IBM Systems MAGAZINE

Features



# Herd Your Rogue Servers

**Tips for sizing and server consolidation**

February | March 2008 | by Jaqui Lynch


Table 1

Now that we're in 2008, we can look around at the IBM* server landscape and see that server consolidation offers incredible savings in hardware, software licenses, environmentals (power, cooling, floor space) and people time to manage the systems (as fewer systems are easier to manage). The bigger issue arising is how one gets from here to there: How do you figure out what can be consolidated onto one server? How do you know what will play well together? Once you consolidate onto fewer servers, how do you deal with planned downtime for maintenance? We saw some hints in the POWER5* line of servers and the POWER6* line seems to provide many of the answers we've been seeking.

## Sizing

First, let's look at sizing. One of the most poorly understood areas in distributed systems is that of proper capacity planning, which incorporates both workload characterization and workload granularity. Workload granularity refers to how small the workload can be. As an example, if the workload needs 7 rPerf to run well, then you need to ensure that it still gets 7 rPerf after you consolidate the servers. You also need to understand the characteristics of the server—if the rPerf needed equates to 0.05 of a core you'll still need to give that workload 0.10 of a core (twice what it needs) as that's the minimum you can go down to.

This points to another issue in terms of consolidations—namely sizing by total instead of component. By this I mean you should take each workload being consolidated, translate it into rPerf, figure out how many cores that equates to and then sum the cores (not the rPerf). Misunderstandings around workload granularity have led to many undersized servers and performance problems.

As can be seen from Table 1, the number of cores needed varies widely depending on the type of server and whether shared processors or dedicated processors are being used for the LPARs. If the server for this consolidation effort had been sized using total rPerf instead of cores, then it would have been undersized significantly. These are all components to be considered in any sizing study.

## To Share or Not to Share

The next issue that arises is whether all workloads should go into the shared-processor pool or whether they need dedicated processor cores. This is an important question and requires that the planner understand the workload characteristics. As an example, is this a heavily multithreaded workload with many small random transactions or is it a workload that involves long, complex processor-intense transactions?

Why does this matter? To answer this question, let's use two examples—one where each new transaction needs one-tenth of the dispatch cycle and the second where each new transaction needs eight-tenths of the dispatch cycle to fully execute. (For the sake of this point we'll assume no I/O.) If the LPAR is using a dedicated core, then once the thread is dispatched it can use a full dispatch cycle before being involuntarily context switched. If it completes sooner, then it voluntarily cedes unused resources.

Now compare that to the situation where the LPAR is in the shared pool and is allocated 1 core but across 10 virtual processors (VPs), allowing each thread one-tenth of a cycle. Since our first thread only wants one-tenth we're fine, but the second one needs eight-tenths of a cycle and runs the risk of being involuntarily context switched seven times before completion. Dedicated cores have a 1-to-1 mapping with procs in the LPAR and the dispatch unit is the real processor. In the shared pool the dispatch unit is the VP, which gets assigned to its core at the time of dispatch and may not necessarily be assigned to the same core the next time it's dispatched. This can lead to a small dispatch latency, which is fine with most workloads but badly affects a few others. One example used for comparison was a specific SAS workload being benchmarked—there was a significant difference when we went back to dedicated cores. This is why it's important to understand the workload characteristics. Most workloads perform well in a well-planned shared-processor pool environment, but some perform much better with dedicated cores.

So how do you figure this out? The best way, if you can, is to do a small benchmark and try it both ways. If that's not a possibility, try looking at statistics on the various threads.

## What to Virtualize

The other question everyone is asking is what should and shouldn't be virtualized? This yields another "it depends" answer. It depends on what you're trying to do, the workload and the end goal. For example, Live Partition Migration (LPM) requires that the Virtual I/O Server (VIOS) own all resources for the LPAR at the time of the migration. The key here is in those last six words—"at the time of the migration." This means that, if necessary, the workload could use real adapters to get to its data or network most of the time and that you migrate them to VIOS-owned resources when you need to take advantage of LPM. Or you may just run them as VIOS-owned all the time.

While the trend is to virtualize everything wherever possible to take best advantage of the physical resources this may not always be the preferred answer. The good news is that it's not an either/or situation. An LPAR could use shared Ethernet adapter (SEA) for some resources and a dedicated Ethernet for others, and the same applies to shared SCSI and dedicated fibre cards. For an extremely high I/O workload the preference may be for dedicated fibre cards to the database. The same applies to Ethernet cards. There's a section in the "Advanced POWER* Virtualization Introduction" Redbooks* publication (number SG24-7940 at wwww.redbooks.com) that you'll find very helpful when it comes to the sizing calculations. Again it's important to understand the workload in order to make these decisions as the sizing recommendations are based on throughput, blocksizes and other performance-related values.

## Other Sizing Notes

When sizing memory it's important to remember the hypervisor and page-table overhead. In particular, page-table overhead can be surprising as it's based on one-sixty-fourth of whatever the maximum (not desired) memory setting is per LPAR. On a server with 128 GB of memory if an LPAR is setup with 4 GB desired and 64 GB maximum, the system will reserve 1 GB of memory on top of the 4 GB being used so that it can map page tables. It's important to set maximum memory values sensibly and ensure the memory overhead is calculated when sizing memory for the server. As an example, I normally start with the following steps to be safe:

- Hypervisor: 1 GB (it's less than this to start but I'm conservative)
- Page table: maximum memory of 64 and round up to nearest 256 MB
    - If 256 isn't your default logical memory blocksize, then round to that
    - Calculate this number for every LPAR and total
- I/O drawers: If I have these then I add 1 GB for every 4 GB. No number has been published for this so I use this to be safe.
- Integrated virtual Ethernet (IVE): Each active IVE adapter port takes 102 MB

You may need to add additional overhead if you make extensive use of virtual Ethernet between LPARs.

Another method is to take 8 percent of real memory and assume it's taken for overhead. A third method is to use the free System Planning Tool from IBM (www.ibm.com/servers/eserver/support/tools/systemplanningtool) as it provides an estimate of the memory overhead. It doesn't seem to include anything for the I/O drawers so I still add my 1 GB for every 4 GB. Either way, the key point is that you need to reserve some of the real memory for system overhead for the hypervisor and that needs to be included in the calculation for total memory.

## VIO Servers

When sizing the server it's important to include enough processor and memory for the VIO servers. Some customers run two (for redundancy) and some run four (two for Ethernet and two for shared SCSI). Either way, each VIO server must be allocated sufficient memory and processor to run the workload. Sizing recommendations can be found in the Redbooks publication. I normally start with at least .2 of a core (uncapped) and 2 or 3 GB of memory. But you may need to start with more depending on its use.

## Critical Work

The purpose of this article was to highlight the need for not just planning when consolidating workloads, but also for understanding both workload granularity and workload characterization. These two areas are critical to correctly sizing the server and any associated cards. Workload granularity affects how small the work-load can be and its required minimum resources. This then needs to be translated into actual physical resources, which must also adhere to the minimums required by the server itself. Workload characterization refers to the kind of workload and how it uses the resources that it's supplied. Understanding this can help you correctly choose the number of VPs. Without understanding these two areas it's very difficult to correctly size the system.