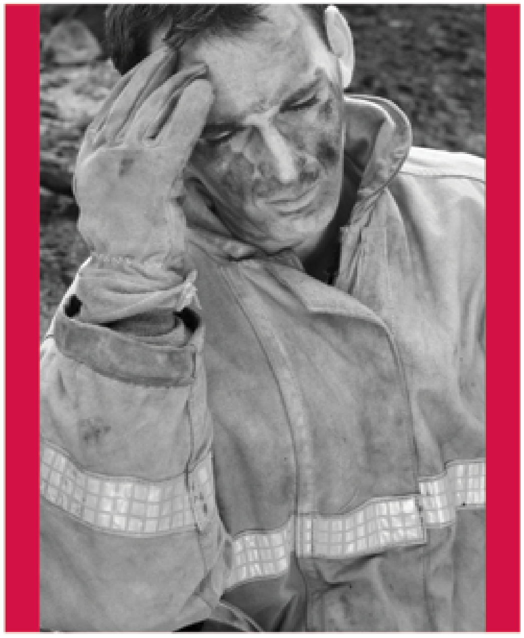
The chart above shows that the CPUs are much more in balance, with very little time sitting at or near 100% busy. The chart below shows that the CPU spread number rarely breaks 20%. Certainly much better than before. We also have evidence that their jobs were completing more quickly after the fix.



Balance In All Things

The above example shows CPU activity, but the same principle applies to any component on the system or in the network. Disks, files, network interfaces, server processes, anything can become out of balance or saturated and can slow down processing. The challenge is to always be vigilant, and to re-balance before things start to slow down.

Tired of putting out fires?



Ban Bottlenecks®
augments your procedures
and empowers your staff:

- Free Toolkit for connectionless high-resolution data collection
- Business transaction traffic analysis with service level reporting by transaction originators and authorizers
- Automatic daily reports covering the business, CPUs, memory, paging, disks, processes, and communications, suitable for feeding SAS
- Automatic weekly peak and problem analysis
- Analyst-scored, superbly-detailed, custom reports with 24 month historical and forward-looking perspectives
- Web conferences to review the report, discuss our findings.
- "Needle in the haystack" multi-system, multi-tier, and/or multi-architecture problem determination

BAN BOTTLENECKS®
Transaction Design, Inc.
542 San Pedro Cove, San Rafael, CA 94901
inform@banbottlenecks.com
www.banbottlenecks.com

Call us for a free system analysis 1.415.256.8369