close window

Print 🖶

Web Exclusive

VIOS 101: Memory

June 2015 | by Jaqui Lynch

In Part 1, we did an introduction to virtualization and then discussed CPU virtualization with PowerVM. In this article, we will delve into the options for memory in a virtualized environment.

Memory

Memory allocation is still fairly straightforward. It's basically dedicated to an LPAR. As with CPU, you provide the system with three values—*minimum* which is what is required to boot, desired which is what the system attempts to boot and run with normally and *maximum* which is what you can grow to with a DLPAR (dynamic LPAR) operation.

When setting memory values, it's important to remember that the size of the Hypervisor page table (HPT) entries that are used to keep track of the real memory to virtual memory mappings for the LPAR is calculated based on maximum, not desired, memory. This means that common sense needs to be applied to setting maximum memory for an LPAR or Hypervisor memory overhead will be 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 maximum to something like 8 GB, not 128 GB. Hypervisor overhead for these HPT entries is normally calculated by dividing the maximum memory setting by 64 and rounding up to the nearest LMB (logical memory block) size. For POWER7+ and POWER8 the divisor is now 128, not 64, but it can still make a significant difference.

Using the previous example on a POWER7, if we set maximum memory to 128 GB then the hypervisor would reserve at least 2 GB of memory for page table entries for that one LPAR, even though it's desired is only 4 GB. If maximum is set to 8 then 128 MB (or 256 MB depending on the LMB) is reserved. LMB is also referred to as MRS or memory region size. You can find the LMB size by going to the HMC and looking at the properties for the server—at the HMC, click on Systems management, Servers and then on the server itself. If you click on properties and then the memory tab, you will see your memory settings, which include two lines:

```
Memory region size:          0.25 GB (256 MB)
Active memory sharing support:  Yes
```

According to the memory region size or LMB my server is set to 256 MB. It's important to know this setting if you plan to use live partition mobility (LPM) as all servers in an LPM operation must have the same LMB size. To change this, you'll need to power off the server after making the change, so it's important to set it on all servers to a consistent value.

Four additional technologies affect memory—these are active memory sharing (AMS), active memory deduplication, active memory expansion (AME) and active memory mirroring. You can check on the HMC whether your server can support these technologies. Look at the server properties and click on capabilities. Check for each of the technologies you're interested in utilizing.

Active Memory Sharing

As you can see in the two lines we've already pulled from this server, it also supports AMS. AMS is a PowerVM feature that provides pools of memory that can be shared by partitions. This allows for memory to be overcommitted in certain circumstances so that it's not necessary to buy as much memory. Use of AMS requires a minimum of AIX v6.1 (or IBM i or Linux), POWER6 hardware and some specific firmware versions. It requires PowerVM Enterprise and that all resources in the LPAR be virtualized (you must be using a VIO server for network and disk I/O) and the LPAR must be in the shared processor pool (micro-partitioned).

AMS is designed for LPARs with variable memory requirements. The VIO server has a special set of AMS paging devices and it pages out memory from one LPAR when it's needed by another LPAR, depending on which options were chosen during setup. AMS is not designed for workloads that have high sustained memory use or that mandate predictable high-quality performance. It is ideal right now for most test and development environments.

Chapter 9 in the PowerVM Introduction Redbooks publication includes all the information on planning for AMS including prerequisites and potential configurations. If you're using AMS, you need to pay careful attention to the READMEs (you should anyway) whenever you upgrade your VIO servers. There are also some technologies that you may want to take advantage of that require AMS (like active memory deduplication) and others that won't work with it.

AMS basically turns your VIO server into a paging device for the LPARs so you'll need to ensure that your VIO server has sufficient high performance disk to accommodate those paging devices. Those paging devices should be put on fast disks that are redundantly configured. Additionally, the VIO server will require additional CPU entitlement to support AMS—the heavier the paging rate for the LPARs the more I/O operations there will be and therefore, the more CPU entitlement the VIO server will need. You can segregate out the impact by having a VIO server dedicated to AMS operations and use other VIO servers for network and I/O virtualization.

Active Memory Deduplication

Active memory deduplication is designed to minimize the existence of identical memory pages in main memory. To take advantage of this, the LPARs must be using AMS. Basically PowerVM changes its logical memory map to make all identical logical memory pages point to the same physical memory location.

Active memory deduplication is supported on AIX, IBM i and Linux and, as with AMS, it requires PowerVM Enterprise. All resources must be virtualized and the cores must be in the shared processor pool—this is because AMS is a requirement for active memory deduplication. Although AMS is supported on POWER6 and higher, active memory deduplication requires a POWER7 or higher. When sizing the VIO server, you'll need to add some additional entitlement above what's needed to service the LPARs and for AMS. Section 9.2.2 of the PowerVM Introduction Redbooks publication provides details on this.

Active Memory Expansion

AME is a POWER7 and higher hardware feature that's purchased when the server is ordered or that can be added on later. It does not require PowerVM and is only supported on AIX 6.1 tl06 or higher. AME works fine with virtualized or dedicated resources (I/O, network or cores) and is the simplest of all these options to configure. In particular, AME is used in SAP implementations to make the memory go further.

AME compresses and decompresses memory content to allow an LPAR to do more work with less memory. AME uses CPU resources in the LPAR for the compression so it's a tradeoff between increased CPU usage and less memory. However, as of POWER7+, AME is able to take advantage of the hardware accelerator, which increases potential memory expansion from 100 percent expansion with POWER7 to 125 percent expansion with POWER7+.

AME is very useful and can lead to significant savings in memory. The server can be purchased with AME enabled and you can set the expansion factor for every LPAR to 0. This means it won't do any

expansion but if you decide to change that you can do it dynamically for that LPAR. AME has only two settings—the amount of physical memory for the LPAR (usually the same as the desired memory) and the expansion factor. As an example, if the LPAR has 8 GB of physical memory then an expansion factor of 2 would let the operating system think it actually had 16 GB of memory.

AME splits the memory into two pools—an uncompressed pool and a compressed pool. If the uncompressed pool is not full, then it's business as usual for the LPAR. But when it fills up, then AME starts compressing pages that are infrequently used and moves them to the compressed pool. This is transparent to the applications but requires CPU cycles. Thus, AME is for LPARs that have available CPU resource but need additional memory. To determine the best settings, there's a command called amepat that can be run on any AIX v6.1 system over a period of time. The output from that command provides details on what additional CPU might be needed for various expansion factors and it can also be used to monitor performance.

Active Memory Mirroring

Active memory mirroring is a hypervisor function that was initially only available on the high-end systems (770, 780 and 795). It's now been expanded to the POWER8 E870 and E880. It's supported for AIX, IBM i and Linux LPARs and doesn't care if resources (I/O, network and cores) are dedicated or virtualized. It doesn't require PowerVM. The purpose of active memory mirroring is to protect the system firmware from uncorrectable memory errors, thus avoiding a systemwide outage. Although these kinds of errors are very rare, active memory mirroring maintains two copies of the POWER hypervisor in memory to protect against them. Just remember that this means the hypervisor will require twice as much memory as it would normally need.
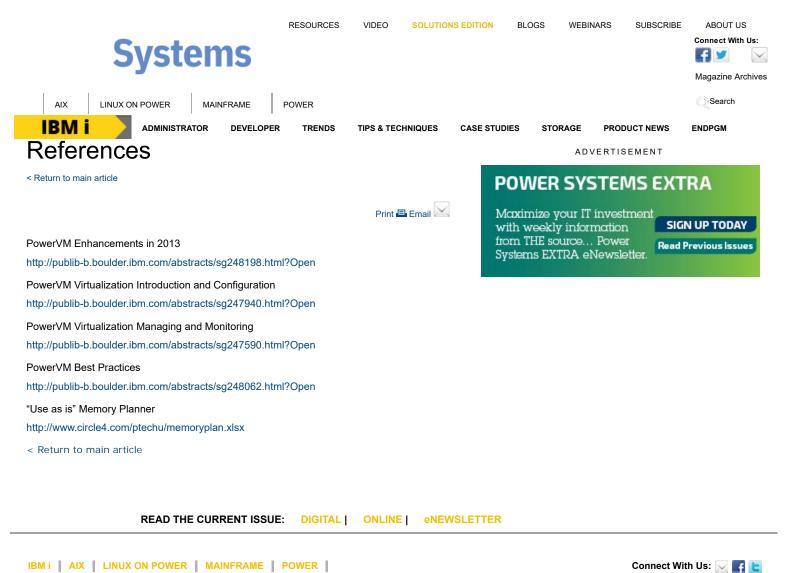
Memory Sizing

Taking advantage of these technologies has an impact on memory sizing as well as CPU sizing. In the case of AMS and active memory deduplication, we can reduce our memory requirements at the cost of additional CPU being needed for the VIO servers. With AME, we again reduce our memory needs but additional CPU is needed in the LPAR that's using AME. And with active memory mirroring, we need to double the hypervisor memory when we do our memory calculations. Memory sizing also needs to take into account high performance cards such as 10 Gb network and 8 Gb fibre cards as well as whether NPIV or vSCSI are being used. I have a spreadsheet that I use to do memory sizing, which tries to estimate how much memory you should have at a minimum for your VIO servers and to support your LPARs. It's a "Use as is" planner (see Resources) but feel free to try it out. Another option is the VIOS Performance Advisor, now the part command in PowerVM.

More to Come

In Part 1, we covered information about virtualization of processors. In this article, we looked at memory. Part 3 will cover virtualization of I/O (network and disk). As you can see, there are now multiple memory planning options and these all need to be taken into account when looking at virtualizing your systems.

Connect With Us:

Magazine Archives

Search

References

< Return to main article

Print 🖶 Email ✉

PowerVM Enhancements in 2013

http://publib-b.boulder.ibm.com/abstracts/sg248198.html?Open

PowerVM Virtualization Introduction and Configuration

http://publib-b.boulder.ibm.com/abstracts/sg247940.html?Open

PowerVM Virtualization Managing and Monitoring

http://publib-b.boulder.ibm.com/abstracts/sg247590.html?Open

PowerVM Best Practices

http://publib-b.boulder.ibm.com/abstracts/sg248062.html?Open

"Use as is" Memory Planner

http://www.circle4.com/ptechu/memoryplan.xlsx

< Return to main article

READ THE CURRENT ISSUE: **DIGITAL** | **ONLINE** | **eNEWSLETTER**

Connect With Us: