Active Memory Deduplication

also called AMD or DeDup & built on Active Memory Sharing (AMS)

Presentation Version 8

Nigel Griffiths
IBM Power Systems
Advanced Technology Support, Europe

IBM Announcement
IBM PowerVM V2.2 refresh includes new function

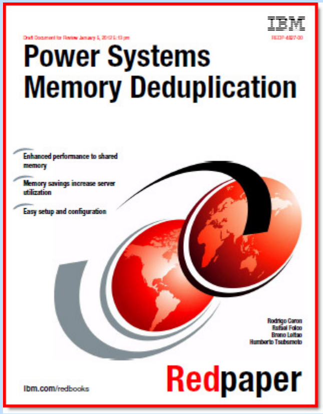
- Announcement 12th Oct 2011
- Generally available (GA) 14th Oct 2011

- Active Memory™ Deduplication detects and removes duplicate memory pages to optimize memory usage in Active Memory Sharing configurations.

http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS211-354&appname=USN

Power Systems Memory Deduplication Redbooks®

- **Redpaper** 98 pages (80 really)
- Pretty good content & easy reading
- Content
  - Concepts
  - Planning & Set-up
  - Monitoring commands
  - Tuning
  - Worked examples & Best Practice
  - Trouble shooting



Draft Document for Review January 5, 2012 5:13 pm

REDP-4827-00

Power Systems Memory Deduplication

Enhanced performance to shared memory

Memory savings increase server utilization

Easy setup and configuration

Rodrigo Caron
Rafael Folco
Bruno Leitao
Humberto Tsubamoto

ibm.com/redbooks

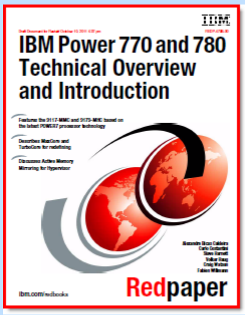
Redpaper

New Power 770/780 MMC Redbook

http://www.redbooks.ibm.com/redpieces/pdfs/redp4798.pdf

- Red Piece 4798 (not 4639)

- Section 3.4.7 “Active Memory Deduplication”
  - Good 4 page summary



Draft Document for Review October 13, 2011 4:37 pm

REDP-4798-00

IBM Power 770 and 780 Technical Overview and Introduction

Features the 9117-MMC and 9179-MHC based on the latest POWER7 processor technology

Describes MaxCore and TurboCore for redefining

Discusses Active Memory Mirroring for Hypervisor

Alexandre Bicas Caldeira
Carlo Costantini
Steve Harnett
Volker Haug
Craig Watson
Fabien Willmann

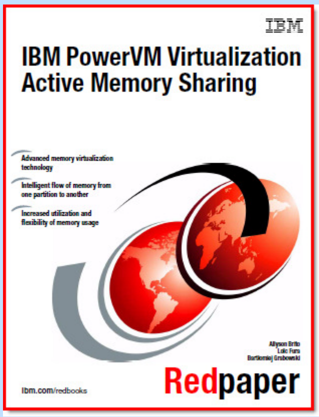
ibm.com/redbooks

Redpaper

Active Memory Sharing

Redbooks

- **Red**paper 122 pages (110 really)
- Foundation for DeDup
- Recommended reading
  - The AMD Redbook covers only the basics.



IBM

IBM PowerVM Virtualization
Active Memory Sharing

Advanced memory virtualization technology

Intelligent flow of memory from one partition to another

Increased utilization and flexibility of memory usage

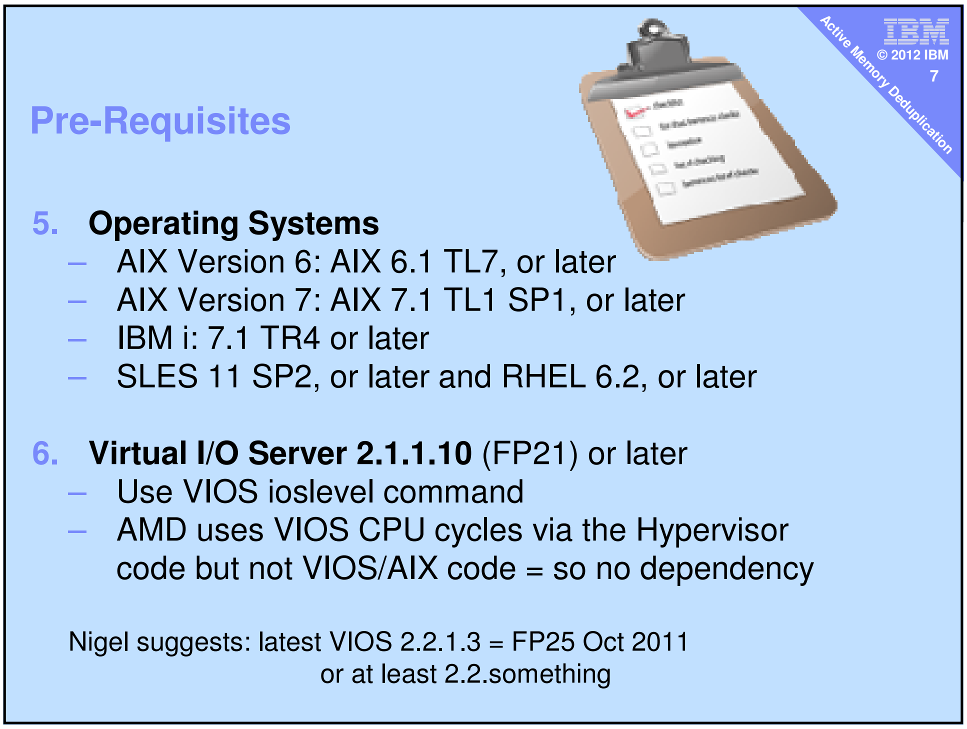
Allyson Brito
Loic Fura
Bartlomiej Grabowski

ibm.com/redbooks

Redpaper

Pre-Requisites

1. **POWER7** only
2. PowerVM **Enterprise** Edition
   - HMC → Server → Capabilities: “AMS Capable”=true
   - Suspect there is also a “Deduplication Capable” too
3. System **Firmware** level **740**
   - HMC → Update panel “EC Number”=01A*740
   - Power7xx C models introduced in Oct 2011 only
4. **HMC** level **7.7.4**
   - Matches the system firmware
   - Not possible with SDMC until at least Q2 2012

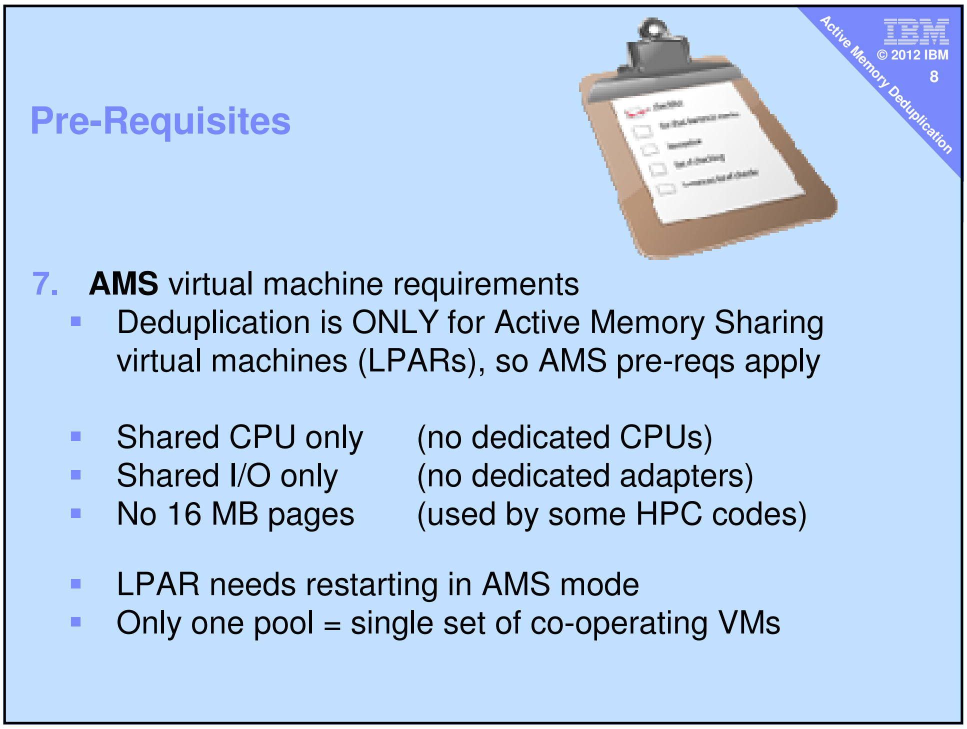


Pre-Requisites

5. **Operating Systems**
   - AIX Version 6: AIX 6.1 TL7, or later
   - AIX Version 7: AIX 7.1 TL1 SP1, or later
   - IBM i: 7.1 TR4 or later
   - SLES 11 SP2, or later and RHEL 6.2, or later

6. **Virtual I/O Server 2.1.1.10** (FP21) or later
   - Use VIOS ioslevel command
   - AMD uses VIOS CPU cycles via the Hypervisor code but not VIOS/AIX code = so no dependency

Nigel suggests: latest VIOS 2.2.1.3 = FP25 Oct 2011
or at least 2.2.something

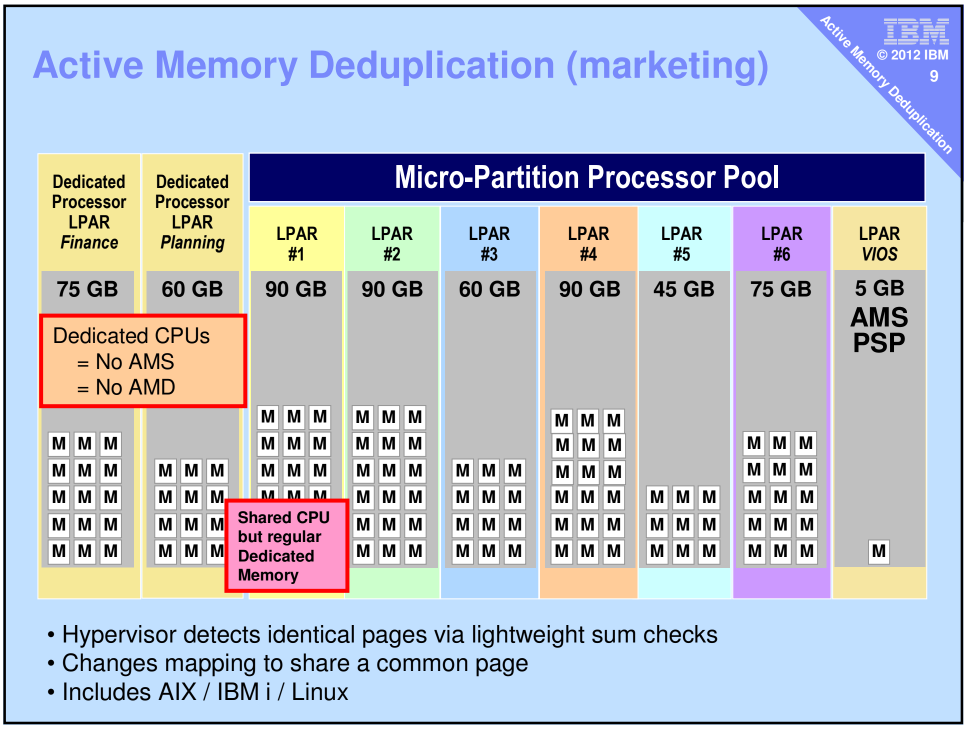


Pre-Requisites

7. **AMS** virtual machine requirements
   - Deduplication is ONLY for Active Memory Sharing virtual machines (LPARs), so AMS pre-reqs apply

   - Shared CPU only (no dedicated CPUs)
   - Shared I/O only (no dedicated adapters)
   - No 16 MB pages (used by some HPC codes)

   - LPAR needs restarting in AMS mode
   - Only one pool = single set of co-operating VMs



Active Memory Deduplication

© 2012 IBM

9

Active Memory Deduplication (marketing)

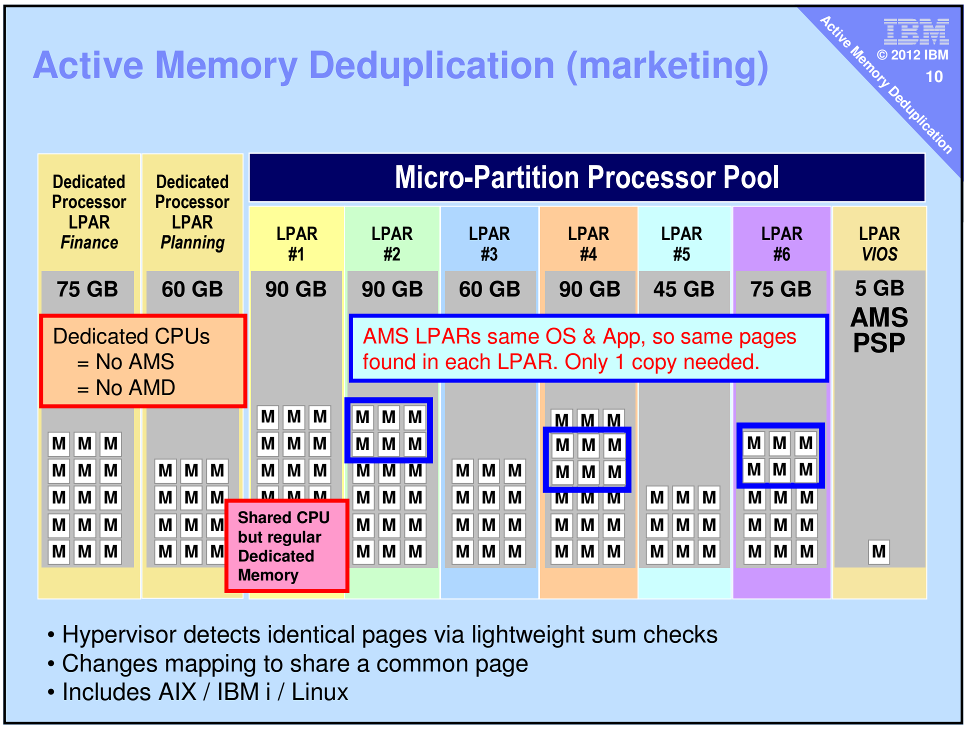
Micro-Partition Processor Pool

Dedicated Processor LPAR *Finance*	Dedicated Processor LPAR *Planning*	LPAR #1	LPAR #2	LPAR #3	LPAR #4	LPAR #5	LPAR #6	LPAR *VIOS*
75 GB	60 GB	90 GB	90 GB	60 GB	90 GB	45 GB	75 GB	5 GB AMS PSP

Dedicated CPUs
= No AMS
= No AMD

Shared CPU but regular Dedicated Memory

- Hypervisor detects identical pages via lightweight sum checks
- Changes mapping to share a common page
- Includes AIX / IBM i / Linux



Active Memory Deduplication

© 2012 IBM

10

Active Memory Deduplication (marketing)

Micro-Partition Processor Pool

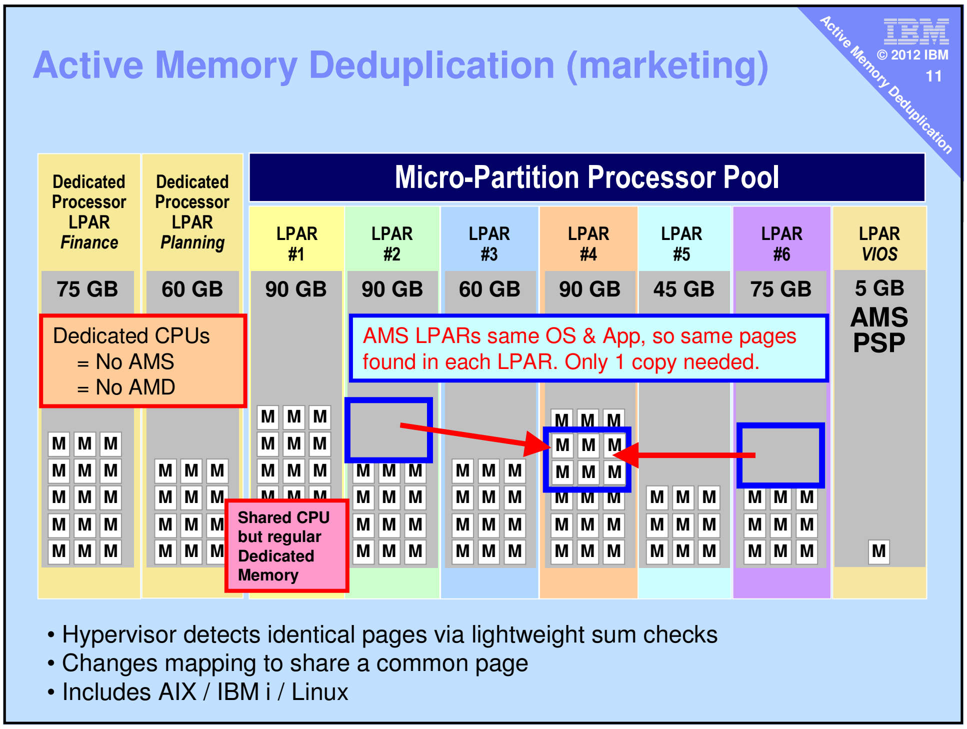
Dedicated Processor LPAR *Finance*	Dedicated Processor LPAR *Planning*	LPAR #1	LPAR #2	LPAR #3	LPAR #4	LPAR #5	LPAR #6	LPAR *VIOS*
75 GB	60 GB	90 GB	90 GB	60 GB	90 GB	45 GB	75 GB	5 GB AMS PSP

Dedicated CPUs
= No AMS
= No AMD

AMS LPARs same OS & App, so same pages found in each LPAR. Only 1 copy needed.

Shared CPU but regular Dedicated Memory

- Hypervisor detects identical pages via lightweight sum checks
- Changes mapping to share a common page
- Includes AIX / IBM i / Linux



Active Memory Deduplication (marketing)

Micro-Partition Processor Pool

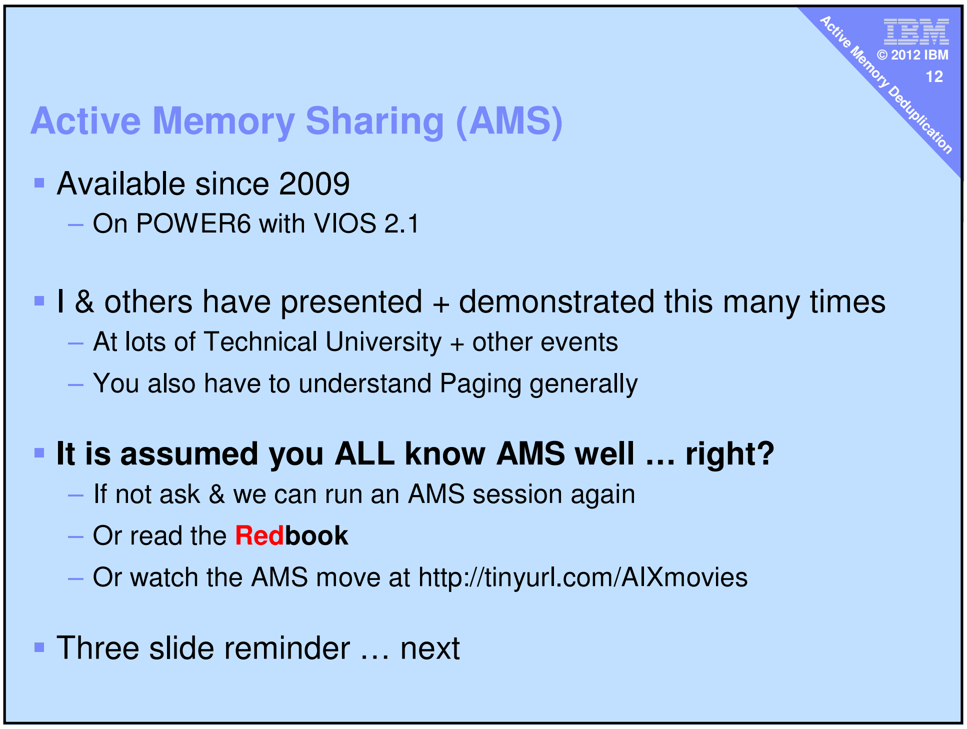
Dedicated Processor LPAR *Finance*	Dedicated Processor LPAR *Planning*	LPAR #1	LPAR #2	LPAR #3	LPAR #4	LPAR #5	LPAR #6	LPAR *VIOS*
75 GB	60 GB	90 GB	90 GB	60 GB	90 GB	45 GB	75 GB	5 GB AMS PSP

Dedicated CPUs
= No AMS
= No AMD

AMS LPARs same OS & App, so same pages found in each LPAR. Only 1 copy needed.

Shared CPU but regular Dedicated Memory

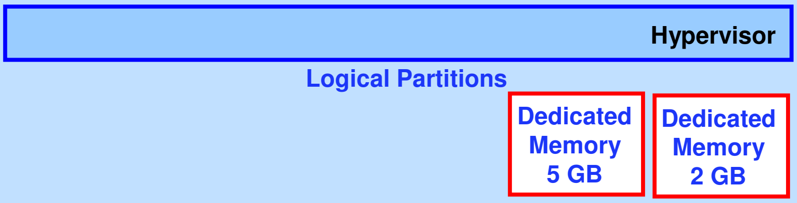
- Hypervisor detects identical pages via lightweight sum checks
- Changes mapping to share a common page
- Includes AIX / IBM i / Linux



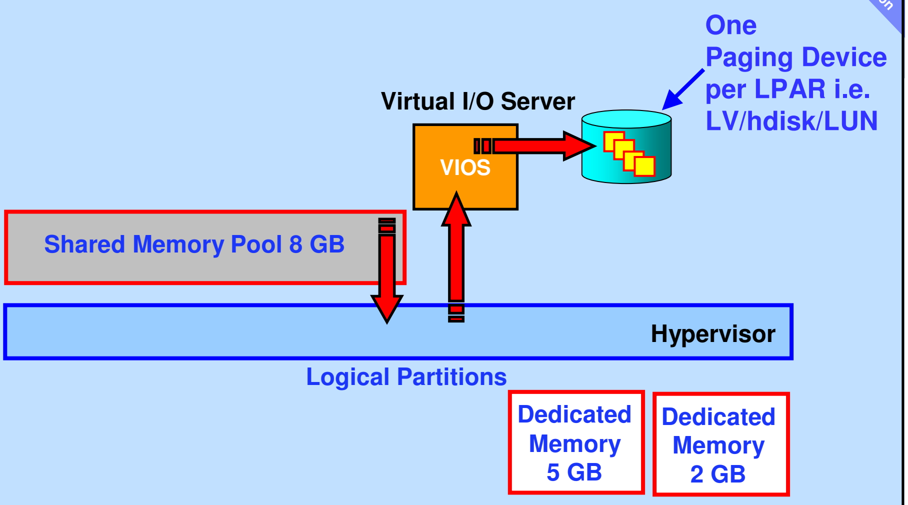
Active Memory Sharing (AMS)

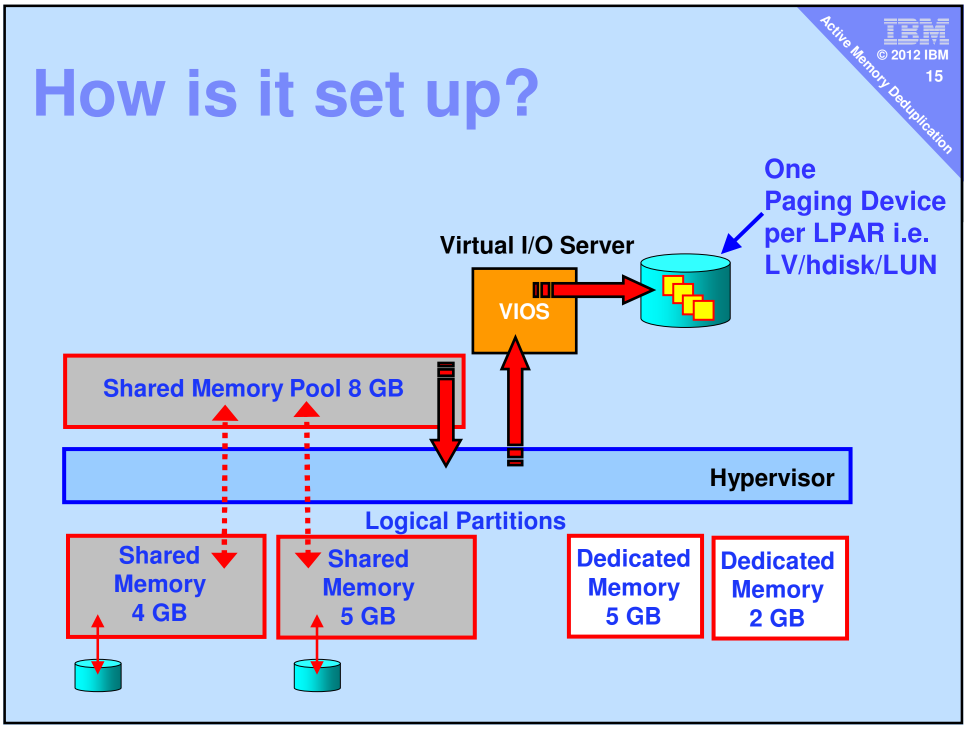
- Available since 2009
  - On POWER6 with VIOS 2.1
- I & others have presented + demonstrated this many times
  - At lots of Technical University + other events
  - You also have to understand Paging generally
- **It is assumed you ALL know AMS well … right?**
  - If not ask & we can run an AMS session again
  - Or read the **Redbook**
  - Or watch the AMS move at http://tinyurl.com/AIXmovies
- Three slide reminder … next

How is it set up?

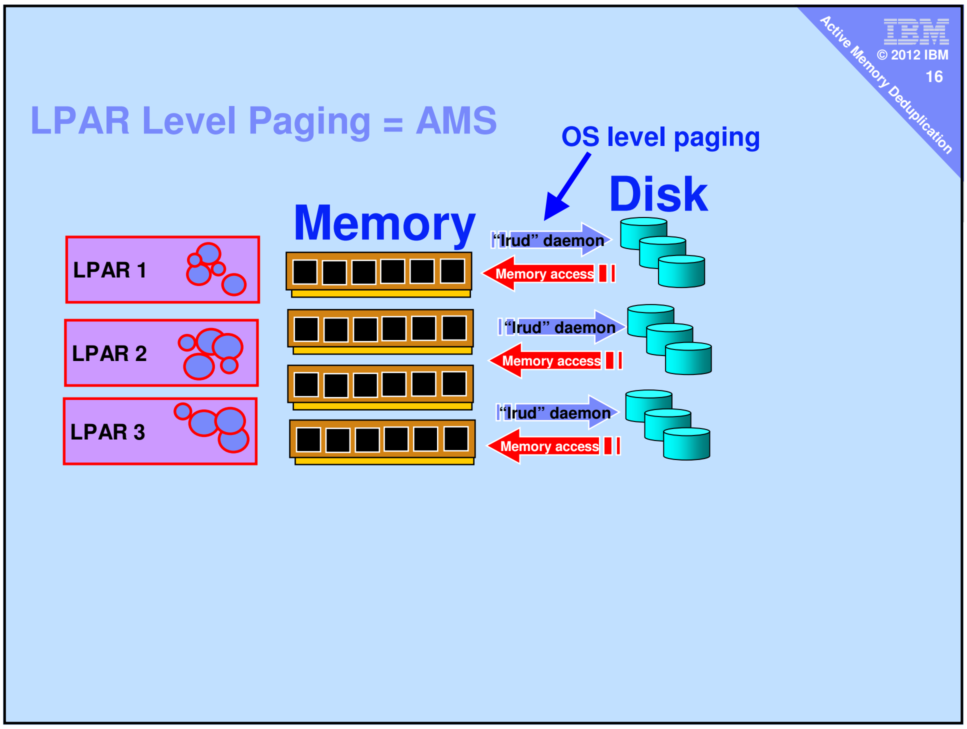


How is it set up?

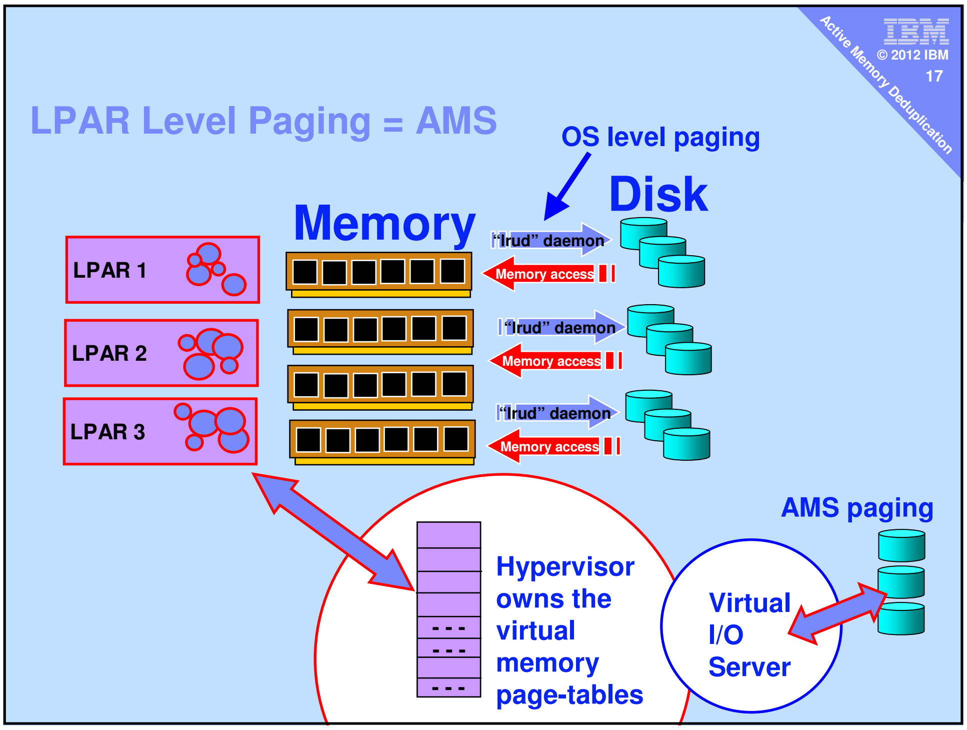




How is it set up?



LPAR Level Paging = AMS



LPAR Level Paging = AMS

OS level paging

Disk

Memory

LPAR 1

LPAR 2

LPAR 3

"lrud" daemon

Memory access

"lrud" daemon

Memory access

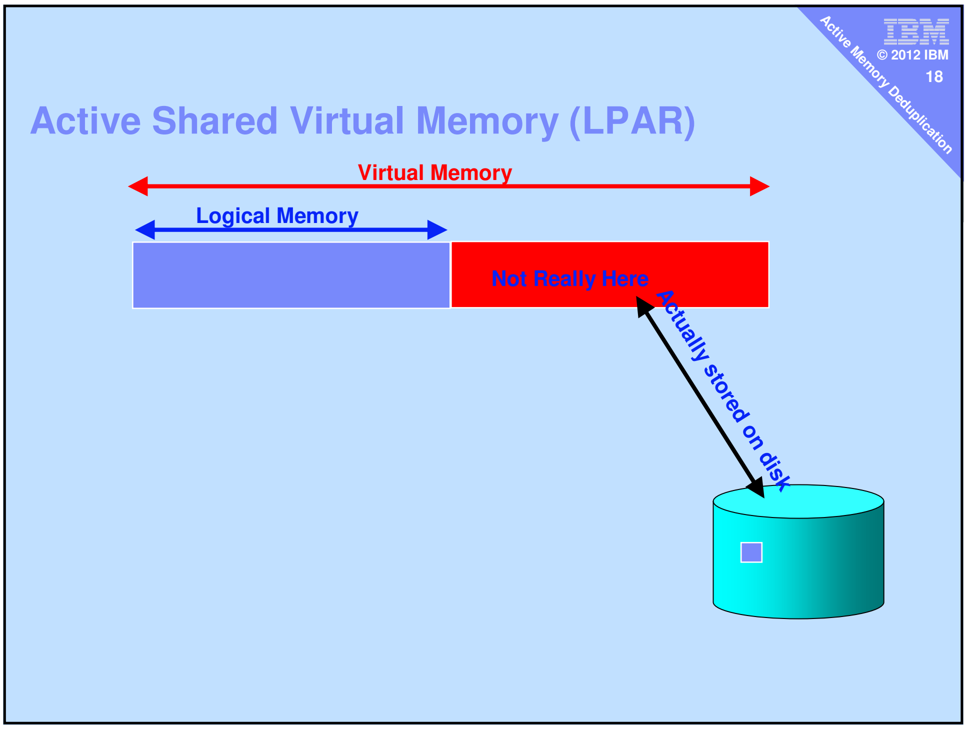
"lrud" daemon

Memory access

Hypervisor owns the virtual memory page-tables

AMS paging

Virtual I/O Server



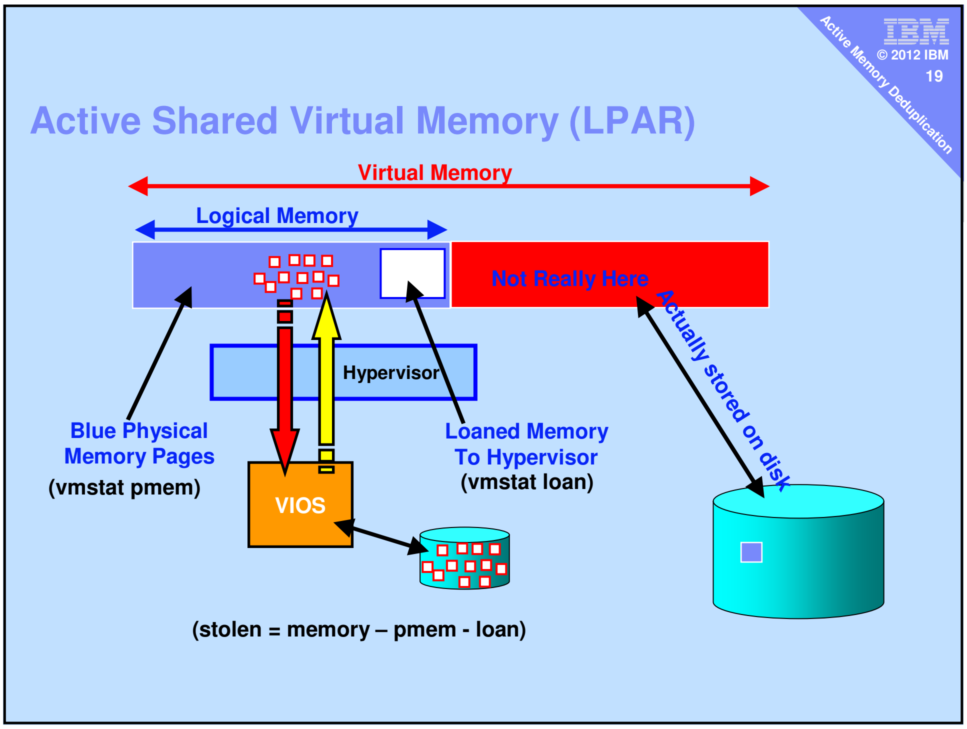
Active Shared Virtual Memory (LPAR)

Virtual Memory

Logical Memory

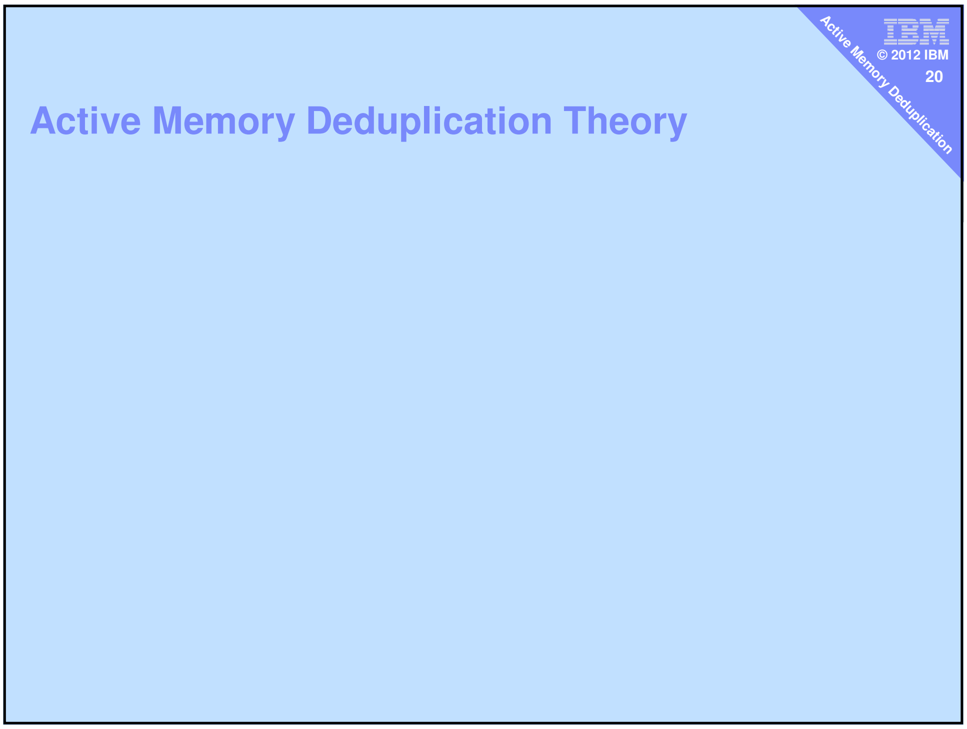
Not Really Here

Actually stored on disk

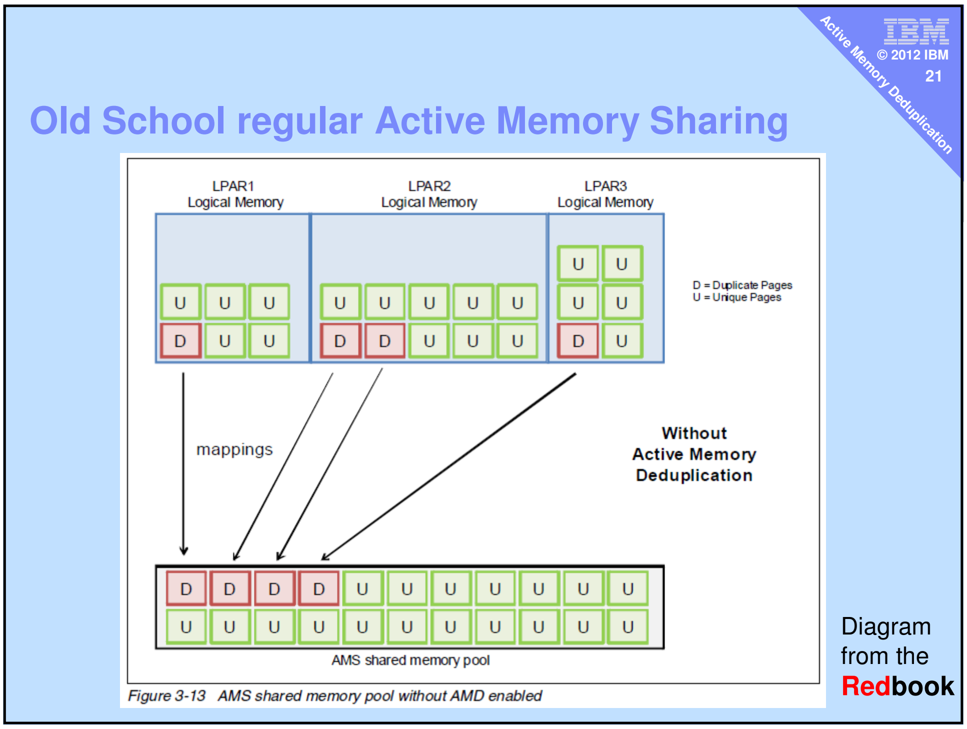


Active Shared Virtual Memory (LPAR)

(stolen = memory – pmem - loan)



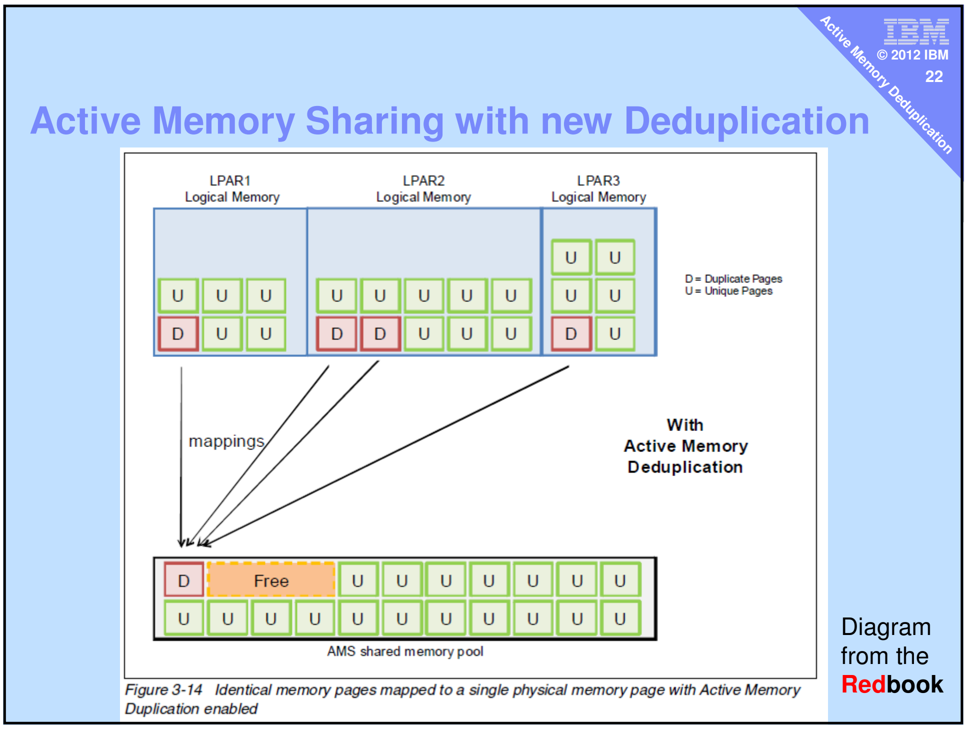
Active Memory Deduplication Theory



Old School regular Active Memory Sharing

Figure 3-13 AMS shared memory pool without AMD enabled

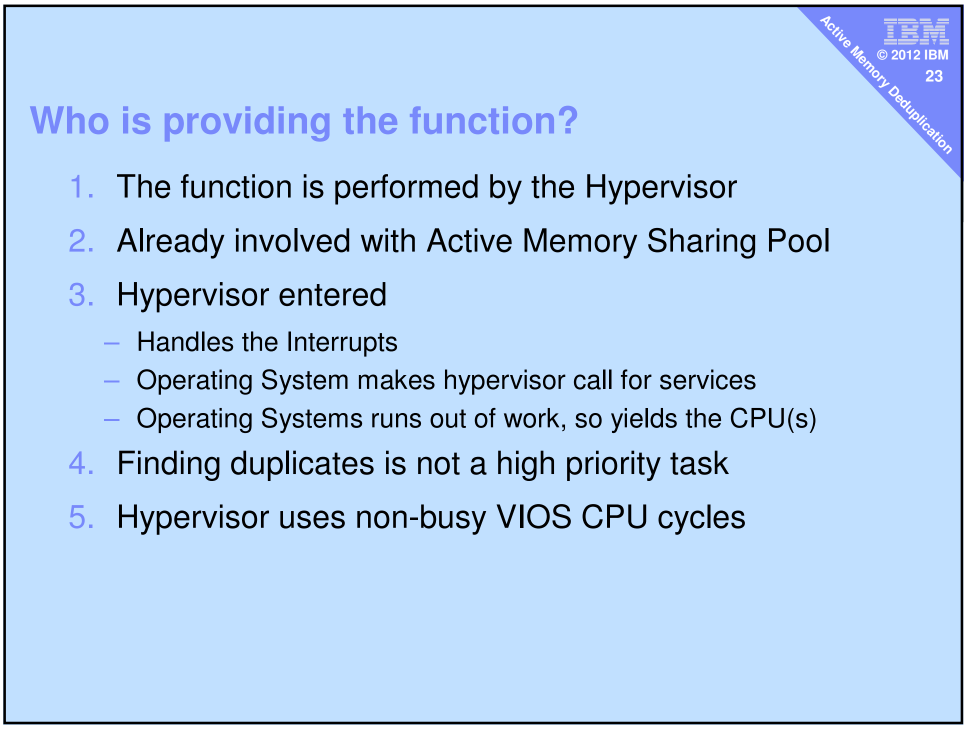
Diagram from the **Redbook**



Active Memory Sharing with new Deduplication

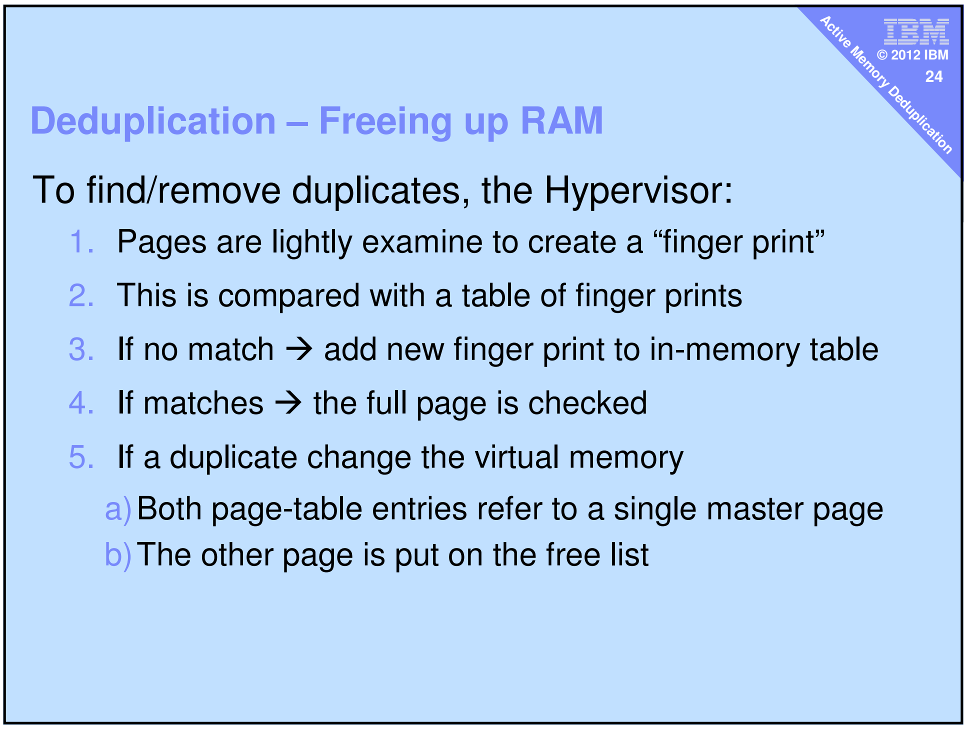
Figure 3-14 Identical memory pages mapped to a single physical memory page with Active Memory Duplication enabled

Diagram from the **Redbook**



Who is providing the function?

1. The function is performed by the Hypervisor
2. Already involved with Active Memory Sharing Pool
3. Hypervisor entered
   - Handles the Interrupts
   - Operating System makes hypervisor call for services
   - Operating Systems runs out of work, so yields the CPU(s)
4. Finding duplicates is not a high priority task
5. Hypervisor uses non-busy VIOS CPU cycles



Deduplication – Freeing up RAM

To find/remove duplicates, the Hypervisor:

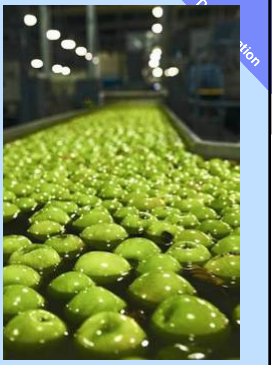
1. Pages are lightly examine to create a “finger print”
2. This is compared with a table of finger prints
3. If no match → add new finger print to in-memory table
4. If matches → the full page is checked
5. If a duplicate change the virtual memory
   a) Both page-table entries refer to a single master page
   b) The other page is put on the free list

Duplication – if pages need to be different later

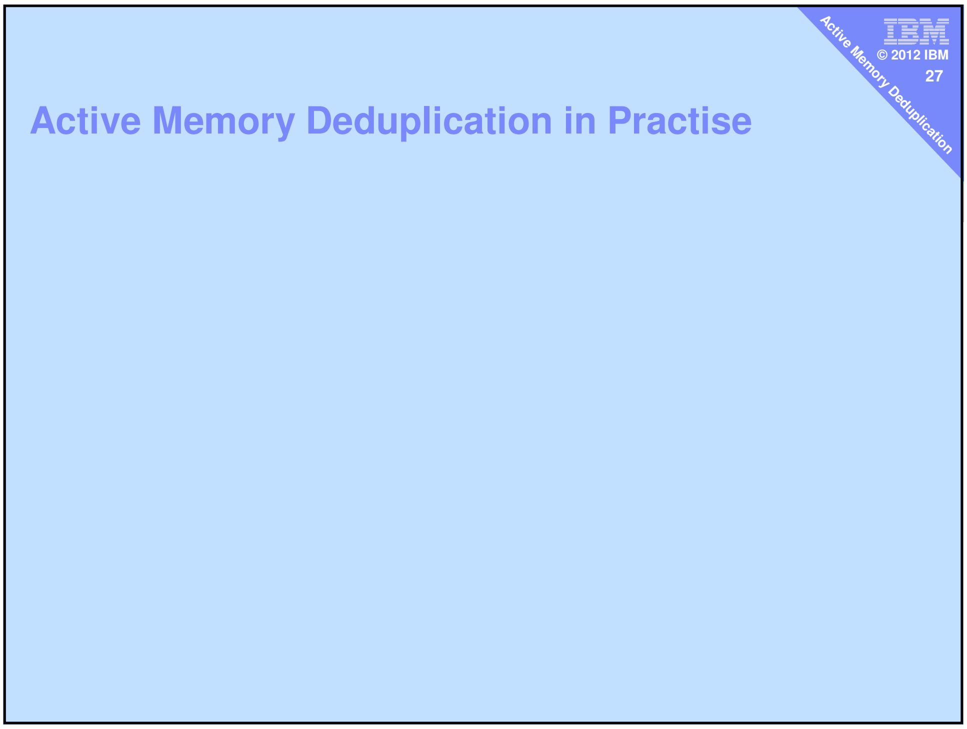
What happens on a page write attempt

1. Master pages are set to read-only
2. The page write generates a memory exception interrupt
3. If a real read-only page
   - Generate process crash signal – this is not allowed
4. If a read-write page
   a) Find a free page
   b) Make a copy of the master page to the new one
   c) Change page-table to refer to the new copy
   d) Change new copy to read-write
   e) Exit the interrupt & the process reties the write and it works

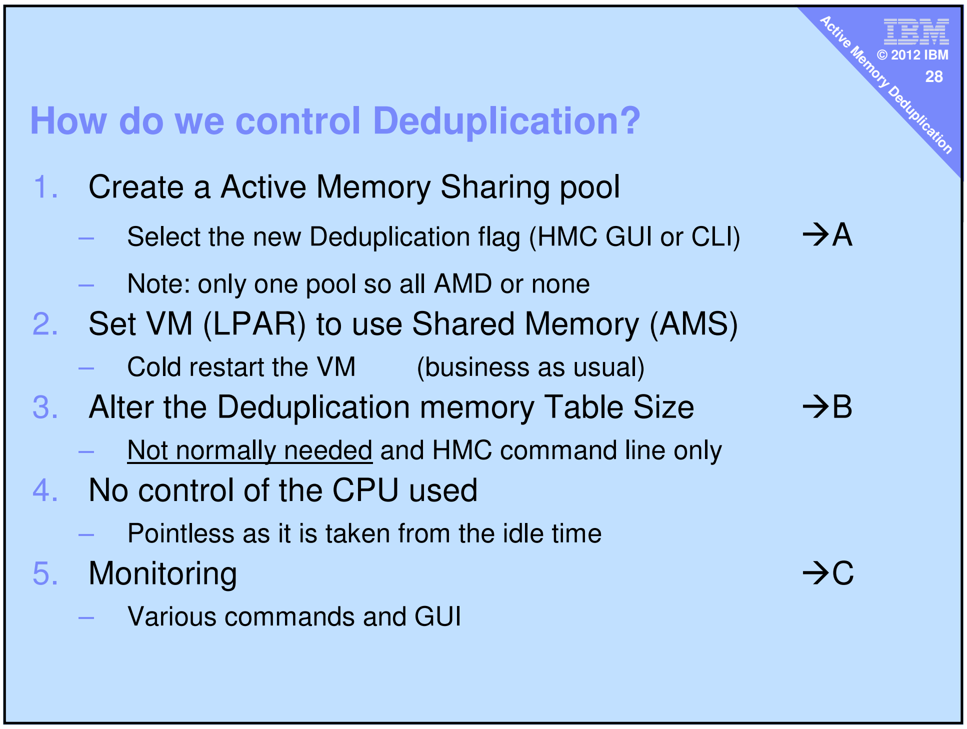
Memory page targets



- Good
  - Zero filled memory (perfect!)
    - All heap memory is zero filled to start with
  - Partly used pages (the rest is zeros)
    - Database disk blocks
  - Common read-only program code & static data
    - Operating systems code
    - Applications
  - Anything used by Java ☺
- Bad = memory pages very likely to be unique
  - Every VM running 100% different applications
  - HPC and every VM handling different data models
  - In memory images/movies editing - JPEG, GIF, TIF, MPEG
  - Encrypted data



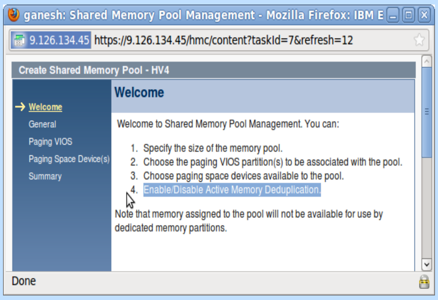
Active Memory Deduplication in Practise



How do we control Deduplication?

1. Create a Active Memory Sharing pool
   - Select the new Deduplication flag (HMC GUI or CLI) →A
   - Note: only one pool so all AMD or none
2. Set VM (LPAR) to use Shared Memory (AMS)
   - Cold restart the VM (business as usual)
3. Alter the Deduplication memory Table Size →B
   - Not normally needed and HMC command line only
4. No control of the CPU used
   - Pointless as it is taken from the idle time
5. Monitoring →C
   - Various commands and GUI

A) Active Memory Sharing Pool Setup

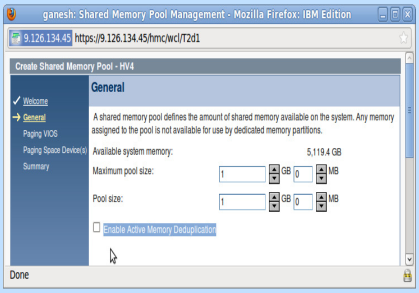

ganesh: Shared Memory Pool Management - Mozilla Firefox: IBM E
9.126.134.45 https://9.126.134.45/hmc/content?taskId=7&refresh=12
Create Shared Memory Pool - HV4
Welcome
General
Paging VIOS
Paging Space Device(s)
Summary
Done

Welcome

Welcome to Shared Memory Pool Management. You can:

1. Specify the size of the memory pool.
2. Choose the paging VIOS partition(s) to be associated with the pool.
3. Choose paging space devices available to the pool.
4. Enable/Disable Active Memory Deduplication.

Note that memory assigned to the pool will not be available for use by dedicated memory partitions.


ganesh: Shared Memory Pool Management - Mozilla Firefox: IBM Edition
9.126.134.45 https://9.126.134.45/hmc/wcl/T2d1
Create Shared Memory Pool - HV4
✓ Welcome
General
Paging VIOS
Paging Space Device(s)
Summary
Done

General

A shared memory pool defines the amount of shared memory available on the system. Any memory assigned to the pool is not available for use by dedicated memory partitions.

Available system memory: 5,119.4 GB

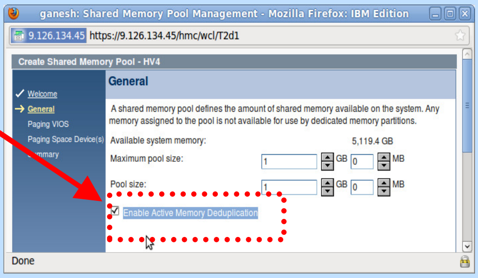
Maximum pool size: 1 GB 0 MB

Pool size: 1 GB 0 MB

□ Enable Active Memory Deduplication

Deduplication flag

“Enable Active Memory Duplication”


ganesh: Shared Memory Pool Management - Mozilla Firefox: IBM Edition
9.126.134.45 https://9.126.134.45/hmc/wcl/T2d1
Create Shared Memory Pool - HV4
✓ Welcome
General
Paging VIOS
Paging Space Device(s)
Summary
Done

General

A shared memory pool defines the amount of shared memory available on the system. Any memory assigned to the pool is not available for use by dedicated memory partitions.

Available system memory: 5,119.4 GB

Maximum pool size: 1 GB 0 MB

Pool size: 1 GB 0 MB

☑ Enable Active Memory Deduplication

Pictures
from the
Redbook

B) Overview – cached table ratio

- Deduplication Table Ratio:
  - Performance tuning parameter for the memory pool
  - To tune the memory resources consumed by AMD
  - Default is 1 in 1024 (256, 512, 1024, 2048, 4096, 8192)
    Largest……………………Smallest
- Size!
  - 1/8192 up to 1/256 of the AMS memory pool size
  - To big → small waste of RAM
  - To small → you miss spotting some duplicates
  - Neither is a large problem

B) HMC command line **For Reference Only**

- Enable & Disable AMD
  - `# chhwres -r mempool -m XXXX -o s -a "mem_dedup=1"`
  - `# chhwres -r mempool -m XXXX -o s -a "mem_dedup=0"`
- List settings

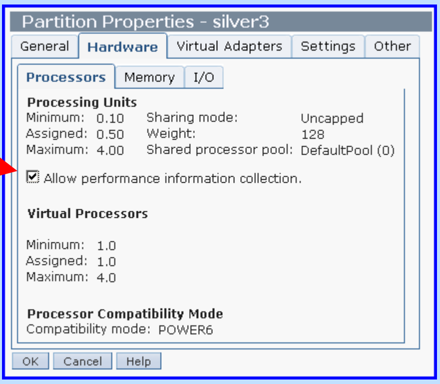
  -
    ```
    # lshwres -r mempool -m XXXX
    curr_pool_mem=512,curr_avail_pool_mem=512,curr_max_pool_mem=1024,
    pend_pool_mem=512,pend_avail_pool_mem=512,pend_max_pool_mem=1024,
    sys_firmware_pool_mem=0,paging_vios_names=vios1,paging_vios_ids=1,
    mem_dedup=1,dedup_table_ratio=1:1024
    ```
- Modify memory pool – modify deduplication table ratio **←The only way**
  - `# chhwres -r mempool -m XXXX -o s -a "dedup_table_ratio=1:512"`
- Possible deduplication table ratio values

  -
    ```
    # lshwres -r mem -m XXXX --level sys
    ……….,max_paging_vios_per_mem_pool=2,
    "default_hpt_ratios=IBM i and all shared memory partitions 1:64,all others 1:1",
    "possible_hpt_ratios=1:4,1:8,1:16,1:32,1:64,1:128,1:256,1:512,1:1024",
    default_dedup_table_ratio=1:1024,
    "possible_dedup_table_ratios=1:256,1:512,1:1024,1:2048,1:4096,1:8192"
    ```
- **Recommend using the default unless you can benchmark**
- XXXX in the machine name as seen on HMC

C) Monitoring

- AMD is largely set and forget
  Monitoring its effect is largely “nice to know”
- Two Levels:
  - VM (LPAR) level view
  - Machine level view

For Machine level stats
from within a VM (LPAR)
switch them on in the
LPAR properties
(as usual)



Partition Properties - silver3

Processing Units
Minimum: 0.10
Assigned: 0.50
Maximum: 4.00
Sharing mode: Uncapped
Weight: 128
Shared processor pool: DefaultPool (0)

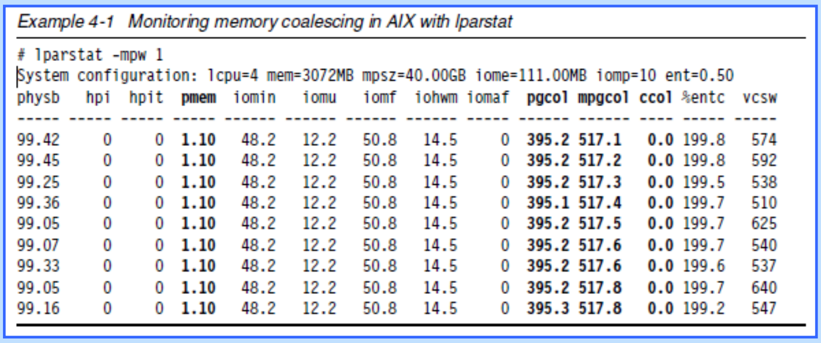
☑ Allow performance information collection.

Virtual Processors

Minimum: 1.0
Assigned: 1.0
Maximum: 4.0

Processor Compatibility Mode
Compatibility mode: POWER6

VM level for AIX: lparstat –mpw 1



Example 4-1   Monitoring memory coalescing in AIX with lparstat

```
# lparstat -mpw 1
System configuration: lcpu=4 mem=3072MB mpsz=40.00GB iome=111.00MB iomp=10 ent=0.50
physb   hpi  hpit  pmem  iomin   iomu   iomf  iohwm iomaf  pgcol mpgcol ccol %entc  vcsw
----- ----- ----- ----- ------ ------ ------ ------ ----- ------ ------ ---- ----- -----
99.42     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.1  0.0 199.8   574
99.45     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.2  0.0 199.8   592
99.25     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.3  0.0 199.5   538
99.36     0     0  1.10   48.2   12.2   50.8   14.5     0  395.1 517.4  0.0 199.7   510
99.05     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.5  0.0 199.7   625
99.07     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.6  0.0 199.7   540
99.33     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.6  0.0 199.6   537
99.05     0     0  1.10   48.2   12.2   50.8   14.5     0  395.2 517.8  0.0 199.7   640
99.16     0     0  1.10   48.2   12.2   50.8   14.5     0  395.3 517.8  0.0 199.2   547
```

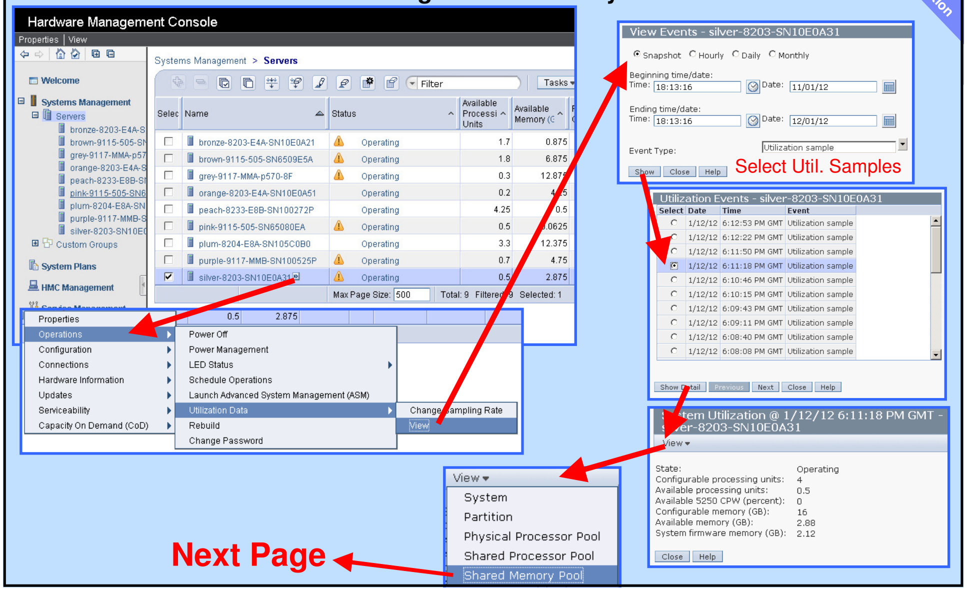
- **pgcol** VM (LPAR) coalesced memory, in MB
- **mpgcol** Whole machine coalesced memory, in MB
- **ccol** CPU used, in physical CPU-cores
- **pmem** VM physical memory, in GB (regular AMS stat)

- IBM i – no equivalent data
- Linux **amsstat** command (extra IBM package)

Picture
from the
Redbook

Machine level for HMC: Utilisation Stats

This collects lots of stats including Shared Memory Pool but new statistics



Selec	Name	Status	Available Processi Units	Available Memory (G
☐	bronze-8203-E4A-SN10E0A21	⚠ Operating	1.7	0.875
☐	brown-9115-505-SN6509E5A	⚠ Operating	1.8	6.875
☐	grey-9117-MMA-p570-8F	⚠ Operating	0.3	12.875
☐	orange-8203-E4A-SN10E0A51	Operating	0.2	4.25
☐	peach-8233-E8B-SN100272P	Operating	4.25	0.5
☐	pink-9115-505-SN65080EA	⚠ Operating	0.5	.0625
☐	plum-8204-E8A-SN105C0B0	Operating	3.3	12.375
☐	purple-9117-MMB-SN100525P	⚠ Operating	0.7	4.75
☑	silver-8203-SN10E0A31	⚠ Operating	0.5	2.875

Max Page Size: 500   Total: 9   Filtered: 9   Selected: 1

View Events - silver-8203-SN10E0A31

● Snapshot ○ Hourly ○ Daily ○ Monthly

Beginning time/date:
Time: 18:13:16 Date: 11/01/12

Ending time/date:
Time: 18:13:16 Date: 12/01/12

Event Type: Utilization sample

Select Util. Samples

Utilization Events - silver-8203-SN10E0A31

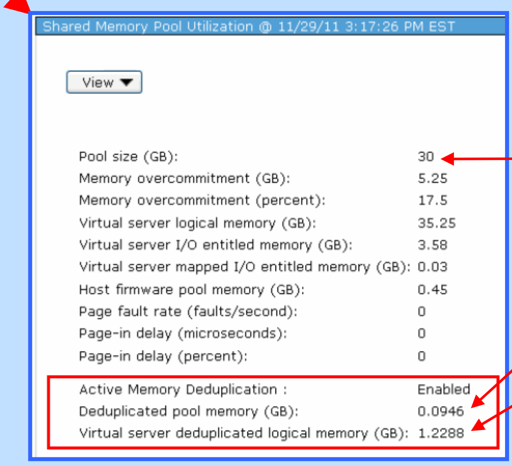
Select	Date	Time	Event
○	1/12/12	6:12:53 PM GMT	Utilization sample
○	1/12/12	6:12:22 PM GMT	Utilization sample
○	1/12/12	6:11:50 PM GMT	Utilization sample
●	1/12/12	6:11:18 PM GMT	Utilization sample
○	1/12/12	6:10:46 PM GMT	Utilization sample
○	1/12/12	6:10:15 PM GMT	Utilization sample
○	1/12/12	6:09:43 PM GMT	Utilization sample
○	1/12/12	6:09:11 PM GMT	Utilization sample
○	1/12/12	6:08:40 PM GMT	Utilization sample
○	1/12/12	6:08:08 PM GMT	Utilization sample

System Utilization @ 1/12/12 6:11:18 PM GMT - silver-8203-SN10E0A31

State: Operating
Configurable processing units: 4
Available processing units: 0.5
Available 5250 CPW (percent): 0
Configurable memory (GB): 16
Available memory (GB): 2.88
System firmware memory (GB): 2.12

Next Page

Machine level for HMC: Utilisation Stats



```
Shared Memory Pool Utilization @ 11/29/11 3:17:26 PM EST

View ▼

Pool size (GB):                                   30
Memory overcommitment (GB):                       5.25
Memory overcommitment (percent):                  17.5
Virtual server logical memory (GB):               35.25
Virtual server I/O entitled memory (GB):          3.58
Virtual server mapped I/O entitled memory (GB):   0.03
Host firmware pool memory (GB):                   0.45
Page fault rate (faults/second):                  0
Page-in delay (microseconds):                     0
Page-in delay (percent):                          0

Active Memory Deduplication :                     Enabled
Deduplicated pool memory (GB):                    0.0946
Virtual server deduplicated logical memory (GB):  1.2288
```

Pool size 30GB

Dedup RAM in the pool

Amount of RAM saved

- **Larger as the memory was duplicated many times**
- **1.2/0.09 = ~13 times**

Just a point in time or averaged over hour, day, …
Probably not worth the bother!!!

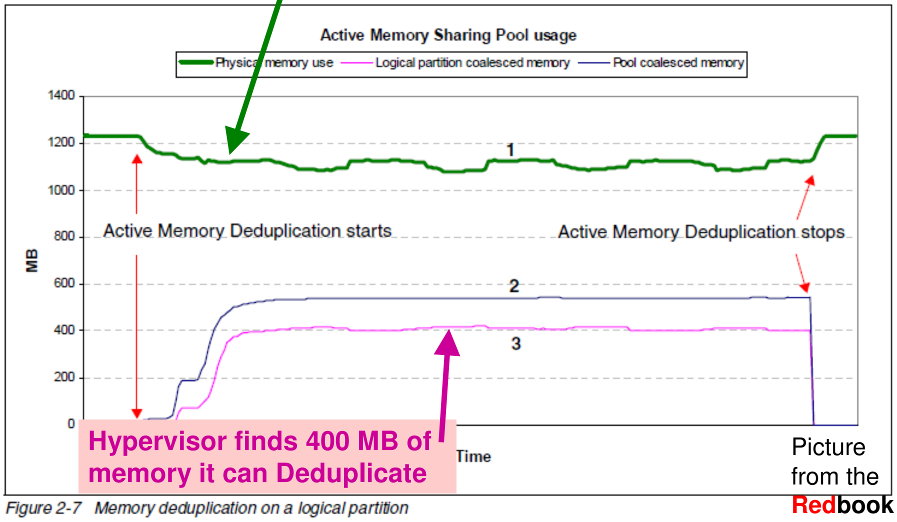
Picture
from the
Redbook

The AMD Redbook includes benchmarks

- Of various workloads with graphs

AMD Example while under load

VM Memory Use
3 GB Max Logical RAM from pool
1.25 GB in actual use and dips ~ 10%



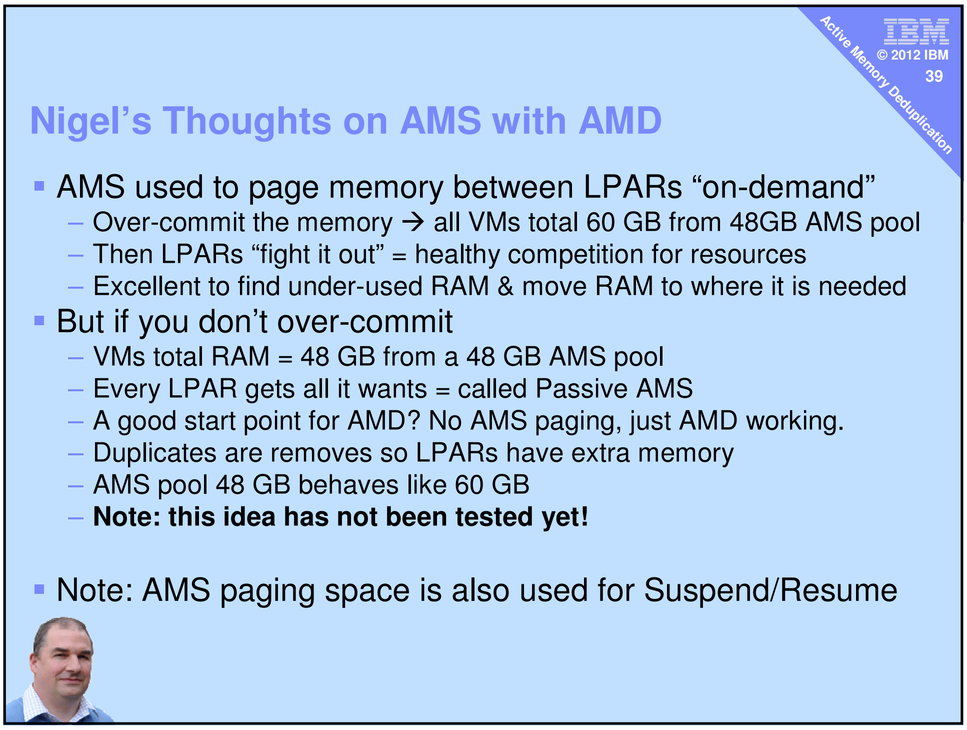
Hypervisor finds 400 MB of memory it can Deduplicate

Picture from the **Red**book

Figure 2-7 Memory deduplication on a logical partition

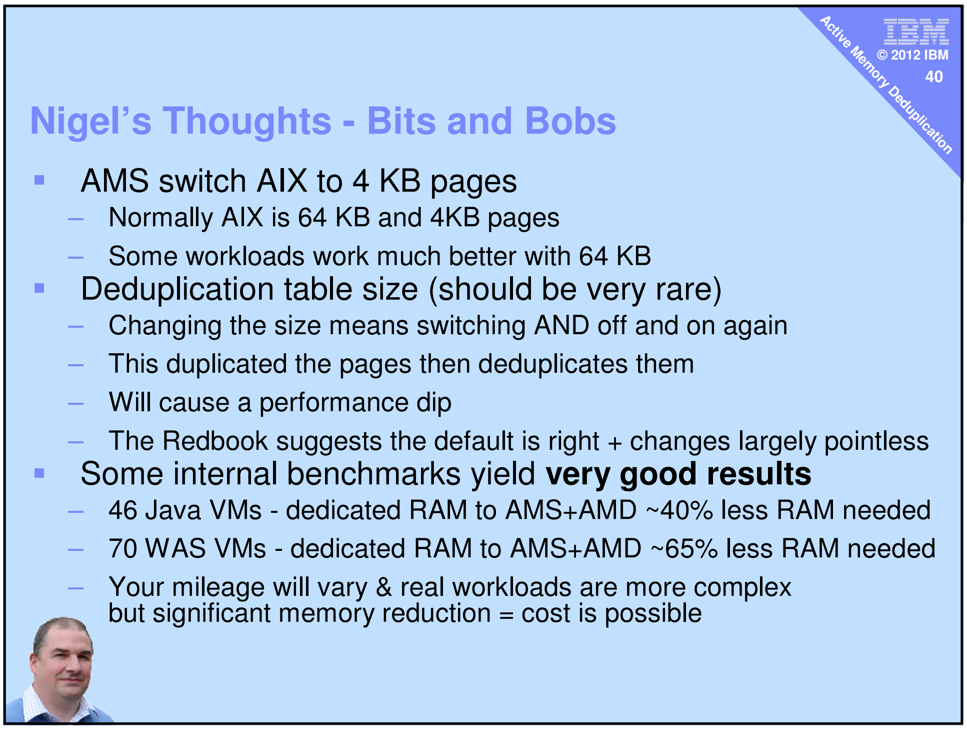
Summary: Active Memory Deduplication

- Largest pre-req in the Oct 2011 C models firmware
- Good **Red**book(s) & simple to understand
- Very simple to implement (once AMS set-up)
- Very low CPU impact (VIOS idle time)
- High gains in memory use
- Set & forget
- On AIX monitor with: lparstat –mpw



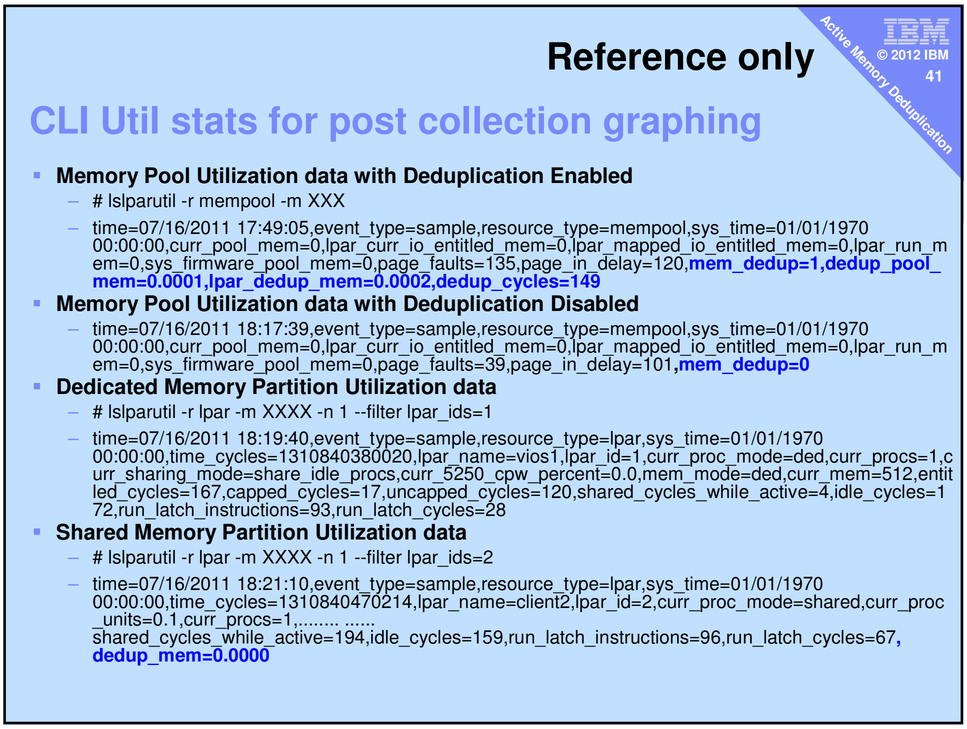
Nigel’s Thoughts on AMS with AMD

- AMS used to page memory between LPARs “on-demand”
  - Over-commit the memory → all VMs total 60 GB from 48GB AMS pool
  - Then LPARs “fight it out” = healthy competition for resources
  - Excellent to find under-used RAM & move RAM to where it is needed
- But if you don’t over-commit
  - VMs total RAM = 48 GB from a 48 GB AMS pool
  - Every LPAR gets all it wants = called Passive AMS
  - A good start point for AMD? No AMS paging, just AMD working.
  - Duplicates are removes so LPARs have extra memory
  - AMS pool 48 GB behaves like 60 GB
  - **Note: this idea has not been tested yet!**
- Note: AMS paging space is also used for Suspend/Resume



Nigel’s Thoughts - Bits and Bobs

- AMS switch AIX to 4 KB pages
  - Normally AIX is 64 KB and 4KB pages
  - Some workloads work much better with 64 KB
- Deduplication table size (should be very rare)
  - Changing the size means switching AND off and on again
  - This duplicated the pages then deduplicates them
  - Will cause a performance dip
  - The Redbook suggests the default is right + changes largely pointless
- Some internal benchmarks yield **very good results**
  - 46 Java VMs - dedicated RAM to AMS+AMD ~40% less RAM needed
  - 70 WAS VMs - dedicated RAM to AMS+AMD ~65% less RAM needed
  - Your mileage will vary & real workloads are more complex
    but significant memory reduction = cost is possible



Reference only

CLI Util stats for post collection graphing

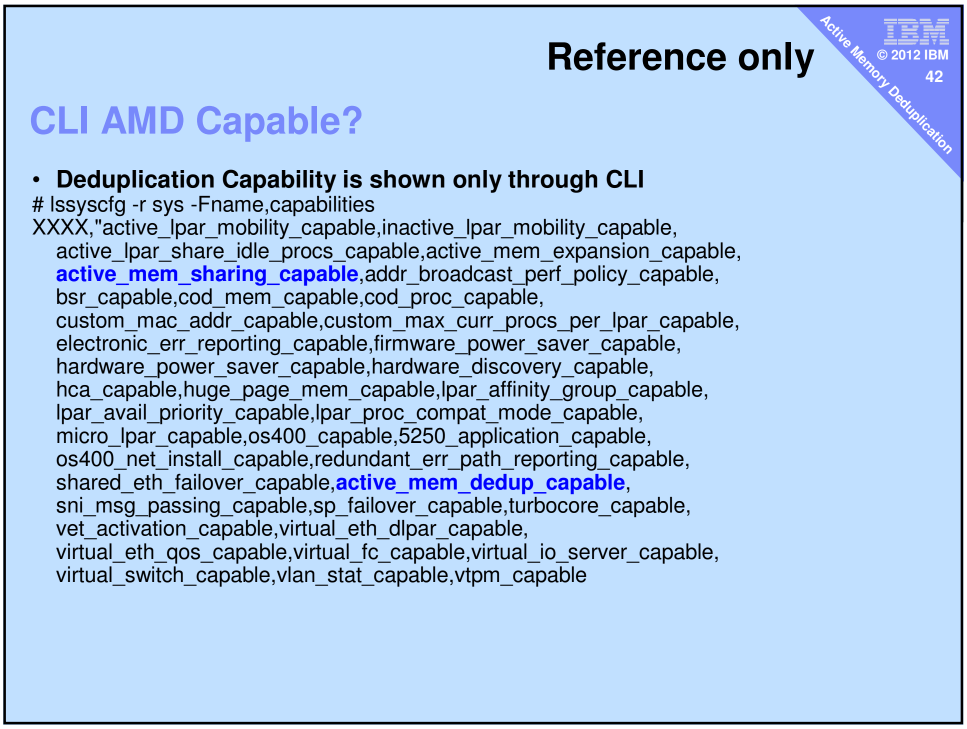
- **Memory Pool Utilization data with Deduplication Enabled**
  - `# lslparutil -r mempool -m XXX`

  -
    ```
    time=07/16/2011 17:49:05,event_type=sample,resource_type=mempool,sys_time=01/01/1970
    00:00:00,curr_pool_mem=0,lpar_curr_io_entitled_mem=0,lpar_mapped_io_entitled_mem=0,lpar_run_m
    em=0,sys_firmware_pool_mem=0,page_faults=135,page_in_delay=120,mem_dedup=1,dedup_pool_
    mem=0.0001,lpar_dedup_mem=0.0002,dedup_cycles=149
    ```
- **Memory Pool Utilization data with Deduplication Disabled**

  -
    ```
    time=07/16/2011 18:17:39,event_type=sample,resource_type=mempool,sys_time=01/01/1970
    00:00:00,curr_pool_mem=0,lpar_curr_io_entitled_mem=0,lpar_mapped_io_entitled_mem=0,lpar_run_m
    em=0,sys_firmware_pool_mem=0,page_faults=39,page_in_delay=101,mem_dedup=0
    ```
- **Dedicated Memory Partition Utilization data**
  - `# lslparutil -r lpar -m XXXX -n 1 --filter lpar_ids=1`

  -
    ```
    time=07/16/2011 18:19:40,event_type=sample,resource_type=lpar,sys_time=01/01/1970
    00:00:00,time_cycles=1310840380020,lpar_name=vios1,lpar_id=1,curr_proc_mode=ded,curr_procs=1,c
    urr_sharing_mode=share_idle_procs,curr_5250_cpw_percent=0.0,mem_mode=ded,curr_mem=512,entit
    led_cycles=167,capped_cycles=17,uncapped_cycles=120,shared_cycles_while_active=4,idle_cycles=1
    72,run_latch_instructions=93,run_latch_cycles=28
    ```
- **Shared Memory Partition Utilization data**
  - `# lslparutil -r lpar -m XXXX -n 1 --filter lpar_ids=2`

  -
    ```
    time=07/16/2011 18:21:10,event_type=sample,resource_type=lpar,sys_time=01/01/1970
    00:00:00,time_cycles=1310840470214,lpar_name=client2,lpar_id=2,curr_proc_mode=shared,curr_proc
    _units=0.1,curr_procs=1,........ ......
    shared_cycles_while_active=194,idle_cycles=159,run_latch_instructions=96,run_latch_cycles=67,
    dedup_mem=0.0000
    ```



Reference only

CLI AMD Capable?

- **Deduplication Capability is shown only through CLI**

```
# lssyscfg -r sys -Fname,capabilities
XXXX,"active_lpar_mobility_capable,inactive_lpar_mobility_capable,
  active_lpar_share_idle_procs_capable,active_mem_expansion_capable,
  active_mem_sharing_capable,addr_broadcast_perf_policy_capable,
  bsr_capable,cod_mem_capable,cod_proc_capable,
  custom_mac_addr_capable,custom_max_curr_procs_per_lpar_capable,
  electronic_err_reporting_capable,firmware_power_saver_capable,
  hardware_power_saver_capable,hardware_discovery_capable,
  hca_capable,huge_page_mem_capable,lpar_affinity_group_capable,
  lpar_avail_priority_capable,lpar_proc_compat_mode_capable,
  micro_lpar_capable,os400_capable,5250_application_capable,
  os400_net_install_capable,redundant_err_path_reporting_capable,
  shared_eth_failover_capable,active_mem_dedup_capable,
  sni_msg_passing_capable,sp_failover_capable,turbocore_capable,
  vet_activation_capable,virtual_eth_dlpar_capable,
  virtual_eth_qos_capable,virtual_fc_capable,virtual_io_server_capable,
  virtual_switch_capable,vlan_stat_capable,vtpm_capable
```