TLA overload!

Today
AMS = Active Memory Sharing
– PowerVM feature for Power6 & Power7

NOT
• AME = Active Memory Expansion – AIX only feature
• AEM = Active Energy Manager – part of Systems Director
Why Active Memory Sharing (AMS)?

Don’t Set memory to an Virtual Machine (LPAR) and forget?

We monitor CPU use

- Shared CPU = good for moving cycles where they are needed
- Now the same for memory
- AMS moves Ram to where it does most good

Like:

1. VM in different time zone
2. Day time users and night time batch
3. Many small LPAR but only a few really active

Active Memory Sharing Reference

Redbooks http://www.redbooks.ibm.com/portals/power

PowerVM Active Memory Sharing
New Update June 2011

Movies http://tinyurl.com/AIXMovies - 4 movies
Active Memory Sharing Regular Paging 19 mins
Active Memory Sharing Concepts 16 mins
Active Memory Sharing Setup 16 mins
Active Memory Sharing Simple Monitoring 11 mins
~60 mins
PowerVM Editions are tailored to client needs

<table>
<thead>
<tr>
<th>PowerVM Editions</th>
<th>Express</th>
<th>Standard</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concurrent VMs</td>
<td>VIOS + 2 per VMs</td>
<td>10 per core (up to 1000)</td>
<td>10 per core (up to 1000)</td>
</tr>
<tr>
<td>Virtualization Management</td>
<td>VIM</td>
<td>VIM, HMC</td>
<td>VIM, HMC</td>
</tr>
<tr>
<td>Virtual I/O Server</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>PowerVM Lx86</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Suspend/Resume</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Shared Processor Pools</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Shared Storage Pools</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Thin Provisioning</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Live Partition Mobility</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Active Memory Sharing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Note:
- IBM supports shared storage but does not yet support Suspend & Resume nor LPM.

Active Memory Sharing Media & Installing

There is none!

Note:
- First released in May 2009 on Power6
- At that time required new FW, VIOS & OS
- Now all current
Active Memory Sharing Media & Installing

You need:
1. PowerVM AMS Activated → see HMC
2. VIOS → recent like 2.1.1 or 2.2
3. Operating system that supports AMS like current
   • AIX 6.1 TL3+ (not AIX 5.3)
   • IBM i 6.1.1+ Fixes
   • Linux – current
4. Uses 4 KB pages
5. Virtual Machines
   • Pure Virtual (shared CPU, network & disks)
   • Assumes differing workload pattern but friendly
   • Co-operate in sharing RAM across the system

AMS pages memory between Virtual Machines

So let’s talk about Paging …
Classic Virtual Memory (LPAR)

Not Really Here

Tiny 4 KB pages

Physical Memory

Virtual Memory

Five Paging Golden Rules

1. Don’t do it! → hurts performance
2. Don’t panic! → 10+ pages/s per CPU = noise
3. Do it fast → use many disks
4. Always use Protection → mirror or RAID5
5. Never ever run out of paging space → mayhem!
What is a “working set”?

A 1 GB program has 250,000 pages

- Code=Text
- Data
- Libraries
- Heap
- Stack

Busy running a few functions
Data being loaded/stored
Temporary data being loaded/stored
Function call return data goes here

Working Set is the pages needed to run in the short term (seconds)
Also called Resident Set (resident in memory), see ps or nmon ResText & ResData

AMS acts on Working Sets but at whole LPAR level

OS Level Paging

Memory

Disk

“lrud” daemon

Memory access

Processes
VM Level Paging = AMS

How is it set up?

1. One Paging Device per LPAR i.e. LV/hdisk/LUN
2. VIOS
3. Shared Memory Pool 8 GB

Shared Memory Pool 8 GB

Hypervisor

Logical Partitions

How is it set up?

Hypervisor owns the paging table
How is it set up?

Diagram showing setup with VIOS, shared memory pools, and logical partitions. There are options for dedicated and shared memory configurations.

1. VIOS
2. One Paging Device per LPAR i.e. LV/hdisk/LUN
3. Shared Memory Pool 8 GB
4. Logical Partitions

For dedicated memory setup:

- Dedicate Memory 5 GB
- Dedicated Memory 2 GB

For shared memory setup:

- Shared Memory 5 GB
- Shared Memory 4 GB

Other VM's including VIOS.
AMS Algorithm 1 – It all fits

Assuming many VM sharing a pool & total VM logical memory > pool

Local paging at OS level
Not an issue

“Relaxed Mode”

AMS Algorithm 2 - If it nearly fits?

Hypervisor asks OS images for help
→ once a second
OS then frees memory, if necessary paging out
Loans pages to Hypervisor
Hypervisor gives pages to high demand VM

OS level AMS Tuning on how aggressive:
none/File system cache/programs too

“Co-operative Mode”
AMS Algorithm 3 – Loans are not enough

VMs refuse to loan more memory
→ Hypervisor gets aggressive
1. Finds pages to steals
   - It can see the page tables
   - It avoids critical memory pages
   - Least Recently Used page table data
2. Asks VIOS to page out VM memory
3. Once the memory page is free
4. Gives pages to high demand LPAR

VM’s are not aware of this happening

“Aggressive Mode”

AMS Algorithm 3 – Loans are not enough

Now VM accesses a page that is not present
→ Causes page fault
Normally, Hypervisor hands interrupts to the VM to handle
Checks: if it’s a Hypervisor paged pages
If yes, it recovers the page and restarts the instruction
If no, it passes the page fault onto OS to handle as normal

“Aggressive Mode”
Creating the AMS Pool

Machine Level - Memory Pool on HMC

Shared Memory Pool Management
Making the VM use the Pool
Modify VM to use Shared Memory

So copy the profile 1st.

Can’t be both

Now logical memory i.e. what we would “like”

Now cold stop & restart the Virtual Machine (LPAR) using the new AMS profile

Memory given to the VM on demand so it starts small and grows
Each VM uses 1 AMS paging space
→ So RAID5 LUNS using lots of underlying spindles is good

AIX Level:
vmsstat -h

Watching AMS on a single LPAR is ... “insane”
CEC Level on VIOS or AIX:
topas –C (hit “g” for the extra top info)

Topas CEC Monitor Interval: 10 Wed Dec 3 10:15:06 2008
Partition Info Memory (GB) Processor Virtual Pools : 0
Monitored : 4 Monitored : 8.0 Monitored : 2.0 Avail Pool Proc : 3.7
UnMonitored: - UnMonitored: - UnMonitored: - Shr Physical Busy: 0.28
Shared : 4 Available : - Available : - Ded Physical Busy: 0.00
Uncapped : 4 UnAllocated : - UnAllocated : - Donated Phys CPUs: 0.00
Capped : 0 Consumed : 6.5 Shared : 2 Stolen Phys. CPUs : 0.00
Dedicated : 0 Dedicated : 0 Hypervisor
Donating : 0 Donated : 0 VIRT. Context Switch: 976
Pool Size : 4 Phantom Interrupts : 1

Host OS M Mem InU Lp Us Sy Id PhysB Vcsw Ent %EntC Phl pmem
-------------------------------------shared-------------------------------------
silver_vios1 A61 U 2.0 1.9 8 08 1 0 90 0.01 873 0.40 2.6 0 -
silver_iparp2 A61 UM 2.0 1.4 2 49 1 0 49 0.26 244 0.50 51.3 1 1.30
silver_iparp3 A61 UM 2.0 1.6 2 0 0 0 99 0.01 294 0.50 1.4 0 0.78
silver_iparp4 A61 UM 2.0 1.7 2 0 0 0 99 0.01 220 0.50 1.3 0 0.87
silver_iparp5 A61 UM 2.0 1.7 2 0 0 0 99 0.01 218 0.50 1.1 0 1.01

Active Memory Sharing in Action

Eventually reach a steady state with memory "flowing between LPARs"
Active Memory Sharing - Expectations

Memory pages flow between Virtual Machines

1. Sudden GB’s of memory moved would require massive paging & a large performance hit.
2. Don’t want this for transitory peak.
3. So few MB/s arrive until demand eases off

Classic Virtual Memory (VM level)

Tiny 4 KB pages

Not really here

Physical Memory
 Virtual Memory

Actually stored on disk
Active Shared Virtual Memory (VM Level)

Virtual Memory

Logical Memory

Blue Physical Memory Pages

Hypervisor

Loaned Memory To Hypervisor

Not Really Here

Actually stored on disk

VIOS