Active Memory Sharing @ Australia Post

2009 IBM Power Systems Technical Symposium. 10-13th August, Sydney, Australia

Who is this bloke!?

- ❑ Chris Gibson - chris.gibson@auspost.com.au
- ❑ AIX Specialist @ Australia Post.
- ❑ IBM CATE, System p platform and AIX 5L, technical writer for IBM Systems Magazine, IBM developerWorks and a co-author of the IBM Redbooks publication, "NIM from A to Z in AIX 5L."

Purpose

- ❑ Share our AMS experience with AIX community.
- ❑ Exchange ideas with other AIX customers.
- ❑ Demonstrate use of latest technology, outside of IBM.
- ❑ Provide feedback to IBM development.

Audience

- ❑ Technical
- ❑ AIX

Active Memory Sharing - Early Ship Program

- ❑ Post was nominated and accepted. Had to sign non-disclosure.
- ❑ Commitment to testing the product and providing regular feedback.
- ❑ Non-production POWER6 kit upgrade to beta pre-reqs.
- ❑ IBM high Interest in SAP, DB2 / Oracle, Websphere & WPAR.
- ❑ Benchmark AMS and monitor it's effect on performance.
- ❑ Weekly Feedback - functionality, performance, usability.
- ❑ Phase 1 Simple AMS (Post).
- ❑ Phase 2 AMS with Dual VIOS & Partition Mobility – Couldn't do it at Post. Not on blades.

- ❑ We received:
  - The code/DVD's & Documents.
  - AMS Forum for Q & As from the actual developers.
  - Access to raise PMRs for bugs.

What is Active Memory Sharing?

- ❑ Active Memory Sharing is an enhancement to IBMs PowerVM virtualisation technology.
- ❑ Available on POWER6 platform.
- ❑ It intelligently flows memory from one partition to another for increased utilization & flexibility of memory.
- ❑ A bit like the shared processor pool concept. Not as fast though!
- ❑ Active Memory Sharing (AMS) initially called Virtual Real Memory (VRM). But UNIX has had VRM for 30 years!
- ❑ Some suggested "virtual virtual real memory" ☺
- ❑ AMS built on top of Virtual Memory/Paging.

How does Post benefit?

❑ On the 570, we have:

- ~30 “SOE” LPARs.
- Spare CPU power for extra LPARs.
- No spare memory.

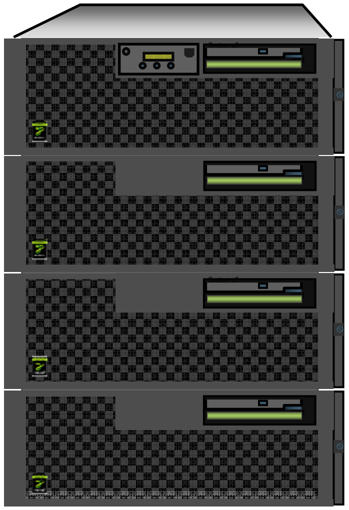
❑ How we design each LPAR: How much memory?

- Policy “*every SAP LPAR gets 8GB of memory*”
- But do they really need that memory?
- Which LPARs could give up some memory?
- Hard to tell due to AIX optimisation of memory!

❑ Analysis shows not all LPARs are busy at the same time:

- Some LPARs busy once a month.
- Some LPARs have occasional use.

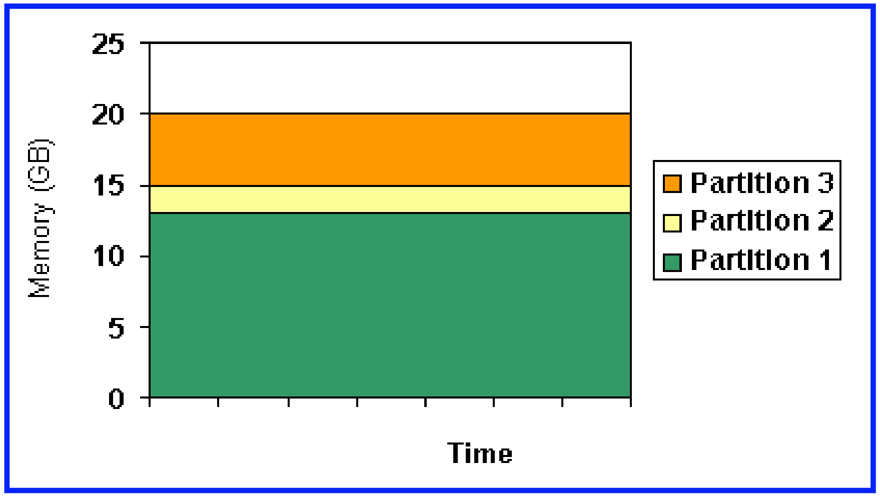
❑ Can we rebalance memory use? YES with AMS.



AMS – Memory Utilisation.

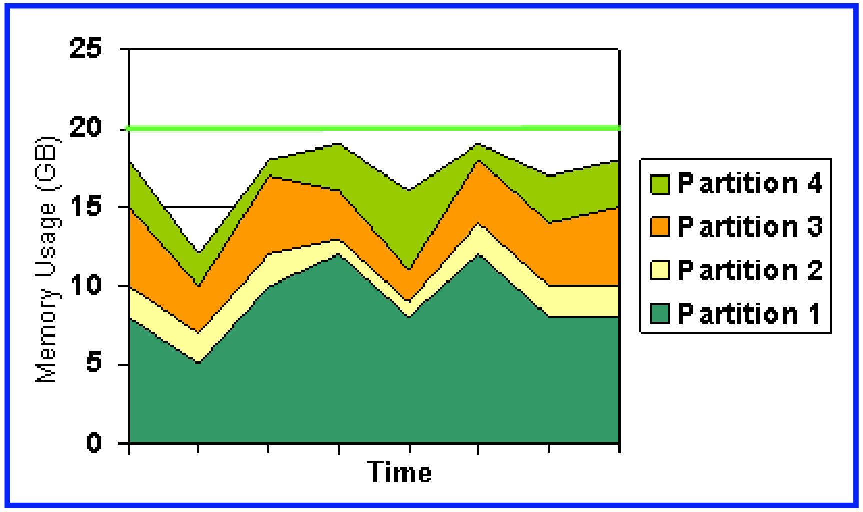
❑ Dedicated memory.

- Each LPAR owns its memory.
- Un-used memory → is wasted.
- Over-used memory → pages to disk ☹.
- Manual dynamic memory change is rare.



❑ Partitions with shared memory.

- Memory is allocated to shared pool.
- Assigned to LPAR "On Demand".
- Un-used memory can be used to build more LPARs.



AMS POC Environment

❑ JS22 Blade. Running VIOS 2.1.0.1 FP 20 and IVM. 16GB Memory and 4 processors.

❑ Two LPARs running AIX v6.1 TL2 SP2 (migrated AP AIX 5.3 SOE image):

- Upgraded the blade firmware to EA340_043_039.
- Upgraded VIOS on blade to 2.1.0.1-FP-20.0.
- Applied AMS efixes to VIOS and AIX LPARs.
- Applied VET code for AMS activation.
- Defined a shared memory pool on the blade.

❑ Shared Memory Pool size = 12GB. Leave some memory for our VIOS and Hypervisor.

❑ Configured two shared memory partitions:

- One LPAR running an instance of SAP/Oracle.
- The other LPAR is running three WPARs for SAP, Wily and Oracle Grid Control.

❑ Memory usage: working set determined via *svmon* prior to switch over to shared memory.

❑ Working set = pages required to run.

- LPAR1 (bxaix85) - 1 SAP Instance - 6GB RAM - 60% wset (3.6GB).
- LPAR2 (bxaix86) – 3 instances. Each instance in WPAR - 8GB - 70% wset (5.7GB).
- 9.3GB working set.

Dedicated to Shared Memory

- ❑ Both LPARs were originally dedicated memory partitions.
- ❑ Converted to shared memory partition by changing LPAR profile.
- ❑ Shared Memory Pool Size 12GB.

LPAR1 6GB

LPAR2 8GB

14GB != 12GB

Does that work?

1. If 2 LPARs started = it fits
2. If Working Set ~ 12 GB → it works
3. If Working Set > 12 GB → paging
4. If Working Set >> 12 GB → lots of paging

AMS Setup

- Paging Virtual I/O Server - provides paging services for a shared memory pool and manages the paging spaces for shared memory partitions.
- Not possible to assign more than one paging Virtual I/O Server to the shared memory pool (at this time).
- Active Memory Sharing enables dynamic memory management among multiple LPARs by allocating memory on demand.
- As a result, the hypervisor has to use a paging device to back up the excess memory that it cannot back up using the physical memory.
- AMS Paging Devices. Auto-config with IVM.

```
$ lsdev | grep Paging
vrmpage0         Available   Paging Device - Logical Volume
vrmpage2         Available   Paging Device - Logical Volume
$ lsvg -lv rootvg | grep vrm
lv00                vrmdevice  64      64      1    open/syncd    N/A
lv02                vrmdevice  64      64      1    open/syncd    N/A
```

AMS – Memory Subscription

- ❑ *Non overcommit*: The amount of real memory available in the shared pool is enough to cover the total amount of logical memory configured.

- ❑ *Logical overcommit*: The logical memory in use at a given time is equal to the physical memory available in the shared memory pool. That is, the total logical configured memory can be higher than the physical memory, however the working set never exceeds the physical memory.

- ❑ *Physical overcommit*: The working set memory requirements can exceed the physical memory in the shared pool. Therefore, logical memory has to be backed by both the physical memory in the pool and by the paging devices. In the case of “over commitment”, the hypervisor backs the excess logical memory using paging devices that are accessed through its paging Virtual I/O Server.

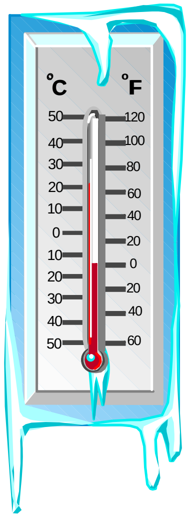
AMS – Workload selections

- ❑ Workloads that are not maximizing physical memory consumption are prime AMS candidates.
- ❑ *Logical overcommit*: For workloads that peak at different times. Have low average memory residency requirements. Do not have sustained loads, such as test and development environments. Failover and backup partitions that are used for redundancy that require resources only when the primary server goes down.
- ❑ Physical overcommit: Workloads that use a lot of AIX file cache. Less sensitive to I/O latency such as file servers, print servers, and network applications. Workloads that are inactive most of the time. NIM?
- ❑ Dedicated memory partitions: Use dedicated memory for workloads that have high quality of service requirements, have high sustained memory consumption, mandate predictable performance and have sustained high CPU utilization and have high memory bandwidth requirements

AMS – Algorithm Part 1

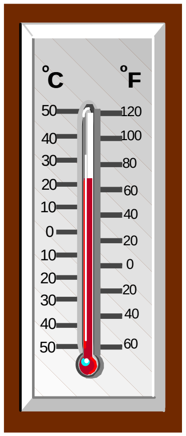
❑ State 1) If it fits:

- Local paging AIX level.
- Not an issue.



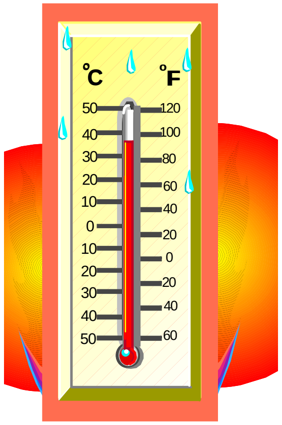
❑ State 2) If it nearly fits? → Co-operative Mode (CMM)

- Hypervisor asks AIX for help once per second.
- AIX then frees memory, if necessary paging out.
- AIX Tuning on how aggressive:
  File system cache , programs too or none.
- Loans pages to Hypervisor.
- Hypervisor gives pages to high demand LPAR.



AMS – Algorithm Part 2

- State 3) If this is not enough?
  - Hypervisor gets aggressive.
  - Steals some pages (Assuming Least Recently Used).
  - Asks VIOS to page memory out.
  - Hypervisor gives pages to high demand LPAR.



- Now LPAR accesses a page that is not present:
  - Causes page fault,
  - Hypervisor checks if it’s a “hypervisor paged” page,
  - If yes, it recovers the page and restarts the instruction,
  - If no, it passes the page fault onto AIX to handle as normal,

- State 4) Buy more memory !!!!!

AMS in action – Part 1

- Memory from bxaix85 has been loaned to bxaix86.

```
kthr          memory                                page                            faults                  cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- ------------------------
  r   b        avm         fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit    pmem    loan
  0   0     685577      263910     0     0     0     0      0     0     6    181   256  1  2 98  0  0.02   3.8     0     0    4.00    2.00
```

- bxaix86 is idle.

```
kthr          memory                                page                            faults                  cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- ------------------------
  r   b        avm         fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit    pmem    loan
  0   0    1897574        2812     0     0     0     0      0     0    22   1212  1066  7  4 89  0  0.05  13.1     0     0    8.00    0.00
```

- Workload starts on bxaix85. The memory that it loaned to bxaix86 is borrowed again.

```
kthr          memory                                page                            faults                  cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- ------------------------
  r   b        avm         fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit    pmem    loan
  0   0     685577      263910     0     0     0     0      0     0     6    181   256  1  2 98  0  0.02   3.8     0     0    4.63    1.37
  …
  1   0     685576      263911     0     0     0     0      0     0     4    232   238  0  2 98  0  0.01   3.6     0     0    4.67    1.34
  …
  1   0     685575      263912     0     0     0     0      0     0     7    174   237  1  2 98  0  0.02   3.9     0     0    4.68    1.31
```

- The working set of both LPARs is now larger than the shared memory pool size. Hypervisor Paging occurs on bxaix86, as memory is given back to bxaix85.

```
kthr          memory                                page                            faults                  cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- ------------------------
r   b        avm          fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec  hpi  hpit    pmem    loan
66   0   1891470         2636     0    47     0     0      0     0    56    870   731 19 26 56  0  0.18  45.2  414  1779    6.44    0.00
```

AMS in action – Part 2

- Hypervisor Paging stops on bxaix86 once bxaix85 has enough memory to complete it’s work.

```
 kthr          memory                         page                       faults                 cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- -------------------------
  r   b        avm        fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit   pmem   loan
  0   0    1891176       9083     0     0     0     0      0     0    35   1226  1213 10  5 86  0  0.06  16.2    12    34   6.57   0.00
  0   0    1891175       9084     0     0     0     0      0     0    38   1149  1109  9  5 86  0  0.06  15.2     0     0   6.57   0.00
```

- Once bxaix85 is finished it’s workload, a job starts on bxaix86. Memory is, again, loaned out from bxaix85 to bxaix86.

```
kthr          memory                         page                       faults                 cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- -------------------------
  r   b        avm        fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit   pmem   loan
  1   0     694653     135090     0     0     0     0      0     0     4    153   233  0  1 98  0  0.01   3.4     0     0   5.43   0.57
  1   0     694653     135090     0     0     0     0      0     0     6    127   246  0  2 98  0  0.01   3.6     0     0   5.43   0.57

System configuration: lcpu=8 mem=6144MB ent=0.40 mmode=shared mpsz=12.00GB

 kthr          memory                         page                       faults                 cpu                   hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- -------------------------
  r   b        avm        fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit   pmem   loan
  0   0     742914       1054     0     0     0   128    128     0     3    404   258  1  3 96  0  0.02   5.5     0     0   4.94   1.06
```

AMS in action – Part 3

- AMS Loan Policy can be changed using vmo.

```
# vmo -L ams_loan_policy
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
ams_loan_policy           1      1      1      0      2      numeric           D
--------------------------------------------------------------------------------

# vmo -h ams_loan_policy
Help for tunable ams_loan_policy:
Purpose:
This tunable toggles the loaning behavior when shared memory mode is enabled.
Values:
        Default: 1
        Range: 0 - 2
        Type: Dynamic
        Unit: numeric
Tuning:
When the tunable is set to 0, loaning is disabled. When set to 1, loaning of file cache
  is enabled. When set to 2, loaning of any type of data is enabled. In response to low
  memory in the AMS pool, the VMM will free memory and loan it to the hypervisor.
```

IBM Early Ship Program experience.

- ❑ Challenging! Finding the time to test it! Finding the most optimal configuration and settings was difficult at first. Along with determining the working set of an LPAR. But with experience it becomes easier.
- ❑ The excellent documentation provided made this process much smoother! The 'Performance White Paper' and the 'Red Paper' were the best sources of information.
- ❑ Ease of configuration. The potential for no more "wasted" or "idle" memory. Makes you wonder "Why didn't we (IBM) think of this sooner?" ☺
- ❑ For our non-prod systems (20-30 LPARS), it is a good fit. Many of our non-prod systems remain idle for extended periods of time. Only a few are busy. Being able to direct "idle" memory away from LPARs that don't need it right now, to LPARs that do, is a amazing!
- ❑ Provide input into best practices guide. Real-world scenarios and advice. Show typical best practice configurations for a variety of workloads. Focus on commercial environments (in particular SAP systems!).
- ❑ Support for multiple shared memory pools (option to create several shared memory pools in one system) – Separate prod from non-prod. Isolate certain workloads from each other.

FAQ - Pre-Requisites for Non-Prod @ Post

- ❑ Required hardware, AIX version, VIOS version and Firmware for our non-production systems:
  - ✓ An IBM Power System based on the POWER6 processor.
  - ✓ Enterprise PowerVM activation for Active Memory Sharing.
  - ✗ Firmware level 340_070.
  - ✓ HMC version 7.3.4 for HMC managed systems.
  - ✗ Virtual I/O Server Version 2.1.0.1-FP20.0.
  - ✗ AIX 6.1 TL3
  - ✓ Micro-partition only
  - ✓ Virtual I/O only

FAQ – Supported Configurations & Future Plans

1. Dual Paging VIOS supported (non-blade env)? - Now supported.

2. LPM + AMS supported? Blade env? - Now supported but not used by any customers yet.

3. How many users of AMS in beta? Prod? - 10 customers part of the beta. Germany=6, Austria=1, USA=2, Australia=1. None using it production at GA.

4. Typical configurations/usage seen thus far? - Small configurations - 2 or 3 LPARs. All SAP/Oracle!

5. Applications? SAP/Oracle.

6. WPARs? German customer - large WPAR site.

7. Future plans - new features? Multiple Shared Memory Pools.

8. Will there be a best practice Redbook at some point in the future? Real world scenarios? Highlight best candidates for AMS? Future Redbook update planned.

AMS - To do list.

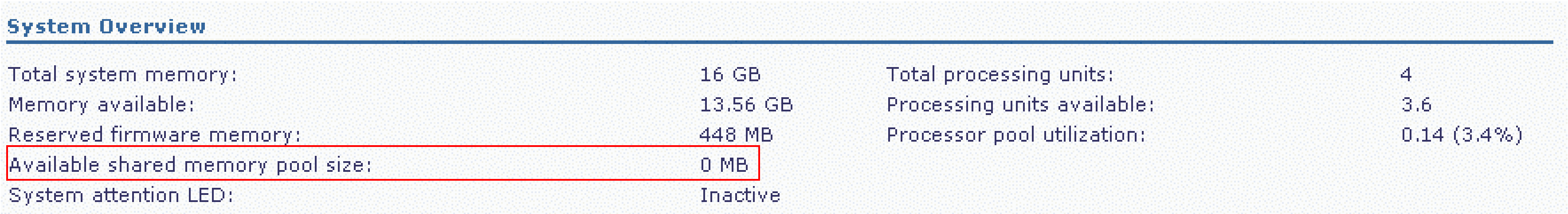
- ❑ Update our HMCs to latest level.
- ❑ Update firmware on all our POWER6 systems.
- ❑ Upgrade VIO servers to version 2.1.
- ❑ Upgrade all our LPARs to AIX 6.1.
- ❑ Produce a migration strategy for moving to AMS in non-prod i.e. handful of LPARs, test for several months. Migrate remaining LPARs on 570 to AMS? 595-2 non-prod LPARs next?
- ❑ HACMP standby LPARs also potential candidates for AMS. The Primary HA LPARs could have dedicated memory, while the Standby LPARs could have shared memory, allowing it to share memory with other LPARs. Less "idle" memory.
- ❑ Test with dual VIOS, Partition Mobility (and HACMP?).
- ❑ Training on AMS. Need to understand performance implications in a virtualised memory environment.

AMS Configuration Part 1.

- Configuring AMS is a good way to learn how it works!
- Enter AMS VET code. Verify applied OK.

```
$ lsvet -t hist | grep Memory
time_stamp=02/25/2009 23:24:31,entry=[VIOSI0500042C-0617] Active Memory Sharing enabled.
```

- Prior to creating a shared memory pool.

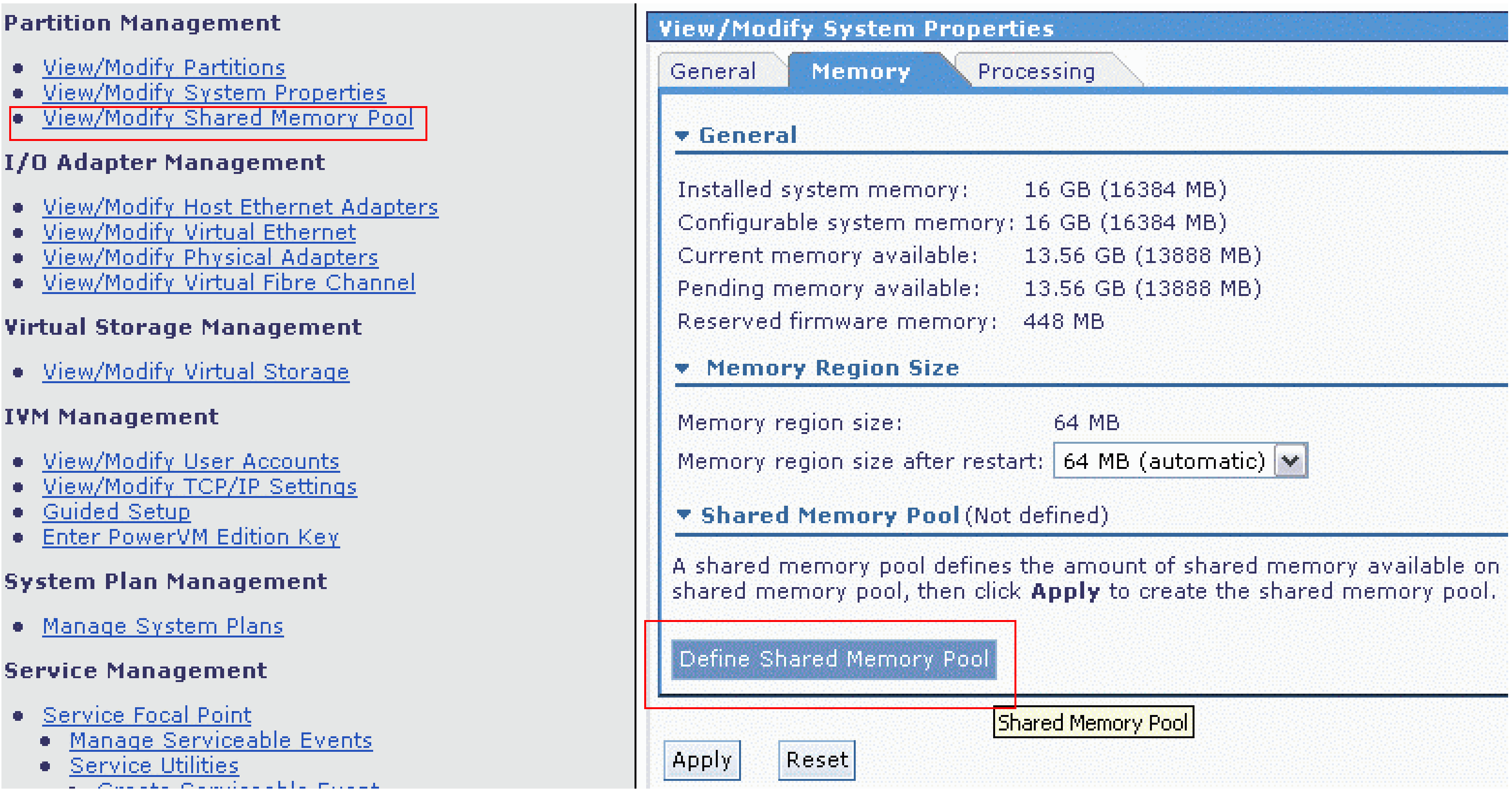


System Overview

Total system memory:	16 GB	Total processing units:	4
Memory available:	13.56 GB	Processing units available:	3.6
Reserved firmware memory:	448 MB	Processor pool utilization:	0.14 (3.4%)
Available shared memory pool size:	0 MB		
System attention LED:	Inactive		

AMS Configuration – Part 2.

- Creating the shared memory pool.



Partition Management

- View/Modify Partitions
- View/Modify System Properties
- View/Modify Shared Memory Pool

I/O Adapter Management

- View/Modify Host Ethernet Adapters
- View/Modify Virtual Ethernet
- View/Modify Physical Adapters
- View/Modify Virtual Fibre Channel

Virtual Storage Management

- View/Modify Virtual Storage

IVM Management

- View/Modify User Accounts
- View/Modify TCP/IP Settings
- Guided Setup
- Enter PowerVM Edition Key

System Plan Management

- Manage System Plans

Service Management

- Service Focal Point
  - Manage Serviceable Events
  - Service Utilities
    - Create Serviceable Event

View/Modify System Properties

General

Installed system memory: 16 GB (16384 MB)
Configurable system memory: 16 GB (16384 MB)
Current memory available: 13.56 GB (13888 MB)
Pending memory available: 13.56 GB (13888 MB)
Reserved firmware memory: 448 MB

Memory Region Size

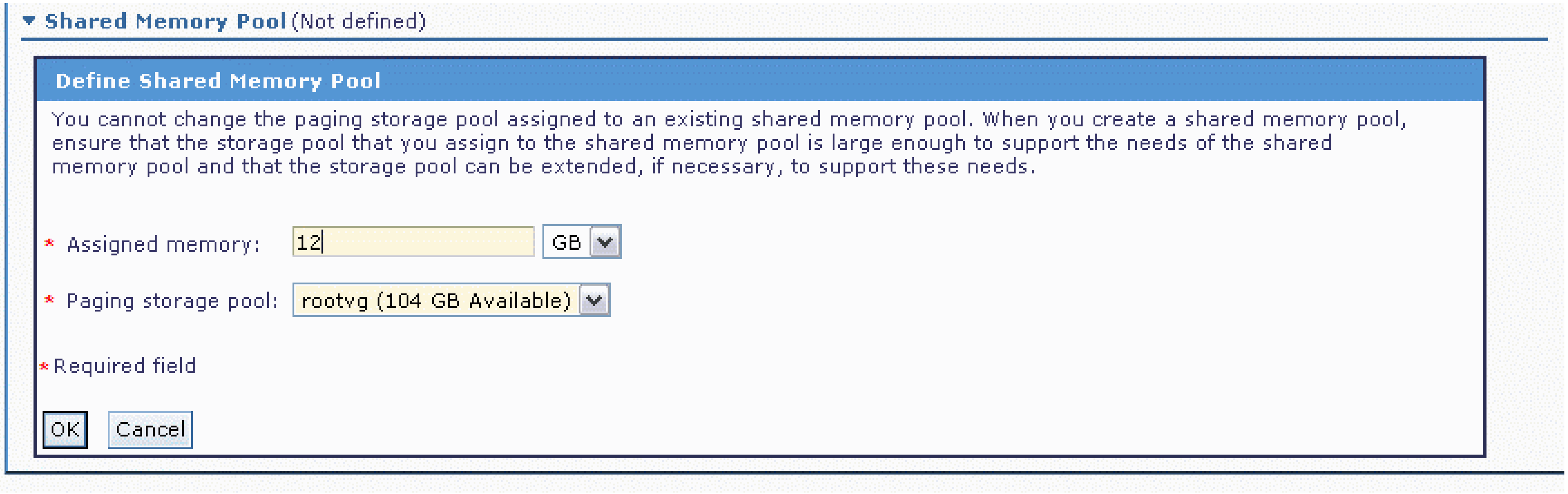
Memory region size: 64 MB
Memory region size after restart: 64 MB (automatic)

Shared Memory Pool (Not defined)

A shared memory pool defines the amount of shared memory available on shared memory pool, then click **Apply** to create the shared memory pool.

AMS Configuration – Part 3.

- Defining the shared memory pool size and paging device location.
- rootvg location for hypervisor paging devices. Recommend SAN.



▼ Shared Memory Pool (Not defined)

Define Shared Memory Pool

You cannot change the paging storage pool assigned to an existing shared memory pool. When you create a shared memory pool, ensure that the storage pool that you assign to the shared memory pool is large enough to support the needs of the shared memory pool and that the storage pool can be extended, if necessary, to support these needs.

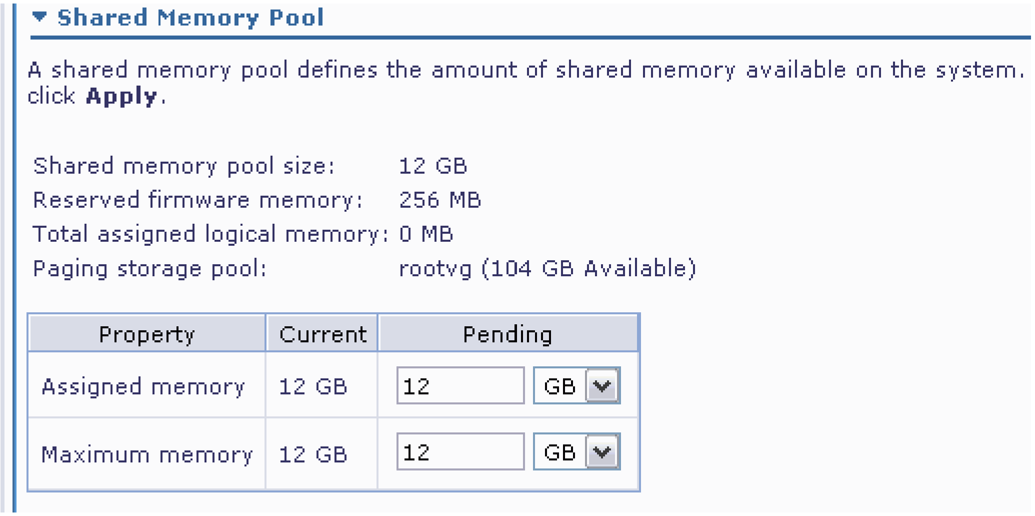
* Assigned memory: 12 GB

* Paging storage pool: rootvg (104 GB Available)

*Required field

AMS Configuration – Part 4.

- Shared memory pool settings.



▼ Shared Memory Pool

A shared memory pool defines the amount of shared memory available on the system.
click **Apply**.

Shared memory pool size:	12 GB
Reserved firmware memory:	256 MB
Total assigned logical memory:	0 MB
Paging storage pool:	rootvg (104 GB Available)

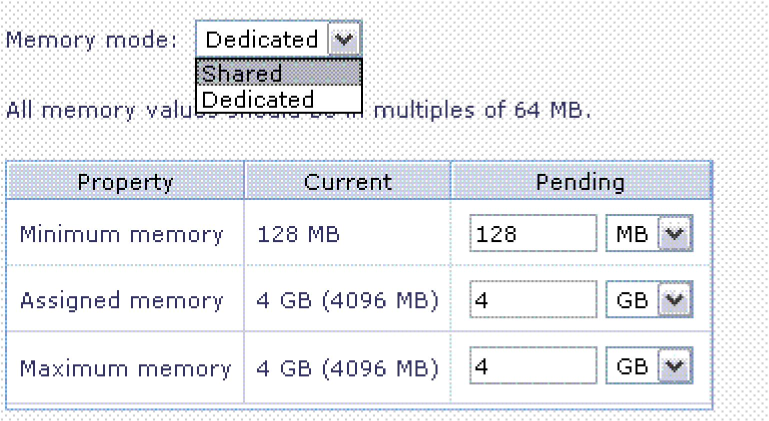
Property	Current	Pending
Assigned memory	12 GB	12 GB
Maximum memory	12 GB	12 GB

- Shared memory pool view from the VIOS/IVM.

```
$ lshwres -r mempool -F curr_pool_mem,paging_storage_pool
12288,rootvg
```

AMS Configuration – Part 5.

- Switch LPAR from dedicated to shared memory. Shutdown LPAR first. Change the profile.

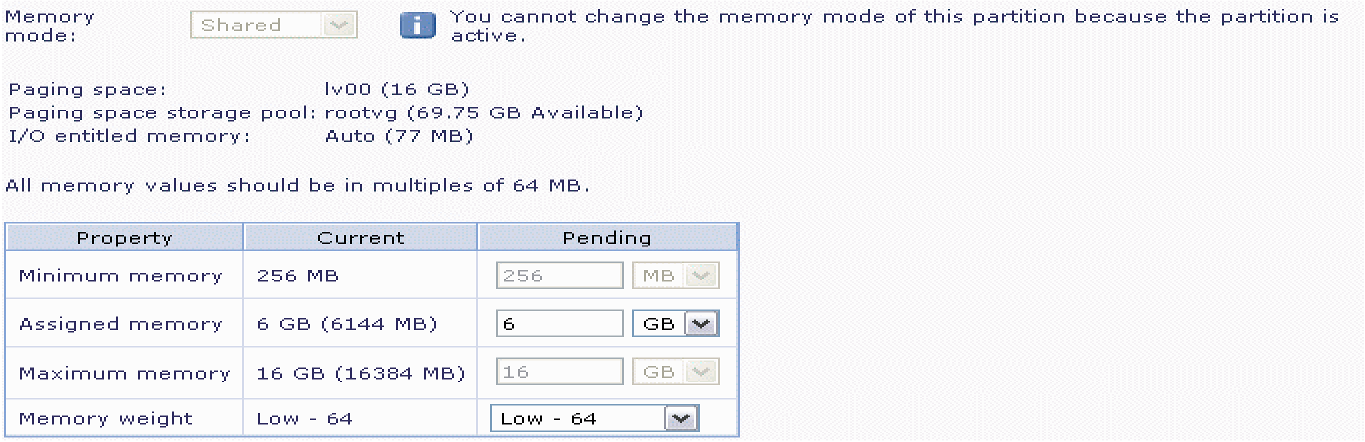


Memory mode: Dedicated

All memory valu multiples of 64 MB.

Property	Current	Pending
Minimum memory	128 MB	128 MB
Assigned memory	4 GB (4096 MB)	4 GB
Maximum memory	4 GB (4096 MB)	4 GB

- Shared memory partition profile settings.



Memory mode: Shared

You cannot change the memory mode of this partition because the partition is active.

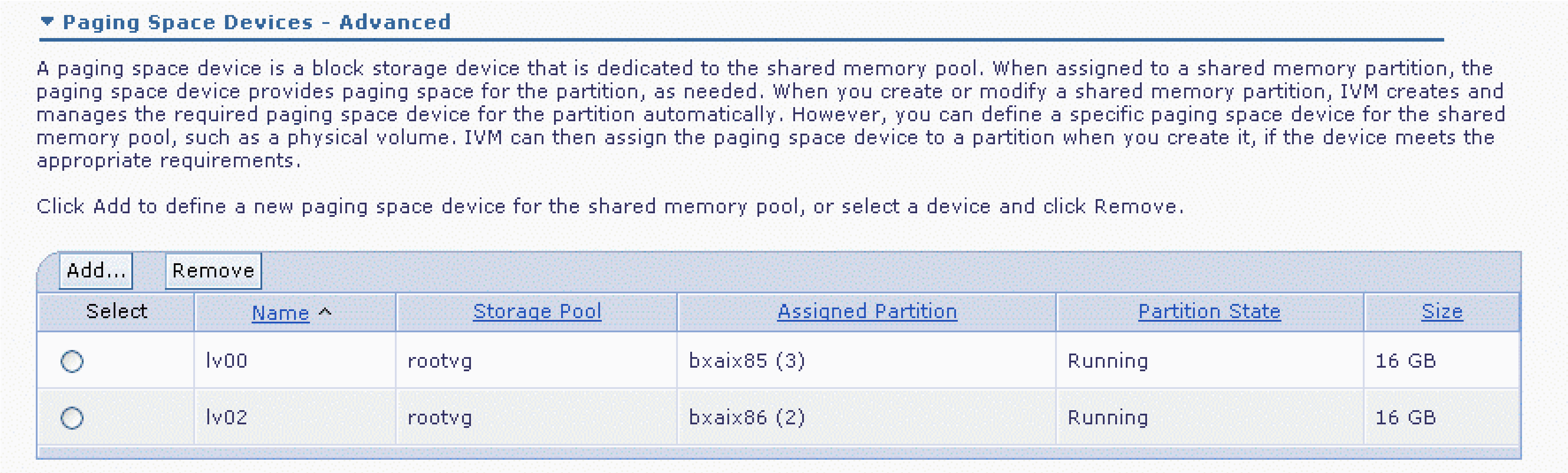
```
Paging space:              lv00 (16 GB)
Paging space storage pool: rootvg (69.75 GB Available)
I/O entitled memory:       Auto (77 MB)
```

All memory values should be in multiples of 64 MB.

Property	Current	Pending
Minimum memory	256 MB	256 MB
Assigned memory	6 GB (6144 MB)	6 GB
Maximum memory	16 GB (16384 MB)	16 GB
Memory weight	Low - 64	Low - 64

AMS Configuration – Part 6.

- Switch LPAR from shared to dedicated memory. Shutdown LPAR first. Change the profile.

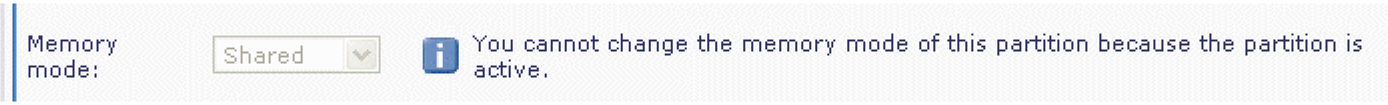


A paging space device is a block storage device that is dedicated to the shared memory pool. When assigned to a shared memory partition, the paging space device provides paging space for the partition, as needed. When you create or modify a shared memory partition, IVM creates and manages the required paging space device for the partition automatically. However, you can define a specific paging space device for the shared memory pool, such as a physical volume. IVM can then assign the paging space device to a partition when you create it, if the device meets the appropriate requirements.

Click Add to define a new paging space device for the shared memory pool, or select a device and click Remove.

Select	Name ^	Storage Pool	Assigned Partition	Partition State	Size
○	lv00	rootvg	bxaix85 (3)	Running	16 GB
○	lv02	rootvg	bxaix86 (2)	Running	16 GB

- Switch LPAR from shared to dedicated memory. Shutdown LPAR first. Change the profile.



Memory mode: Shared

You cannot change the memory mode of this partition because the partition is active.

AMS Configuration – Part 7.

- ❑ AMS paging devices. View from VIOS.

```
$ lsvg -lv rootvg | grep vrm
lv00                vrmdevice  64      64      1    open/syncd    N/A
lv02                vrmdevice  64      64      1    open/syncd    N/A

$ lsdev | grep vrm
vrmpage0         Available   Paging Device - Logical Volume
vrmpage2         Available   Paging Device - Logical Volume

$ lsdev -dev vrmpage0 -attr
attribute       value                    description                  user_settable
LogicalUnitAddr 0x8100000000000000       Logical Unit Address         False
aix_tdev        lv00                     Target Device Name           False
redundant_usage no                       Redundant Usage              True
storage_pool    rootvg                   Storage Pool                 False
vasi_drc_name   U7998.61X.10071DA-V1-C15 VASI DRC Name                True
vrm_state       active                   Virtual Real Memory State    True
vtd_handle      0x100016e24678d          Virtual Target Device Handle False
```

AMS Configuration – Part 8.

- ❑ lparstat and vmstat output from shared memory partition.

```
$ lparstat -i | grep -i memory
Online Memory                              : 4096 MB
Maximum Memory                             : 16384 MB
Minimum Memory                             : 256 MB
Memory Mode                                : Shared
Total I/O Memory Entitlement*              : 77.000 MB
Variable Memory Capacity Weight**          : 64
Memory Pool ID                             : 0
Physical Memory in the Pool                : 12.000 GB
```

*The I/O entitled memory represents the maximum amount of physical memory that is guaranteed to be available for I/O mapping by a partition at any given time.

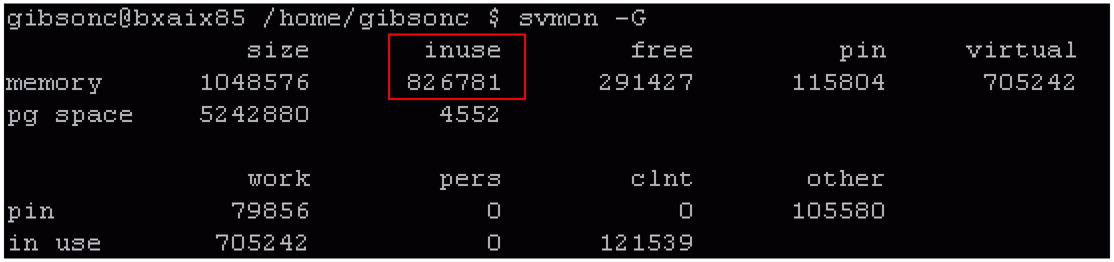
**Partitions are given weight to enforce priority in allocating memory

```
$ vmstat -hw
System configuration: lcpu=8 mem=4096MB ent=0.40 mmode=shared mpsz=12.00GB

 kthr          memory                              page                              faults                 cpu                      hypv-page
------- --------------------- ------------------------------------ ------------------ ----------------------- -------------------------
  r   b        avm        fre    re    pi    po    fr     sr    cy    in     sy    cs us sy id wa    pc    ec   hpi  hpit   pmem   loan
  1   1     705237     288891     0     0     0     0      2     0    16   1214   268  0  0 99  0  0.00   0.2  5968 13564   4.00   0.00
```

AMS Monitoring.

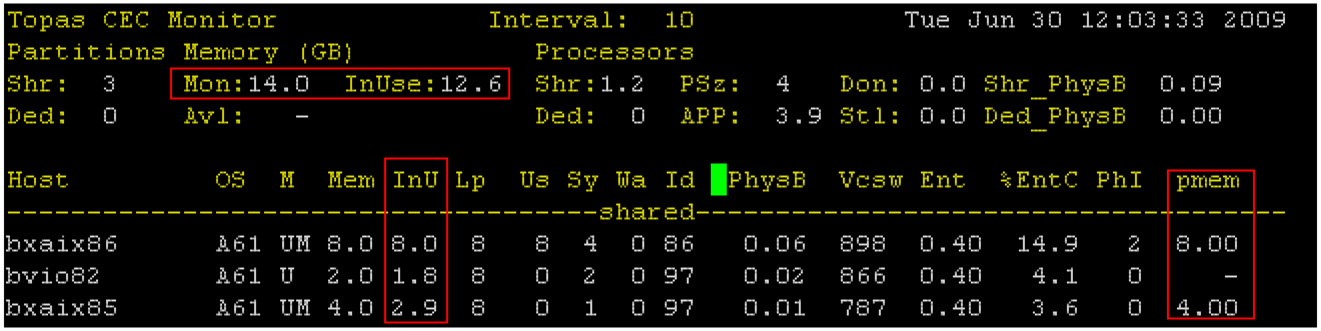
- In a dedicated memory partition, svmon can be used to measure the working set size. The command “svmon –G” shows the “inuse” memory value.



```
gibsonc@bxaix85 /home/gibsonc $ svmon -G
               size       inuse        free         pin     virtual
memory      1048576      826781      291427      115804      705242
pg space    5242880        4552

               work        pers        clnt       other
pin           79856           0           0      105580
in use       705242           0      121539
```

- Existing tools such as topas and vmstat have been enhanced to report physical memory in use, hypervisor paging rate, hypervisor paging rate latency, and the amount of memory loaned by AIX to the hypervisor.



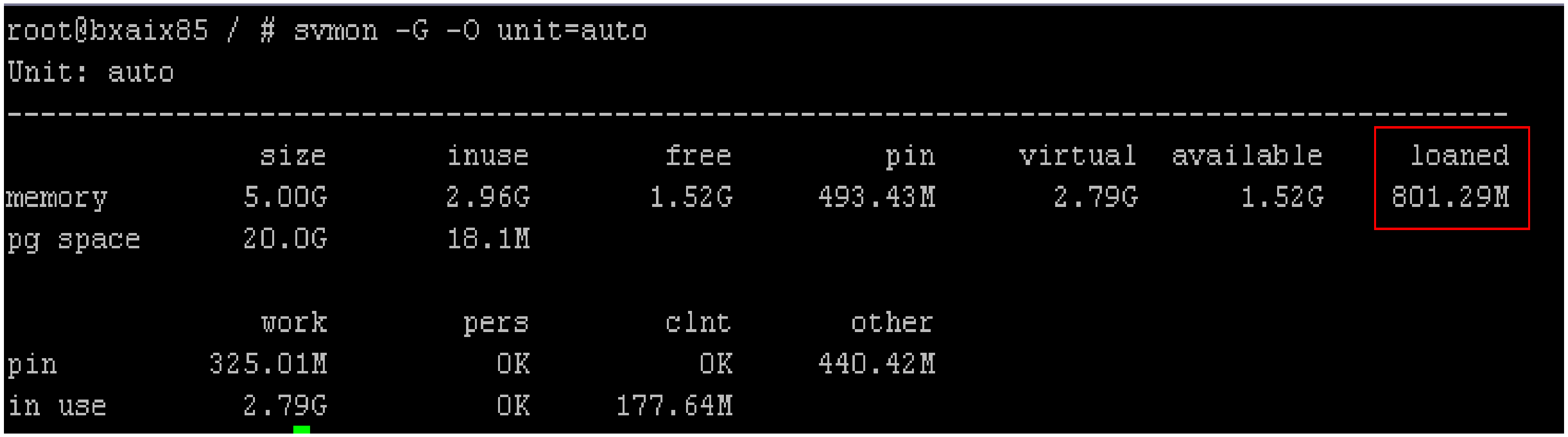
```
Topas CEC Monitor              Interval:  10             Tue Jun 30 12:03:33 2009
Partitions Memory (GB)            Processors
Shr:   3    Mon:14.0  InUse:12.6  Shr:1.2  PSz:   4   Don: 0.0 Shr_PhysB  0.09
Ded:   0    Avl:    -             Ded:   0 APP:   3.9 Stl: 0.0 Ded_PhysB  0.00

Host         OS  M  Mem  InU Lp  Us Sy Wa Id  PhysB  Vcsw Ent  %EntC PhI  pmem
-------------------------------------shared-------------------------------------
bxaix86      A61 UM 8.0 8.0  8   8  4  0 86   0.06  898 0.40  14.9   2  8.00
bvio82       A61 U  2.0 1.8  8   0  2  0 97   0.02  866 0.40   4.1   0     -
bxaix85      A61 UM 4.0 2.9  8   0  1  0 97   0.01  787 0.40   3.6   0  4.00
```

pmem : Physical memory in GBytes allocated to shared memory partitions from the shared memory pool at a given time.

AMS Monitoring - continued.

- svmon is also AMS aware.



```
root@bxaix85 / # svmon -G -O unit=auto
Unit: auto
-----------------------------------------------------------------------------------------
               size       inuse        free         pin     virtual  available     loaned
memory        5.00G       2.96G       1.52G     493.43M       2.79G      1.52G    801.29M
pg space      20.0G       18.1M

               work        pers        clnt       other
pin         325.01M          0K          0K     440.42M
in use        2.79G          0K     177.64M
```

AMS is here! Now what?

- ❑ Whoopee! The feature we have all being waiting for is here - Active Memory Sharing.
- ❑ It is the final piece of the jigsaw to creating a fully virtualised environment.
- ❑ We can now oversubscribe memory on a POWER6 system and let the system deploy memory where we need it. No more DLPAR operations required!
- ❑ Still a lot to learn! Performance tuning and monitoring changes.
- ❑ Traditional AIX memory monitoring will need to be widened. New considerations with AMS and logical memory.
- ❑ Need to adjust our perspective on monitoring and managing memory…just like we did when shared processor LPARs were introduced.
- ❑ Plan for the migration to AMS.

AMS References.

- ❑ PowerVM Virtualization Active Memory Sharing Redpaper – Introduces Active Memory Sharing on IBM Power Systems based on POWER6 Processor Technology:

  http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp4470.html?Open

- ❑ IBM PowerVM Active Memory Sharing Performance White paper – This white paper provides guidance on workload selection and workload consolidation along with performance best practices for AMS:

  http://www-03.ibm.com/systems/power/software/virtualization/whitepapers/ams_perf.html

- ❑ AIX6 & POWER6 Hands-On Technical Demo Movies – Look for demos on AMS concepts, setup and monitoring:

  http://www.ibm.com/developerworks/wikis/display/WikiPtype/Movies

- ❑ Configuring Active Memory Sharing – A customer’s experience.

  http://www.ibm.com/developerworks/aix/library/au-pwr6_ams