



VELOCITY  
SOFTWARE

*z/VM Storage, z/VM 6.3+*

Velocity Software Inc.  
196-D Castro Street  
Mountain View CA 94041  
650-964-8867

Velocity Software GmbH  
Max-Joseph-Str. 5  
D-68167 Mannheim  
Germany  
+49 (0)621 373844

## Objectives:

- Understand Storage Requirements
- Know how/where to measure
- Understand Demand Paging
- Determine Requirements
- Understand Measurements
- Suggest tuning measures
- Understand queue drop requirements (gone in 6.3)

## CP Indicate used to show storage utilization – what does it mean?

- Storage requirements of InQueue users / main storage
- Lots of users with short queue stays have low storage utilization (<30%)
  - High active to inqueue ratio
  - This causes significant paging
- Few users with long queue stays have high storage utilization (>80%)
  - This would have very little paging
  - Low active to inqueue ratio

## What does STORBUF 300 really mean?

## What happens to storage utilization when lots of short queue stays?

When I/O response goes up?

When paging response time goes up?

If the measure of storage requirements goes up, but real requirements did not change, is it a valid measure?

# Storage Load – traditional z/VM workloads

What is the problem to solve?

Project storage requirements so that paging delay is manageable

What causes paging?

On healthy system CMS users become active and thus need storage

On un-healthy systems, users lose their storage between queue stays

What would be the measure of the storage requirement that causes paging?

The measure of the storage required by the active CMS users

Storage Load =  $\text{SUM}(\text{ACTIVE pages}) / (\text{DPA} + \text{ExStore})$

# Storage Overcommit – Linux Workloads

## What is the problem to solve?

Project storage requirements so that paging delay is manageable

## Define storage overcommit

Size of virtual machines logged on / Size of real storage

## Ranges of overcommit?

.9 for no paging, online/real time production

2-3 for development

## Is it a valid or relevant number?

Arbitrarily create extra large servers, overcommit goes up

Tune servers, overcommit goes down

It's only purpose is to gauge paging requirements

## CP Fixed Storage

- Nucleus
- Frame table (32 byte frame table entry per real page)

## CP Non Pageable: Storage made non–pageable by CP

- Trace table (size can be modified, default 100)
- Free storage (control blocks, erep)
  - (most is now maintained in SXS – to make 64 bit addressable)
- Prefix pages (2 per processor / IFL)
- SNTBKs (one per dcss/nss)
- SAVBKs (CP save areas, 32 per page)
- VMDBKs (2 pages per guest virtual processor)
- Segment Tables
- QDIO Buffers
- DIAG98 Buffers

The remaining pages are Dynamic Paging Area(DPA)

# Storage Map – Dynamic paging area (DPA)

## System Execution Space

- CP Free storage, implemented for 64-bit addressing
- Accounting records

## User storage

## Address Space

- Shared address spaces (system and user)

## Vdisk

- Linux swap devices

## DCSS / NSS (cms workloads)

## PGMBKs (page tables - PTRM00xx address spaces)

## MDC

## Available List (greater/less than 2gb)

# Storage Map, ESASTR1

## Storage Map to determine where storage is going

- 104M Pages = 400GB
- In 6.3, available list “small”
- User resident should be major use

## Validate use of storage

Capture ratio shows very good information

Further drill down into ALL areas possible

Overcommit means paging

Note zmap parameter: uspg\_byMB = '0'b;

Report: ESASTR1      Main Storage Analysis      ZVMSYS Z/VM6.3      ZMAP 4.2.4 06/01/16      Pg 64  
Monitor initialized: 05/31/16 at 00:00:00 on 2827 serial xxxxxx      First record analyzed: 05/31/16 00:00:00

Time	Users <-----		Pages-----						----->						Over		
	Loggd On	System Storage	Fixed Store	Non-Pgble	Free Stor	Frame Table	<Available> <2gb >2gb	System ExSpc	User Resdnt	NSS/DCSS Resident	<-AddSpace> System User	VDISK Rsdnt	<MDC> Rsdnt	Diag 98		Commit Ratio	Capt-Ratio
00:15:00	73	104858K	2828	42154	3902	819K	15 233	20163	103M	4741	1338K	0	0	0	4240	2.528	0.999
00:30:00	73	104858K	2828	42147	3902	819K	17 192	20166	103M	4541	1335K	0	0	0	4240	2.528	0.999
00:45:00	73	104858K	2828	42147	3902	819K	17 286	20169	103M	3905	1338K	0	0	2	4240	2.528	0.999
01:00:00	73	104858K	2828	42148	3902	819K	15 340	20166	103M	4187	1339K	0	0	0	4240	2.528	0.999





# Storage Map – Fixed, CP, ESASTR1

## Fixed Storage (no tuning)

- Nucleus

## Frame Table (fixed, no tuning)

## Non-Pageable

- Trace table (default small)

## Free Storage (no tuning)

- CP Control blocks

## System Execution Space

- Accounting disabled?

Report: ESASTR1      Main Storage Analysis      ZVMSYS Z/VM6.3      ZMAP 4.2.4 06/01/16      Pg 64  
Monitor initialized: 05/31/16 at 00:00:00 on 2827 serial xxxxx      First record analyzed: 05/31/16 00:00:00

Time	Users <----->		Pages		-----> Over													
	Loggd	System	Fixed	Non-	Free	Frame	<Available>	System	User	NSS/DCSS	<-AddSpace>	VDISK	<MDC>	Diag	Commit	Capt-		
	On	Storage	Store	Pgble	Stor	Table	<2gb	>2gb	ExSpC	Resdnt	Resident	System	User	Rsdnt	Rsdnt	98	Ratio	Ratio
00:15:00	73	104858K	2828	42154	3902	819K	15	233	20163	103M	4741	1338K	0	0	0	4240	2.528	0.999
00:30:00	73	104858K	2828	42147	3902	819K	17	192	20166	103M	4541	1335K	0	0	0	4240	2.528	0.999
00:45:00	73	104858K	2828	42147	3902	819K	17	286	20169	103M	3905	1338K	0	0	2	4240	2.528	0.999
01:00:00	73	104858K	2828	42148	3902	819K	15	340	20166	103M	4187	1339K	0	0	0	4240	2.528	0.999

# CP Storage Management – Frame Table

The Frame Table (CP Fixed Storage) is the usage map of REAL (main) storage.

- One entry (32 bytes) for each page (2 page per MB Real)
- All entries chained on specific list for (current) type of page
- If 196 GB of storage, 1,536 MB free storage

Resident pages for a user are dynamic Frame Table entries chained together in a List: the User Frame Owned List (UFOL).

CP address spaces and Shared Segment (NSS/DCSS) pages are similarly chained. (SUFO) (not stolen so much)

z/VM 6.3 adds VUFO (virtual disk frame owned list)

When storage is needed by ANY process, it is acquired from the Available List, another chain of Frame Table entries.

# CP Storage Management - Trace Table

Trace table for Master processor and each other real CPU

Trace Table size: Master = Min(100 pages)

Trace table size = master + (NCPU-1) \* .75 \* Master

“CP SET TRACEFRAMES MASTER 100 ALTERNATE 50 PERCENT”

- Must be greater than 2
- Storage is “below the line”
- Size displayed on ESASTRC

CP SET TRACEFRAMES MASTER 2000

Screen: ESASTR1 Velocity Software

1 of 2 Main Storage Analysis

<-----Pages----->

Time	System Storage	Fixed Store	Non-Pgble	Free Stor	Frame Table	<Avai <2gb
13:56:00	1310720	2244	<b>5088</b>	4	10240	<b>468K</b>
13:55:00	1310720	2244	3188	4	10240	470K
13:54:00	1310720	2244	3197	4	10240	470K

# CP Storage Management - Free Storage

Free storage (now in SXS) used for recording for:

Erep, Accounting, Symptom

## Issue CP QUERY RECORDING

- **Very common configuration error**

To stop recording, free up storage:

- CP RECORDING ACCOUNT OFF PURGE
- Or disable it in SYSTEM CONFIG

### q recording

RECORDING		COUNT	LMT	USERID	COMMUNICATION
EREP	ON	00000088	002	EREP	INACTIVE
ACCOUNT	ON	<b>00044232</b>	020	DISKACNT	INACTIVE
SYMPTOM	ON	00000000	002	OPERSYMP	ACTIVE

# Free Storage Recording

To stop recording, free up storage:

- CP RECORDING ACCOUNT OFF PURGE
- Or disable it in SYSTEM CONFIG.

**cp recording account off purge**

HCPCRC8058I User BARTON has purged 00044234 records from the \*ACCOUNT queue.

Command complete

**Screen: ESASTR1 Velocity Software**

2 of 2 Main Storage Analysis

<-----Pages----->

Time	System ExSpc	User Resdnt	NSS/DCSS Resident	<-AddSpace> System User	
16:05:00	1059	28184	9210	3386	0
16:04:00	<b>1060</b>	24539	9215	3386	0
16:03:00	<b>2279</b>	27702	9207	3386	0
16:02:00	2274	25189	9209	3374	0
16:01:00	2259	23617	9209	3374	0

# Free Storage Recording

## Free Storage analysis – zmon esastr1 (split screen)

Screen: **ESASTR1** Velocity Software ESAMON 4.240  
2 of 2 Main Storage Analysis

Time	System ExSpc	User Resdnt	NSS/DCSS Resident	Pages <-AddSpace> System User	VDISK Rsdnt	<MDC> Rsdnt	Diag 98
16:05:00	1059	28184	9210	3386	0	0 64966	1037
16:04:00	<b>1060</b>	24539	9215	3386	0	0 64919	1037
16:03:00	2279	27702	9207	3386	0	0 64905	1037
16:02:00	2274	25189	9209	3374	0	0 65407	1037
16:01:00	2259	23617	9209	3374	0	0 65407	1037

====>

Screen: **ESASTR1** Velocity Software ESAMON 4.240  
1 of 2 Main Storage Analysis

Time	System Storage	Fixed Store	Non-Pgble	Free Stor	Frame Table	<Available > <2gb >2gb	Capture Ratio
16:05:00	1310720	2244	3143	4	10240	473K 694793	0.995
16:04:00	1310720	2244	3143	4	10240	473K <b>694848</b>	0.995
16:03:00	1310720	2244	3146	4	10240	473K <b>693692</b>	0.995
16:02:00	1310720	2244	3167	4	10240	473K 693716	0.995
16:01:00	1310720	2244	3166	4	10240	473K 693714	0.995



## CP Allocations

- ESASTR1/2 (Storage Analysis)

## Determine Pageable Storage Requirements

- ESAUSR2 (User Addressable Storage)
- ESAASPC (Address Space – page tables, vdisks, system)
- ESAMDC (Minidisk Cache)
- ESAVDSK (Virtual Disk)
- ESADCSS (Saved Segments – CMS, monitor)
- ESAUSPG (User paging analysis)

## Other

- ESASUM (Demand Scan (Prior to z/VM 6.3))
- ESAUSR3 – Pages resident at reset (Prior to z/VM 6.3)

# User Storage - Traditional

## User Storage analysis – ESAUSP2 / ESAUSR2

- Resident vs WSS (working set)
- Active vs total (active cms users vs total users)

```
Report: ESAUSP2          User Resource Rate Report
-----
      <---CPU time--> <---Main Storage (pages)----->
UserID  <(Percent)> T:V <Resident> Lock <-----WSS----->
/Class  Total  Virt  Rat  Totl  Activ  -ed  Totl  Activ  Avg
-----  -----  ---  ----  -----  -----  ----  -----  -----  ----
00:15:00  1346  1323  1.0  103M  103M  112  123M  123M  2M
  ***Key User Analysis ***
TCPIP      0.20  0.10  2.0  1355  1355    8   588  588.3  588
  ***User Class Analysis***
Servers    0.01  0.01  1.1   447  302.1    4   438  296.9   33
Velocity   5.41  5.18  1.0  6638  5318    0  6116  4796  461
LDXmno    130.0 126.8  1.0   12M  11.9M    7   14M  14.3M   4M
LDXecs    383.8 380.5  1.0   36M  36.0M    7   43M  43.3M  11M
LDXsph    135.8 134.5  1.0   17M  16.5M    7   20M  19.8M   5M
TheUsrs   68.59 67.10  1.0   1.4M  1360K   27   1.8M  1813K  139K
  ***Top User Analysis***
LDX304    246.4 245.9  1.0   4.9M  4892K    2   5.9M  5883K   6M
LDX131    125.1 124.8  1.0   943K   943K    1   915K   915K   1M
```



## User Storage analysis – ESAUSPG

- Above / Below 2GB line, xstore, dasd paging
- Xstore vs DASD, page writes / reads

```

Report: ESAUSPG          User Storage Analysis
-----
          <---Storage occupancy in pages---> <--Main Storage
UserID   <---Main Storage---> <---Paging---> <-Page Writes
/Class   Total    >2gb    <2GB    Xstor    DASD    Xsto    Disk
-----
00:15:00  103M    102M 480416         0    147M         0 650087
  ***Key User Analysis ***
TCPIP         1355    1346         9         0    2916         0    10
  ***User Class Analysis***
Servers        447     440         7         0   15557         0     40
Velocity       6638    6602         36        0    9208         0      0
LDXmno        11869K  11813K  56412         0  25461K         0   64837
LDXeecs       35965K  35799K 166144         0  22388K         0  123539
LDXsph        16550K  16471K  78323         0  22938K         0   53098
TheUsrs       1360K   1354K   6498          0   8046K         0   53264
  ***Top User Analysis***
LDX304        4892K   4869K  22672         0   5438K         0   21733
LDX131       943437  939060   4377         0   2803K         0    9713
  
```

# User Storage (ESAUSTR) – 6.3+

## User Resident Storage = Active + IBR + Agelist

- UFO Active – User frame list
- IBR – Invalid but resident
- AgeList - ready for “steal”

```
Report: ESAUSTR          User Storage Analysis
Monitor initialized: 05/31/16 at 00:00:00 on 2827 serial
-----
                <-----Virtual Server Storage (Pages)----->
UserID      Size  Alloc Resi- UFO    <-----IBR-----> <AgeList>
/Class      /Class          dent Activ TOT    <2gb >2gb <2gb >2gb
-----
05/31/16
00:15:00   265M  191M  103M  101M   14K 1033   13K 7609  2.1M
  ***Key User Analysis ***
TCPIP      32768  3837  1355  1018  55.9 12.0  43.9   0.1  281
  ***User Class Analysis***
Servers    90112 15625   447   9.0 27.0  2.0 25.0  1.1  410
Velocity   108K 13363  6638  3791  359  128  231  3.9 2485
LDXmno     29.4M 27.5M   12M 11.6M  625 13.0  612  583 238K
LDXecs     55.6M 47.2M   36M 35.7M  531 14.8  516  981 230K
```

# Shared Storage (DCSS) Requirements

Report: ESADCSS      NSS/DCSS Analysis      Velocity Software Corporate      ZMAP 4.2.0 05/29/14      PPg 1482  
 Monitor initialized: 05/28/14 at 15:00:00 on 2094 serial 53E5D      First record analyzed: 05/28/14 15:00:00

Name	Spool ID	Creation		First Use		Non-Shrd		Non-Resident		Locked		PagedOut		Shared		Stolen	DASD		ExStore				
		Date	Time	Date	Time	Shrd	Shrd	<2GB	>2GB	HOST	<2G	>2GB	DASD	XSTOR	Trans		In	Out	In	Out	Mig		
CMS	73	06/10	11:15	05/28	12:34	27	0	1298	0	2	27	0	0	1184	128	23.4	23.4	0.9	0.9	23	23	0.0	
CMSFILES	9	07/01	09:34	05/28	12:34	4	0	768	0	0	4	0	0	635	57	2.9	2.9	0.2	0.2	2.7	2.7	0.0	
CMSPIPES	66	05/24	20:13	05/28	12:34	31	0	256	0	0	2	0	0	256	36	0.5	0.5	0.2	0.2	0.4	0.5	0.0	
CMSVMLIB	65	05/24	20:13	05/28	12:34	31	0	256	0	0	2	0	0	249	28	0.8	0.8	0.1	0.1	0.7	0.8	0.0	
GCS	63	05/24	19:56	05/28	12:34	2	0	120	917	1	5	0	0	1036	13	1.3	1.3	0.0	0.0	1.3	1.3	0	
INSTSEG	5	07/01	09:34	05/28	12:34	31	0	768	0	0	0	0	0	640	0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>MONDCSS</b>	<b>32</b>	<b>02/19</b>	<b>10:18</b>	<b>05/28</b>	<b>12:34</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>16K</b>	<b>2</b>	<b>121</b>	122	0	105	3	4	94.6	26.2	0.3	0.4	22	26	0.0
NLSAMENG	68	05/24	20:13	05/28	12:34	18	0	256	0	0	0	0	0	128	0	0	0	0	0	0	0	0	
SCEE	17	07/01	09:45	05/28	12:34	3	0	256	0	0	1	0	0	128	7	0.4	0.4	0.0	0.0	0.4	0.4	0	
SCEEX	67	05/24	20:13	05/28	12:34	4	0	2048	0	0	3	0	0	1850	36	0.7	0.7	0.0	0.0	0.7	0.7	0	
ZMON	55	10/12	13:07	05/28	12:34	3	0	0	16K	5618	0	0	0	0	0	0	0	0	0	0	0	0	
ZVWS	56	01/10	12:40	05/28	12:34	6	0	0	256	0	0	0	0	55	0	0	0	0	0	0	0	0	
<b>Totals:</b>						163	0	6026	34K	5622	164	122	0	105	6164	309	124.6	56.3	1.9	1.9	50	55	0.1

- “Pages Res” shows how many pages in storage
- “Shared Trans pages/second” is page reads
- “Stolen pages/second” is page writes (DASD or Xstore) Due to Available List Replenishment algorithms,



PGMBK is page table for virtual storage

PGMBK storage per referenced 1MB segment:

Two 4k page PGMBK per 1MB segment (8mb/GB)

- 2048 pages/gb (100GB virtual requires 800mb real)

(1GB Linux server: 8mb PGMBKs)

Locates all user pages in

- Expanded Storage (pre z/vm 6.3)
- DASD Paging (and IBR list)
- Main Storage

A pageable PGMBK is eligible for page-out when it maps no virtual pages into real storage.

# CP Page Tables Requirements

The CP address spaces are reported on the Shared Address Space Analysis

- Resident page tables relative to active pages.
- 128 PTRM address spaces defined in z/VM 6.3
- Only PTRM address spaces in use are displayed

Virtual disks in storage (V-disks) are system-owned address spaces.

```

Report: ESAASPC          Shared Address Space Anal Z/VM 6.3          ZMAP 4..4 06/01/16
-----
                                <--Size-->   <-----Address Space Pages-----> <---Pages pe
                                <-(pages)->   <Resident> <Locked-> <PagedOut> Sto- <DASD->
Owner   Space Name          Avg   Max   <2GB >2GB <2GB >2GB DASD XSTOR len  In Out
-----
00:15:00
SYSTEM  PTRM0000          1049K 1049K  1210  709K    0    0  338K    0  2.5  3.2  5.1
SYSTEM  PTRM0001          1049K 1049K   565  627K    0    0  393K    0  1.3  3.8  2.7
SYSTEM  SYSTEM            524K  524K    0    90     0    0    2     0  0.0  0.0  0.0
-----
System Totals:
Virtual Disk_Subset          0     0     0     0     0     0     0     0     0     0     0     0
-----
00:30:00
SYSTEM  PTRM0000          1049K 1049K  1209  708K    0    0  340K    0  1.7  4.3  3.4
SYSTEM  PTRM0001          1049K 1049K   563  626K    0    0  395K    0  1.6  1.4  3.2
SYSTEM  SYSTEM            419K  524K    0    35     0    0    2     0  0.1  0  0.1
-----
System Totals:
                                241M  241M  1772 1333K    0    0  734K    0  3.4  5.7  6.8
    
```

## Minidisk cache defaults to “all of it”

- MUST BE CONTROLLED!!! Very common configuration error

## Minidisk cache should be enabled for “velocity”!!!

- CMS makes good use of MDC

## Example is “very constrained”, why?

- 2M pages, 8gb
- MDC consumes almost 1M pages (4GB)
- Set to 128MB SET MDC STORAGE 128M 128M
- (or 10mb plus 1mb per linux server)

Report: ESASTR1                      Main Storage Analysis    zVM1 Prod.

---

Time	Users <----->		Pages----->													
	Loggd On	System Storage	Fixed Store	Non-Pgble	Free Stor	Frame Table	<Available> <2gb >2gb	System ExSpc	User Resdnt	NSS/DCSS Resident	<-AddSpace> System User	VDISK Rsdnt	<MDC> Rsdnt	Diag 98		
01:05:12	34	2064368	2970	28013	3802	16128	3186 114	7909	1075K	783	2717 19156	24	905K	1913		
01:20:12	34	2064368	2970	27954	3802	16128	2633 105	7926	1045K	569	2472 19054	5	938K	1913		
01:35:12	34	2064368	2970	27721	3802	16128	3860 139	7916	1120K	1357	6683 19255	647	857K	1913		
01:50:12	34	2064368	2970	27734	3802	16128	3145 111	7897	1178K	1714	20460 19571	4882	787K	1913		

# Available List Replenishment

The Available List is the source of page frames that are ready for immediate allocation.

Casual replenishment of available list:

- LOGOFF
- Page release diagnose x'214'
- MDC Invalidates

Demand Scan replenishment:

- Invoked when Available List frame count < low threshold
- Runs until Available List frame count = high threshold

3 types of scan (pre 6.3)

- Scan 1
- Scan 2
- Emergency (yes, it's bad)

## Each page of storage has a key (4 bits)

- 3 status bits Usually expressed as two hex digits with the last bit always 0.
- For storage alteration, the key on the page must match the key in the PSW (bits 8-11).

## Flags

- F – fetches are protected, as well
- R – some location has been referenced
- C – some location has been changed

## Pre 6.3,

- Re-order sorts user pages in based on R, and C



## Reorder Invoked after CPU time per machine

- Time same for all users
- Interval dynamic

## Reorder

- Resets reference bits
- Segregates list of referenced and unreferenced

Repeated calls cause unreferenced pages to sink in the list.

## Special reorder task every 30 seconds

More frequent in z/VM 5.2 to reduce Linux in queue impact

Cost 1 CPU second per 8GB resident

Known to “stop” large servers for many seconds

Gone in z/vm 6.3

Check MDC address space if over “ideal” (arbiter chosen) size,

- take  $\text{Min}(1\text{M}, 1/8 \text{ of pages})$  on LRU

Pageable PGMBK's if now pageable

- **Paged PGMBK Aging List (PPAL)**
- **Deferred PGMBK Paging List (DPPL)**

Round robin unreferenced pages (one of these on each call)

- Shared address spaces (owned by SYSTEM, DCSS/NSS address spaces, shared user address spaces)
- **Dispatch List user address spaces (unreferenced)**
- Eligible List (unreferenced pages)
- **Dormant List**

Traditional method for page replenishment

- Only useful in Linux environment where servers drop from queue

Dormant list (Dormant < 20 seconds)

- **Only unreferenced pages**

Dormant List (Dormant > 20 seconds)

- Linux never goes dormant > 20 seconds
- Unreferenced and referenced pages are taken

# Demand Scan – Passes 2 and 3

## Pass 2: Referenced pages take from:

- Dormant List (from current pointer)
- Eligible List (back to front)
- Dispatch List (back to front) Q3 users only (LINUX, TPF)

## On pass 2,

- Leave RESERVE count of pages
- Take Shared and System (including vdisk)
- Normal scan on Linux systems

## On pass 3, (emergency scan)

- take anything that's not nailed down.
  - Monitor buffers
  - CMS saved segments
- Very common when linux servers do not drop from queue

# z/VM 6.3 Storage Management

## The problem to be solved:

- Linux servers remain “active, inqueue” (WAS, DB2)
- Scan 1 completely ineffective –
- Scan 1 steals only from mission critical cms servers (small target)
- Scan 2 steals from saved segments, vdisks
- Scan 2 steals from in queue servers, active **AND** inactive
- Scan 3 the norm, steals rotate thru server list
- Scan 3 steals from monitor DCSS
- **ONLY PAGE LOCKING EFFECTIVE**

## Expanded storage requirement

- Need sufficient staging area because page scan broken
- Most installations configured 20% as Expanded Storage
- Expanded storage buffer only works if 30 second buffer

Scan based management gone

Expanded storage gone

Lists (UFO, SUFO, VUFO) contain two sections

- Active pages
- IBR (Invalid But Resident)

**IBR Pages:**

- Invalidated, so not addressable by address space
- Page Status table entry flag
- Pages “stolen” from end of IBR page list

**Re-Referenced pages “validated”**

- Page fault causes page to move to “active Pages”
- Page moved to top of “active list”

# z/VM 6.3 Storage Management

## New functions:

- IBR – Invalid but resident, replaces re-order
- Global Aging list – sorts pages LRU
- Saved segments can now be protected

## IBR List – pages belonging to user in “test” mode

- Replacement for expanded storage
- Pages marked IBR (Invalid but resident) as a “test Pages can be reclaimed if referenced
- Referenced pages go to top of frame list
- **Size is configurable (to 5%)**

## Global Aging List – list ready to page out

- Pages moved from IBR to Global Aging list
- Pages can be reclaimed if referenced
- Pages will be oldest pages to be paged out

# z/VM 6.3 Storage Management

## Impacts of New functions:

- Reserved pages stolen less frequently
- Reserve now applies to SUFO (shared address spaces)
- Monitor dcss can now be protected
- Pages in storage, ready for “steal”
- Pages may still be reclaimed by re-reference

## Pre-write

- Some percent of Global Aging list pre-written
- Pages pre-written make slots quickly available
- “staying ahead” of storage requirements

## Available list thresholds calculated on rate

- High threshold removed



# z/VM 6.3 Storage Management

Per userid: zwrite 20mb, 5mb touched, all resident

```
cp q reserved zwrite
```

Resource	Type	SPID	Reserved	Resident	Instantiated
ZWRITE	USER	----	20000K	4876K	4876K

Per dcss (mondcss)

▪ **Replaces Velocity Software's page locking???**

```
cp set reserved dcss mondcss 1000
```

Resource	Type	SPID	Reserved	Resident	Instantiated
MONDCSS	DCSS	0011	4000K	364K	364K

Instantiated pages

▪ Any pages touched, and now backed

# z/VM 6.3 Storage Management

## System Age List

- Is target large enough?
- SET agelist size 3%

Report: ESASTR3                      Storage IBR / DPA Analysis                      Velocity Sof  
Monitor initialized: 07/16/13 at 15:27:11 on 2817 serial FE0A6                      First record

```
-----<br> <-----Age List in Pages-----> <- Page--><-Invalidate-> <-ReVali<br> <Ready to Reclaim> <Not> <ReWrites><-Pages/Sec--> <-Pages/<br>Time        Target size Ready Chngd Refd    Ready Chgd Refd Priv Vdsk Shrd Priv Vds<br>-----<br>15:29:00    2745K    1M   196K            0    0 703155    0    0 8182    0   152 1463<br>15:30:02    2745K  947K   140K            0    0 645887    0    0 4637    0   136 1799<br>15:31:00    2745K  951K   126K            0    0 641654    0    0  656    0    12  911<br>15:32:00    2745K  704K 90858            0    0 485985    0    0  615    0    15  568<br>-----</pre>
```

# z/VM 6.3 Storage Management

## User frame owned list, IBR, Agelists

```
Report: ESAUSTR          User Storage Analysis          ri 0   Ve
Monitor initialized: 07/16/13 at 15:27:11 on 2817 serial FE0A6   Fi
-----
          <-----Virtual Server Storage (Pages)-----> <Resident> Page
UserID   Size   Alloc Resi- UFO   <--IBR--> <AgeList> <Unreferd> slots
/Class          dent Activ TOT   >2gb <2gb >2gb <2gb >2gb  used
-----
15:29:00 265M 169M 134M 133M 1756 1029 217 1.2M 142 331K 41M
***User Class Analysis***
Servers   117K 18119 2060 454.0 287 256 0 1319 0 120 17421
ZVPS     49152 5409 2305 6.0 18.0 18.0 0 2281 0 13.0 4988
Linux    14336 1980 96 3.0 6.0 6.0 0 87.0 0 0 1929
TheUsers 265M 169M 134M 133M 1240 618 217 1.2M 142 331K 41M
***Top User Analysis***
LNXT007 52.4M 39.2M 23M 22.1M 16.0 16.0 57.0 499K 0 241K 18M
LNXT009 52.4M 43.4M 36M 35.6M 166 20.0 0 27K 0 2220 8979K
LNXT010 52.4M 27.2M 24M 23.6M 188 50.0 90.0 160K 89.0 55K 4761K
LNXT013 52.4M 36.5M 30M 30.2M 212 92.0 1.0 175K 0 8116 7331K
LNXT011 52.4M 22.1M 22M 21.6M 14.0 14.0 53.0 167K 53.0 15K 1583K
```

## Limit virtual machine sizes

- PGMBKs – cost 8mb (ptrm address space) per virtual GB
- CMM

Limit the amount of main storage used by MDC set one of:

- SET MDCACHE STORAGE minM maxM
- SET MDCACHE STORAGE BIAS 0.nn

# Managing Storage Demand

Use SET SRM to control aggregate demand for storage.

- SET SRM STORBUF q1 q2 q3
- SET SRM XSTORE pct
- **Defines OVER COMMIT RATIO**

Based on user's projected working sets divided by available DPA.

Recommendation from all "experts" is to disable since release

- SET SRM STORBUF 350 300 300

Default: 125 85 75

- 100 for Q1+Q2+Q3
- 85 for Q2+Q3
- 75 for Q3

Default increased in VM/ESA V2.1 to 125 (why?)

Current settings reported on ESASUM

# Managing Storage: SET Reserved

## Traditional:

- Use SET Reserved to fence storage for a user.
- If TCPIP is paged out, who is delayed?

## If ZWRITE paged out, what happens?

- zVPS locks pages, reserved just not sufficient

## Some servers should not page

- Use LOCK when important

## What servers or users should have locked storage?

- Servers/Users that have time dependencies
- Servers with multi-user impact

SET RESERVED command, Privilege class A

Fencing Storage used to protect address spaces

- Set REServed USER userid frames
- SET RESERVED NSS filename frames
- SET RESERVED DCSS filename frames

Enables fencing of

- Mondcss
- CMS, etc

# DPA Suggestions: Linux Servers

Reduce Virtual Machine sizes

Tune application/Server for VM

Use MDC for shared caching

Use Virtual Disk for Swap



## Page stealing prior to z/VM 6.3

- Note where pages stolen from!!!
- What “very few” users are dormant?
- Pages stolen from very few nicely behaved “dormant” users
- Pages stolen from very SMALL saved segments

```

Report: ESAME           Memory Extension Analysis           ZMAP 4.2.0   1470
Monitor initialized: 05/28/14 at 15:00:00 on 2094 serial 53E5D
-----
      <---Users----> <Pgs Resident> <---Emergency Page Steals/Sec--> <---Scan Two Page Stls/
      <-avg number-> <Above 2GB Ln> Dormant  Disp  NSS/ System Elig Long  Dorm  NSS/ DISP
Time    Actv In Q   on <User>  <MDC->  Users  Users  DCSS  Pages list  Dorm Users  DCSS Users
-----
15:01:00  131  107  210 44473K      0  1041  2160  108    0    0    0  4034    0  9017
15:02:00  131  107  204 44478K      0   852  1953  170    0    0    0  4616    0  8385
15:03:00  131  106  196 44396K      0  1076  1079  150    0    0    0  4538    0  6623
15:04:00  131  104  200 44101K      0   602  1189  108    0    0    0  3537    0  5055

```

## Full storage map available

- System
- User
- Address space / vdisk
- MDC

Manage storage to meet requirements

Tune users as required

6.3 changes lots of things