Active Memory Expansion
for AIX 6 & 7

Nigel Griffiths
IBM Power Systems

Active Memory Expansion
Announcement Confusion

1. Marketing thought:
   – “Expansion” sounded better than “Compression”

2. It does not “just compress memory pages”
   – It’s a lot cleverer than that!

3. “AME” also used for AIX Management Edition
   – So ActMemExp was used ….. “doh!”
     – Now AME = Active Memory Expansion
Active Memory Expansion Pre-Reqs:

POWER7 based machine
AIX 6.1 TL04 SP2+

Also note:
- Transparent to all applications
- Not IVM - Activation key via the HMC
  - But configured at LPAR level
- AME will switch off AIX 64KB page support
  - Can be enabled but tests showed it was slower

Permanent Enablement - Chargeable

- One feature per server
  - No matter how many partitions (LPARs) use it
  - Permanent enablement → new server or via MES order
  - Enablement “VET” code applied to the VPD anchor card
  - Once enabled: no mechanism to move it to a different server

- Power 750 & Power 755
  - #4792 AME Enablement Feature

- Power 770 & Power 780
  - #4791 AME Enablement Feature
  - It is a processor feature !!!

- One-time, 60-day Trial - No charge
  - Request via Capacity on Demand Web page
    www.ibm.com/systems/power/hardware/cod/
How do we switch AME on?

Is the machine AME Capable?

HMC
→ Server Properties
⇒ Capabilities
then scroll to the bottom
Activate on AME on the LPAR profile

Hard reboot (not restart) to activate LPAR in AME mode

Expansion Factor:
- 1.0 = AME on but inactive
- 1.2 to 1.5 = Good start point
- 10.0 = suicidal!

Dynamically changing the Expansion Factor

Use Dynamic LPAR Memory Add/Remove and change the Expansion Factor
How does AME work?

AME Conceptual Model

Memory Pages
AME Conceptual Model

Memory Pages

RAM Disk

Not actually a RAM disk
but similar concept

Use it like a very fast paging device

Page in “on demand”

Page out LRU pages

LRU = Least Recently Used = oldest unused
AME Conceptual Model

Memory Pages

Now while paging, shrink the memory pages so many more pages fit

15 true memory → 11 + 24 = 35 so Expansion Factor = 15:35 = 2.33

AME Conceptual Model

Memory Pages

Dynamically adjusted depending on compression ratio & target
AME Practicalities

Lower Expansion Factor
- Compress once little used pages
- Near zero CPU cycles

Higher Expansion Factor
- More compression
- More CPU cycles

Balance more RAM versus more CPU cycles

Technical Details
Bad Compression Targets

- AIX Kernel
  - Not a AME target

- Filesystem cache, code or memory mapped files
  - Best to page out to filesystems
  - Performance tools → “numperm”

- Pinned Memory
  - Pinned = never page out (AME is like paging)
  - Performance tools → “pinned pages”

- So what can AME compress?

Good Compression Targets

- Mostly private pages within programs
  - Data
  - Heap
  - Stack
  - Not the code
Excellent Compression Targets

- Data that compresses well
  - Data only used on program initialisation
  - Pages allocated but unused = full of zeros/blanks
  - Pages with lots of repeat data like database records

Access Pattern

- Some hot pages, some warm, some freezing
- All pages equally used (HPC) – not so good

How can I work that out?

Do I have
→ Good or bad compression ratio?
→ Friendly or hostile access pattern?

Normally, you can't !!

until now ....
Planning for Active Memory Expansion

- A new AIX command: `amepat`
  - Active Memory Expansion Performance Analysis Tool
  - Or someone called Patrick/Patricia – you decide!

- Scans actual memory use
  - Determines compression ratio & CPU requirement

- With AME on or AME off
  - AIX 6.1 TL04 SP2+ also works on POWER4/5/6/7

What is your Plan?

Memory Shrinking or
Memory Growing

10GB

Looks like 10GB but is actually 8GB, thanks to AME

2GB released for another LPAR

10GB

But want 14GB to improve performance

Actually still using 10 GB but looks like 14 GB, thanks to AME
- Busy processor cores don’t have resources to spare for AME
- The Expansion Factor “knee” depends on the compressibility of memory

---

**amepat command - Don’t worry as its easy**

```
# amepat -?
Usage: amepat [-u minucomp_poolsize] [-m min_mem_savings]
    {[-t tgt_expmem_size] | [-a]} [-n num_entries] [-P recfile]
    {[-e startexpfactor[:stopexpfactor[:incexpfactor]]]|%
    [-c max_cpu_overhead%] | [-C max_cpu_overhead] )}
    [-v] { [Duration] | [interval [samples]] }

amepat -R recfile { [Duration] | [Interval [Samples]] }
amepat -N {{[-P|-R] recfile} [-v] {Duration|[Interval [Samples]]}}

-m min_mem_savings Unit is MB
-c max_cpu_overhead% Unit is percentage
-C max_cpu_overhead Unit is in number of Physical Processors
-u minucomp_poolsize Unit is MB
-t tgt_expmem_size Unit is MB
Duration Unit is minutes
interval Unit is minutes
```

Note: -N flag will turn off Active Memory Expansion Modeling. All options except -P, -R will be disabled when -N is used.
amepat - Basics are Easy

Run & report mode gets frustrating so …

Capture your busy hour for the whole hour
   – amepat –R ame.out 60  [60 minutes]

Then try various reports
   ▪ Shrink memory:
     – amepat –P ame.out
   ▪ Expand memory:
     – amepat –P ame.out –t 4096  [target memory size in MB]

Warning ....

   ▪ Small micro-partition example
     – Less than a whole CPU & only 1 GB memory
     – Easier to generate workload to use all memory

   ▪ Typically, LPARs are much larger
     – Rule of Thumb: 8 -16 GB per CPU or higher

   ▪ Large memory LPARs will give AME more scope
**amepat output – Machine Summary**

```plaintext
# amepat
Date/Time of invocation    : -
Total Monitored time       : NA
Total Samples Collected   : NA
```

**Machine Summary**

System Configuration:

- **Partition Name**: diamond3
- **Processor Implementation Mode**: POWER7
- **Number Of Logical CPUs**: 16
- **Processor Entitled Capacity**: 0.80
- **Processor Max. Capacity**: 4.00
- **True Memory**: 1.00 GB
- **SMT Threads**: 4
- **Shared Processor Mode**: Enabled-Uncapped
- **Active Memory Sharing**: Disabled
- **Active Memory Expansion**: Enabled
- **Target Expanded Memory Size**: 1.00 GB
- **Target Memory Expansion factor**: 1.00

**Memory Summary**

Not compressed by AME

1 GB = very small for my test case

Factor=1 \(\Rightarrow\) No Compression

**AME Statistics**: Current

- **AME CPU Usage (Phy. Proc Units)**: 0.00 [ 0%]
- **Compressed Memory (MB)**: 0 [ 0%]
- **Compression Ratio**: 2.28

**Active Memory Expansion Modeled Statistics**

- **Model Expanded Memory Size**: 1.00 GB
- **Average Compression Ratio**: 2.28

<table>
<thead>
<tr>
<th>Expansion Factor</th>
<th>Modeled True Memory Size</th>
<th>Modeled Memory Gain</th>
<th>CPU Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.14</td>
<td>896.00 MB</td>
<td>128.00 MB [ 14%]</td>
<td>0.00 [ 0%]</td>
</tr>
<tr>
<td>1.33</td>
<td>768.00 MB</td>
<td>256.00 MB [ 33%]</td>
<td>0.00 [ 0%]</td>
</tr>
</tbody>
</table>

**Active Memory Expansion Recommendation**

- The recommended AME configuration for this workload is to configure the LPAR with a memory size of 768.00 MB and to configure a memory expansion factor of 1.33. This will result in a memory gain of 33%.
- With this configuration, the estimated CPU usage due to AME is approximately 0.00 physical processors, and the estimated overall peak CPU resource required for the LPAR is 0.02 physical processors.

**Note**: it does no try below 512 MB – that is just too small.
Active Memory Expansion

AME Statistics:

<table>
<thead>
<tr>
<th></th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>AME CPU Usage (Phy. Proc Units)</td>
<td>0.00 [ 0%]</td>
</tr>
<tr>
<td>Compressed Memory (MB)</td>
<td>0.3 [ 0%]</td>
</tr>
<tr>
<td>Compression Ratio</td>
<td>2.02</td>
</tr>
</tbody>
</table>

Active Memory Expansion Modeled Statistics:

<table>
<thead>
<tr>
<th>Expansion Factor</th>
<th>Modeled True</th>
<th>Modeled CPU Usage</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Memory Size</td>
<td>Memory Gain</td>
<td></td>
</tr>
</tbody>
</table>
| 1.00             | 1.00 GB      | 0.00 KB [ 0%]     | 0.00 [ 0%]
| 1.14             | 896.00 MB    | 128.00 MB [14%]   | 0.34 [ 8%] |
| 1.33             | 768.00 MB    | 256.00 MB [33%]   | 0.72 [18%] |

Active Memory Expansion Recommendation:

The recommended AME configuration for this workload is to configure the LPAR with a memory size of 896.00 MB and to configure a memory expansion factor of 1.14. This will result in a memory gain of 14%. With this configuration, the estimated CPU usage due to AME is approximately 0.34 physical processors, and the estimated overall peak CPU resource required for the LPAR is 1.18 physical processors.

Various combinations

with increasing factor
with decreasing RAM
with rising CPU use

AME thinks
remove 0.125 GB of RAM is OK with 0.34 CPU used

compression achievable

Nothing Compressed as there is no need

Good compression

achievable

Various combinations

to get to 1.5GB
with decreasing RAM

with increasing CPU

Estimating for 1.5GB

Compression found for current memory content

AME thinks
0.28 CPU for 0.5 GB RAM is a good trade-off = last combination

You have to make up your own mind!
Monitoring Active Memory Expansion in use

Iparstat -i

RAM= True Memory

Target
Expansion Factor
AME memory
vmstat –c 1

<table>
<thead>
<tr>
<th>CPU</th>
<th>mem</th>
<th>tmem</th>
<th>CI</th>
<th>CO</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>wmm</td>
<td>xiv</td>
<td>tIV</td>
<td>hiv</td>
</tr>
</tbody>
</table>

CI = Compressed Page In
tmem = True Memory
CO = Compressed Page Out

mem = apparent Memory

iterstat –c 1

<table>
<thead>
<tr>
<th>USER</th>
<th>TASK</th>
<th>CORE</th>
<th>S_WAIT</th>
<th>S_IDLE</th>
<th>S_PHYS</th>
<th>S_BUSY</th>
<th>V_WAIT</th>
<th>V_BUSY</th>
<th>CHARGE</th>
<th>H_WAIT</th>
<th>H_BUSY</th>
<th>H_CHARGE</th>
<th>XCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

%xcpu percentage of CPU time used in eXpansion!

Note: %user + %sys + %wait + %idle still = 100%
Active Memory Expansion

**Definitions:**
- **TMEM** = True Memory
- **CMEM** = Compressed Memory
- **CI** = Compressed Page In
- **CO** = Compressed Page Out
- **EF** = Expansion Factor
- **T** = target
- **A** = Actual

**Note:**
- `topas_nmon` not aware of AME (yet!)
- Not True Memory but the expanded size like other applications will see it.
nmon capture to file then Analyser

- **MEM tab**
  - Size of the Compressed pool (MB)
  - Size of true memory (MB)
  - Expanded memory size (MB)
  - Size of the Uncompressed pool (MB)

- **MEMNEW tab**
  - Compressed Pool%

- **PAGE tab**
  - Paging rates (pages per second) but this time very quickly in & out of the compressed memory area
  - Compressed pool pgins - other tools like topas call this CI
  - Compressed pool pgouts - other tools like topas call this CO

---

**Topas AME - Very small change**

- Units have moved
  - TMEM, MB 2000 → TMEM 2.00G
svmon

- Good luck with that one!
AME Wiki page & AME Forum

AME Public Wiki  (on the AIX wiki)
http://www.ibm.com/developerworks/wikis/display/WikiPtype/IBM+Active+Memory+Expansion
URL for trial, presentation, Forum, whitepapers, manual page, Perf Tune Guide, movies

AME Forum

Docs

1) AME Overview & Usage Guide by David Hepkin  25 pages
2) AME Performance  by Dirk Michel  18 pages

From
Then click on Whitepapers

AIX Commands Infocenter  → amepat, topas, vmstat, lparstat
AME – at the Movies

http://tinyurl.com/AIXmovies

Look for Movie 79

Summary & Questions

1. POWER7 & AIX 6 TL04 SP2+
2. Activation Key by machine
3. 60 day Trial available
4. Set at LPAR level & dynamic Expansion Factor
5. Planning: amepat
6. Monitor: vmstat -c, lparstat –c, topas

http://www.ibm.com/developerworks/wikis/display/WikiPtype/ …
IBM+Active+Memory+Expansion