close window

IBM Systems MAGAZINE

Technical Corner

Print

Virtualization 101 for Power Systems

October | November 2009 | by Jaqui Lynch

Table 1

Resources

Since the introduction of POWER5* technology, virtualization has offered many great options. It's the foundation for server consolidation; it attempts to reduce planned downtime and is important in the green-computing arena. Not only are the servers faster and larger with more available memory, but they also include options such as the virtual I/O (VIO) server, virtual Ethernet, shared Ethernet adapter (SEA), virtual SCSI and Micro-Partitioning* capabilities. Simultaneous multi-threading (SMT) adds a performance boost to most workloads. However, the key to using these features is implementing a minimum of AIX* V5.3 (or Linux*) as this is the OS version that takes advantage of the new features. AIX V6 on POWER6* processor-based servers offer another step forward in the virtualization paradigm.

Another advantage of POWER5 and POWER6 servers is they provide the capability to concurrently run multiple disparate OSs on the same server, in different LPARs. A server can run AIX V5.3, AIX V6.1, SUSE Linux Enterprise Server and Red Hat Enterprise Linux, all in their own LPARs. In addition, all of these are supported using the VIO server facilities. Either a hardware management console (HMC) or integrated virtualization-manager (IVM) software manages the servers.

Options and Terminology

The terminology around virtualization is confusing. It breaks down into three areas—CPU, memory and I/O (network and disk). Unfortunately, some commands use slightly different acronyms to mean the same thing, which doesn't help. Virtualization on POWER6 processors is provided via a feature called PowerVM*. This was called Advanced Power* Virtualization on the POWER5* systems. The POWER6 line includes three server versions: Express, Standard and Enterprise editions. Express is only available on entry-level systems (Power 520 and Power 550) and has limited functionality, including only supporting IVM and three LPARs inclusive of the VIO server. The Standard Edition provides all of the features of virtualization with the exception of two functions that are reserved for Enterprise Edition. Enterprise Edition enhances the Standard Edition by adding the functionality for live partition mobility and Active Memory* Sharing.

For POWER5 and POWER6 servers the POWER Hypervisor* is always running and provides the functionality that separates the physical entities like CPU, memory and I/O devices from the actual software or LPARs. Running VIO servers to support client LPARs further enhances the virtualization capabilities. VIO servers are custom LPARs that provide the functionality for SEAs and virtual SCSI, as well as the advanced functionality of live partition mobility and Active Memory Sharing. Using VIO servers allows cards that are in PCI slots to be shared among LPARs rather than dedicating whole slots to LPARs, even when the LPAR is making minimal use of the card. This can lead to a significant reduction in I/O drawers, I/O cards and the associated costs.

Settings

It’s important to understand the three settings for memory and CPU resources. Minimum is the minimum needed before the LPAR will boot; desired is what the LPAR will normally boot with if it’s available; and maximum is what the running LPAR can be increased to using a dynamic LPAR operation.

Virtual Ethernet and SEA

Virtual Ethernet has been around since AIX V5.3 and doesn’t require PowerVM. It’s the capability of two LPARs to communicate via the Hypervisor over virtual Ethernet channels. Virtual Ethernet requires some CPU and memory to transfer network traffic. It also allows for virtual LANs and other security mechanisms.

SEA takes advantage of virtual Ethernet. Rather than dedicating an Ethernet adapter in every LPAR on the VIO server, SEA has the actual adapter (or a group of adapters aggregated together) assigned to the VIO server. The LPARs talk to the VIO server using virtual Ethernet, and any traffic that must go outside the box is sent out via the SEA in the VIO server. For redundancy, normally two VIO servers act as SEAs, and it’s common to set up one as primary with the other running as failover. SEAs reduce the number of Ethernet cards on a system and to provide for failover and redundancy.

Virtual SCSI

If you wanted to provide a boot disk to an LPAR previously, it was necessary to take up a slot for a card that connected the disk and that slot got dedicated to the LPAR. As servers get larger and faster, more LPARs are being consolidated onto them–which causes a significant increase in the number of disks and slots being consumed, just to boot the LPAR, never mind the actual data disks. Virtual SCSI provides the capability to have a VIO server own the adapters and the disks, and that VIO server can carve up disks and provide chunks of disks (or whole disks) to LPARs so the client LPAR thinks it has a full disk for booting. As an example, most LPARs rootvgs are between 30 and 45 GB, depending on how clean they keep their rootvg. The smallest disk now is around 146 GB. With a VIO server this 146 GB disk can easily be carved into three logical volumes, each of which could be a boot volume for a different LPAR. Even using two VIO servers for redundancy, this is still a significant reduction in disks, PCI cards and I/O drawers.

Memory

Until recently, memory was straightforward: It was dedicated to an LPAR. The key point to remember was the size of the Hypervisor page table entries—used to keep track of the real memory-to-virtual memory mappings for the LPAR—was calculated based on maximum, not desired, memory. Users applied common sense so the maximum memory for an LPAR or Hypervisor overhead wasn't much higher than necessary. As an example, if a server has 128 GB of memory and the LPAR has a desired of 4 GB then set the maximum to something like 8 GB, not 128 GB.

Hypervisor overhead for these page-table entries is normally calculated by dividing the maximum memory setting by 64 and rounding up to the nearest logical memory-block size. Using the example above, if we set maximum memory to 128 GB the Hypervisor would reserve at least 2 GB of memory for page table entries for that LPAR, even though 4 GB is desired. If the maximum is set to 8 GB then 128 MB (or 256 MB depending on the logical memory block) is reserved.

In April, IBM announced a feature called Active Memory Sharing. This tool provides pools of memory that can be shared by partitions, which allows memory to be over-committed in certain circumstances so it isn't necessary to buy as much memory. Active Memory Sharing requires AIX V6.1 (or IBM i or Linux), POWER6 hardware and some specific firmware versions. It also requires PowerVM Enterprise Edition, all resources in the LPAR must be virtualized and the LPAR must be in the shared processor pool (micro-partitioned).

Active Memory Sharing is designed for LPARs with variable memory requirements. The VIO server has a special set of Active Memory Sharing paging devices and it pages out memory from one LPAR when another LPAR needs it, depending on options chosen during setup. Active Memory Sharing isn't designed for workloads that have a high rate of sustained memory use or that mandate predictable, high-quality performance. It's ideal for most test and development environments. For more information on how Active Memory Sharing works, see "Upping the Ante" online (www.ibmsystemsmag.com/aix/junejuly09/coverstory/25427p1.aspx).

CPUs

A significant amount of confusion surrounds the terminology used in discussing CPUs. In an IBM Power Systems environment, CPU means a single processor, or a single core. This isn't always the case for other vendors, so the safest term to use is cores—especially when discussing licensing.

Cores are either dedicated to an LPAR or in the shared-processor pool. There's a version called dedicated donating, but its behavior is essentially similar to being in the shared-processor pool. In the case of a dedicated core, a physical core is assigned to the LPAR at boot time and the same core stays with the LPAR throughout. Since a clock cycle provides a 10 millisecond (ms) dispatch window, then the dispatch window for that core will be 10 ms. Any cores that aren't currently allocated to dedicated LPARs normally revert to being in the shared-processor pool.

Prior to POWER6 technology there was only one shared-processor pool. POWER6 servers provide the capability to have up to 64 pools with the default pool being pool 0. When LPARs are defined as using the shared-processor pool, then they can take advantage of Micro-partitioning capabilities. This means you can assign an LPAR as little as one tenth of a core and that LPAR can shrink and grow dynamically, provided it’s set up correctly.

The POWER5 processor-based Power Systems servers provided for SMT. In POWER5 and POWER6 processor-based servers, many of the registers got doubled, so it’s possible to dispatch two threads to the same core during the same dispatch window and have them run concurrently—taking advantage of the pipelining register duplication provides.

These three concepts can lead to much confusion as you try to understand capacity settings, virtual CPUs, processing units and logical CPUs. First, capacity is basically a measure of what you can use. Just as a pint bottle has a certain capacity, so does an LPAR when it’s assigned resources such as CPU and memory.

Dedicated CPUs

In the dedicated world, a server with one core will show one processor (probably proc0 from lsdev –Ccprocessor). If SMT is off, then lcpu in a vmstat will show there’s one logical CPU, and lparstat will show one online virtual CPU. This is confusing as neither concept exists in this case. In the vmstat, with SMT off, the lcpu=1 means one real core is assigned to the LPAR. When a thread is dispatched, it’s dispatched to proc0, which represents the real core.

If SMT is on and vmstat and lparstat run, you’ll see lcpu=2 in the vmstat and the lparstat will show one online virtual CPU. In the dedicated world there’s no such concept as a virtual CPU, but the command terminology doesn’t change. A virtual CPU in the lparstat is actually a real CPU mislabeled. However, by turning on SMT, you now have the concept of a logical CPU. Each logical CPU represents one of the threads that can be run on the same core at the same time. It’s now possible to use mpstat –s to see how the threads are being dispatched. In the vmstat, with SMT on, the lcpu=2 means you have one core. When a thread is dispatched, it’s dispatched to the logical CPU, which then maps proc0, which represents the real core.

In a dedicated world there are five key settings: minimum processor, desired processor, maximum processor, dedicated donating and an option that determines whether the cores revert to the pool if the LPAR isn’t running. Desired processor is what this LPAR will try to get at boot to run optimally. All of the processor settings for dedicated LPARs are set in whole core numbers.

Shared CPUs

This is where the concept of virtual processors or CPUs has meaning. Cores are placed into a processor pool and LPARs are then assigned to that pool with specific settings. In terms of core allocations, there are now six settings: minimum, desired and maximum processor units, and minimum, desired and maximum virtual processors. Since fractional cores can be assigned to an LPAR, the term processor units (PUs) is used, rather than cores.

As an example, we have an LPAR assigned with a minimum of 0.1 PUs, a desired of 0.8 PUs and a maximum of 6 PUs. It’s also assigned with a minimum of one virtual processor, desired of two virtual processors, a maximum of six virtual processors and it’s uncapped.

SMT is on. The LPAR booted with its desired settings of 0.8 PUs and two virtual processors.

A vmstat command will show an LCPU of four, lparstat will show two online virtual CPUs. Both vmstat and lparstat will show an entitlement of 0.8 and vmstat will also show a field called %entc and another called pc.

The desired setting for PUs (0.8 in this case) is also known as entitled capacity. In vmstat, % entc is the percentage of entitlement in use and pc is the number of processor units currently in use. When the LPAR boots and obtains its desired PUs, it's then guaranteed it can always get to its desired entitlement PUs whenever it needs them. If the LPAR isn't using those PUs, then the Hypervisor can assign them to another LPAR. Conversely, if the LPAR is uncapped it can grow beyond its entitlement and use processor resources that other LPARs aren't using.

Virtual processors serve several purposes in the shared world. First, virtual processors determine how many cores the LPAR thinks it has. If VP=2 then the LPAR will have a proc0 and a proc2 and will think it has two physical cores. With a desired of 0.8 this equates to two virtual processors, each with an initial dispatch window of 4 ms instead of the 10 ms a full core would have. When both virtual processors are dispatched, they could be dispatched to the same or different cores. The system tries to dispatch them to the same core they ran on previously, but there's no guarantee. Attention must be paid to virtual-processor settings, as they also tell the LPAR how big it can grow and some software vendors use them in licensing calculations.

In the aforementioned example, the LPAR is uncapped, six cores are in the shared-processor pool, entitlement is 0.8 PUs and virtual processors are set to two; SMT is on. LCPU in this case will show as four: one for each virtual processor and then double it to add SMT. Additionally, even though this LPAR is uncapped, it can't grow any bigger than two PUs as the virtual processors are acting as a cap. In order to use all six cores in the pool, virtual processors must be set to six. An uncapped LPAR can exceed its entitled capacity (desired PUs) up to the number of desired virtual processors or the size of the pool, whichever is smaller.

Some important things to note about Micro-partitioning technology:

- As with dedicated cores, the dispatch time for a full core is a 10 ms timeslice.
- A 0.3 entitlement means 30 percent of a core or a 3 ms timeslice.
- A 1.4 entitlement means the LPAR is entitled to 14 ms of processing time for each 10 ms timeslice (obviously across multiple cores). For two virtual processors, this equates to 7 ms on each of two cores.
- The more virtual processors that are allocated for a core, the smaller the dispatch window is.
- An LPAR may run on multiple cores depending on entitled capacity, virtual processors and interrupts.
- The number of desired virtual processors cannot exceed 10 times its entitlement or 64, whichever is smaller.

In an uncapped world, each virtual processor can grow in the capacity it's using, up to a full core (or 10 ms). So, although an LPAR may only have a 0.8 entitlement, if it has two virtual processors and is uncapped, it can grow to two full cores if the resources are available.

However, if virtual processors were set to one, then it couldn’t grow beyond one core. If an LPAR is set up as capped, then the rules change; a capped LPAR can’t exceed its entitlement. So that LPAR could never exceed 0.8 PUs if it’s capped (see Table 1).

Virtualization Benefits and Challenges

Clearly, some significant savings can be made by judiciously implementing virtualization. However, it’s important to understand the terminology in order to understand the concepts behind virtualization. In particular the concepts of entitlement, processor units, virtual processors, capacity and pools are critical.