35 YEARS OF PERFORMANCE MANAGEMENT

Understanding Linux Real Memory Analysis and Tuning

- Barton@VelocitySoftware.com
- HTTP://VelocitySoftware.com

"If you can't Measure it,

I am Just Not Interested ™"

VELOCITY SOFTWARE

PROVEN PERFORMANCE

Memory Subsystem Objectives

Objectives:

- Understand z/VM Memory Requirements
- Understand Linux Memory Requirements
- Know how/where to measure
- Understand Demand Paging
- Determine Requirements
- Understand Measurements
- Suggest tuning measures

z/VM Memory

Real Memory Management

Memory is expensive, overcommit reduces costs

Memory can be limited, use it wisely

Paging objective: Page out idle / unused pages

Inactive Memory? Linux Memory is not idle

- Extra Memory used to cache data and programs

Inactive servers? Linux servers are not idle

- Linux applications poll at 200 times per second
- Which servers are actually doing work if all are “active”?

Memory “overcommit” is no longer a valid metric

Real Memory (RAM)

z/VM Paging

- Over commitment of Memory causes paging
- **Over commitment of Memory reduces cost**
- Paging is common **(manageable)** performance problem

Linux Swapping

- Swapping result of over commitment of Linux Memory
- Swapping to VDISK very fast, uses z/VM real Memory when it happens
- Swapping to DASD very slow, always noticeable

Understanding Linux ram (real Memory) will save gigabytes of real Memory

Understand the Memory Map - ESASTR1

Memory Map to show Memory (14GB) use

- User resident should be major use
- Control MDC, understand VDISK
- If you have an “event”, what changed?

Capture ratio shows accuracy

Report: ESASTR1 Main Memory Analysis Velocity Software Corporate ZMAP 5.1.2 04/16/21 Pg 2
Monitor initialized: 04/15/21 at 00:00:00 on 8562 serial 040F78 First record analyzed: 04/15/21 00:00:00

Time	Users Loggd On	Pages: System Memory	Fixed Store	Non-Pgble	Free Stor	Frame Table	<Available> <2gb	>2gb	Systm ExSpc	User Resdnt	NSS/DCSS Resident	<-AddSpace> Systm	User	VDISK Rsdnt	<MDC> Rsdnt	Diag 98	Over Commit Ratio	Capt-Ratio
04/15/21																		
17:30:00	111	3670016	2878	20883	1166	28672	3164	2669	52296	3383K	35077	75714	0	4307	19741	16K	3.653	0.988
17:45:00	111	3670016	2878	20872	1147	28672	3195	2389	52298	3381K	35074	75716	0	4270	21989	16K	3.653	0.988
18:00:00	111	3670016	2878	20889	1146	28672	3128	2851	52306	3383K	35079	75722	0	4103	19648	16K	3.653	0.988
18:15:00	113	3670016	2878	20876	1141	28672	3077	2508	52316	3384K	35099	75776	0	4028	19283	16K	4.609	0.988
18:30:00	116	3670016	2878	20880	1075	28672	3137	2544	52360	3349K	32071	122K	0	2118	12337	16K	7.354	0.988
18:45:00	116	3670016	2878	20808	1038	28672	3051	2234	52407	3293K	29914	196K	0	0	47	16K	8.227	0.988
19:00:00	116	3670016	2878	20765	1028	28672	3056	2245	52414	3293K	29082	196K	0	0	127	16K	8.227	0.988
19:15:00	115	3670016	2878	20797	1040	28672	3063	2232	52409	3297K	29522	192K	0	22	73	16K	8.754	0.988
19:30:00	116	3670016	2878	20809	1031	28672	3069	2235	52450	3293K	29065	196K	0	0	6	16K	9.363	0.988

What Virtual Machine is using your Memory

Virtual Machine Memory analysis – ESAUSP2 (percent/rate)

- Analyze by user – Large consumers?
- RHOS* users paging too much to get work done
- RHOS* is OpenShift installation

Report: **ESAUSP2** User Resource Rate Report Velocit

	<---CPU time-->			<----**Main Memory (pages)**----->			<-Paging (pages)-					
UserID	<(Percent)>		T:V	<Resident>		Lock	<-----WSS----->			Paged	**<Pgs/Second**	
/Class	Total	Virt	Rat	Totl	Activ	-ed	Totl	Activ	Avg	2Disk	Read	Write
18:30:00	145.3	133.9	1.1	**3.3M**	3348K	7048	3.9M	3909K	34K	9147K	27057	15496
***Key User Analysis ***												
TCPIP	0.15	0.05	3.0	1422	1422	601	817	817.3	817	7750	**43.4**	8.6
User Class Analysis												
Velocity	5.82	5.43	1.1	3763	3598	5	4593	4271	534	14472	137.4	57.0
SUSE	20.17	19.28	1.0	112K	112K	1534	193K	193K	32K	1048K	2754	828.5
ORACLE	4.66	3.84	1.2	195K	195K	734	381K	381K	190K	473K	2895	936.7
GPFS	12.51	11.68	1.1	195K	195K	975	439K	439K	146K	1332K	4008	1383
TheUsrs	95.37	89.07	1.1	**2.6M**	2615K	1145	2.5M	2472K	80K	5017K	12958	11022
Top User Analysis												
RHOSBOOT	39.91	38.51	1.0	727K	727K	30	99K	98642	99K	454K	**1175**	**2346**
RHOSCP2	8.92	8.20	1.1	250K	250K	19	116K	116K	174K	201K	**997.0**	**1965**
RHOSCP1	8.78	8.05	1.1	252K	252K	19	126K	126K	189K	205K	**967.6**	**2005**
RHOSCP3	7.83	7.04	1.1	161K	161K	28	48K	47842	80K	125K	**1230**	**1157**

z/VM Memory and Paging Architecture Moving Target

z/VM shared Memory / Overcommit

- Objective: Page unused pages out to allow re-use
- Need optimal test before paging to slow disk
- Optimize page-in when needed (block paging)

The problem? Which servers, which pages are truly idle?

Architectures to choose from:

- Excessive Memory – enough so no paging (expensive)
- Solid State paging device – sort of fast (SSD, Flash)
- Disk paging devices – not fast

Linux Memory Challenging

The problem: What pages to page out?

- Traditionally, “idle users” get paged out

Inactive Memory? Linux Memory is not idle

- Extra Memory used to cache data and programs
- Linux applications poll at 200 times per second
- Which servers are actually doing work if all are “active”
- What pages can be legitimately paged out of real Memory?

Determining pages for page out:

- **Active server?** Can not know if server is working or **polling**
- Take least recently used, non modified, non referenced
- Fast page-in (page recovery) very important

Managing Linux Memory in z/VM

Strategy / best practices in past if overcommit high

- Need high speed page recovery

~~Expanded Storage was used for "30 second test case"~~

- Pages migrated to disk after 30 seconds
- Minimum 20% of Memory reconfigured to Expanded Memory
- Page-in from expanded Memory was **synchronous,** FAST
- Pages migratable to disk after 30 seconds unreferenced

"New" strategy is IBR (z/VM 6.3)

- Invalid But Resident
- Recall very, very fast
- VERY LIMITED! 5% is the max, will reduce impact of paging
- 2% is the default, Go the max!

z/VM Memory Management Options

System Age List - objective is to reduce delay for page in

- Maximum 5%,
- Recommend 5% always
- SET AGELIST SIZE 5% EARLYWRITES YES KEEPSLOT YES

```
-Set--AGELIST---.-SIZE--.-n.n--PERCent-.-.
                |       |-n.n%---------| |
                |       '-storsize-----' |
                |-EARLYWrites--.-Yes-.---|
                |              '-No--'   |
                '-KEEPSlot--.-Yes-.------'
                            '-No--'
```

- CP QUERY AGELIST (default)

```
Target size    =          280576K (274M)      2.0% of pageable Memory
In use         =          271712K
Pending writes =          120296K
Early writes   = Yes
Sizing         = Variable
```

Sizing Large z/VM: Pageable PGMBKs

PGMBK is page table for virtual Memory (CP Overhead)

PGMBK Memory per referenced 1MB segment:

- Two 4k page PGMBK per 1mb segment (8mb/gb) (about 1%)
- 2048 pages/gb (100mb virtual requires 800mb real)
- (1gb Linux server: 8mb PGMBKs)

Locates all user pages in

- DASD Paging (and IBR list)
- Main Memory

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

z/VM Summary

- **z/VM Real Memory Guidelines**
  - SET MDC MIN 128m MAX 128m
  - Minimize server sizes
  - SET AGELIST SIZE 5%
  - Ensure accounting is disabled
  - Validate page space

Linux System Memory

Linux Memory

Linux Cache

- Linux avoids I/O by using cache
- Linux will cache gigabytes of data if allowed
- Oracle SGA MUST fit in Linux page cache
- MongoDB Memory is in Linux page cache
- Swap historically was slow SCSI device so Memory oversized

Reduce size of Linux Virtual Machine MAJOR Knob.

- Reducing virtual machine size reduces caching of old data
- Define virtual disk for swap
- Virtual Disk paged out when not in use
- Swapping is ok if configured correctly

Linux Memory Analysis

Different levels, different data sources

High Level, UCD (UCD MIB, standard Linux (top))

- Standard Linux system Memory at a high level - ESAUCD2

Linux system Memory (Velocity MIB, more details)

- Linux system Memory details – ESALNXR
- Anonymous details (Inactive)

Linux process Memory (what is using swap space?)

- By process: swap
- Total of process: page tables

Linux System Memory Reporting

Linux Memory – ESAUCD2 – Kernel, page tables not reported (CMM is VSI)

```
Report: ESAUCD2       LINUX UCD Memory Analysis Report                Velocity Sof
-------------------------------------------------------------------------------
Node/     <-------------------------Memory Sizes (in MegaBytes)----------------
Time/     <--Real Memory--> <-----SWAP Memory----> Total <-------Memory in Us
Date      Total  Avail Used  Total Avail Used  MIN   Avail CMM    Buffer Cache O
-------- ------ ----- ----- ----- ----- ----- ----- ----- ------ ------ ----- -
18:30:00
*** Nodes *******
lxsugar   999.4   7.6 991.8 154.9 151.3   3.6  15.6 158.9      0   85.7 648.1 2
mail     8112.8  2318  5795     0     0     0  15.6  2318      0  639.8 907.9
mongo01  3849.8 983.3  2866 371.9 309.6  62.3  15.6  1293      0  150.6  1130
opensuse  15846 160.1 15686  8192  8192   0.3  15.6  8352      0 1524.5  8392
REDHAT6X  996.8  13.8 983.0 495.8 380.4 115.5  15.6 394.2      0  114.7 724.1 1
redhat7   994.0 411.5 582.4  1124  1124     0  15.6  1535      0    1.1 472.6 1
rhel64v   996.1  66.3 929.8  2047  2034  12.5  15.6  2101      0  103.3  39.6 7
rhel7v   2002.3 101.2  1901  2064 766.0  1298  15.6 867.2      0      0 253.0
sles11v3  868.8  88.0 780.8  2046  1406 639.7  15.6  1494      0    3.3  27.7 7
sles11x3  493.2 132.8 360.4 867.9 867.9     0  15.6  1001      0  141.6 149.5
```

Linux System Memory Details Reporting

Linux Memory details – VSI MIB, ESALNXR

```
Report: ESALNXR       LINUX RAM/Memory Analysis Report                Velocity Sof
Monitor initialized: 04/15/21 at 00:00:00 on 8562 serial 040F78     First record
-------------------------------------------------------------------------------
Node/     <--------Memory in megabytes-------------> <-Kernel(MB)-> <-Buffers(MB
                  <---Cache----><---Anonymous---> Stack<-Slab-->
Time      Total Free Size Actv Swap Total Actv Inact Size Size SRec Size Dirty B
--------- ----- ---- ---- ---- ---- ----- ---- ----- ---- ---- ---- ---- ----- -
18:30:00
mongo01    3850  983 1130  939 26.9  1464 1333 201.3  3.5 57.3 46.3  151   0.7
opensuse  15846  160 8392 4346  0.1 915.4  426 512.0  6.2  554  477 1525   0.0
REDHAT6X  930.4 13.0  676  308  2.5  41.8 62.0 154.7  2.7 51.5 41.0  107   0.0
redhat7   994.0  412  473  328    0  40.8 40.9  56.0  2.6 46.8 26.9  1.1     0
rhel64v   996.1 66.2 39.6 74.6  1.2  14.0  1.1  13.9  1.8  101 42.9  103   0.0
rhel7v     2002  101  253  105 10.0  1437 1142 407.7  4.0  112 67.7    0   0.0
sles11v3  868.8 88.0 27.7 17.4 51.6 106.0 44.6  69.6  2.6 35.6  8.5  3.3   0.0
sles11x4  492.8  102  235  160    0  26.8 26.8   0.7  1.4 31.2 23.2 78.1   0.0
sles12     3374  124 2259 1557  2.7 534.0  483 459.6 30.8  153 51.8  110   0.1
sles12v   995.6  101  440  206  8.1 339.2  162 230.1  2.0 73.9 51.2  0.0   0.0
sles12x3  820.9  182  334  377    0  38.5 38.7  42.2  2.5 88.9 70.5  154   0.0
```

Understanding Linux Memory

Linux admins oversize

Linux data shows

- Real Memory
- Available Memory
- Swap Memory
- "Cache"

Some Swapping is "good"

If not swapping,

- Reduce VM size
- **Use CMM to reduce**

Watch for opportunities

- HIGH available
- No swap

```
Report: ESAUCD2       LINUX UCD Memory Analysis    Velocity Software Corpo
Monitor initialized: 10/03/14 at 07:22:27 on 2    First record analyzed:
-------------------------------------------------------------------------
Node/     <--------------------Memory Size (MB)-------------------------->
Time/     <--Real Memory--> <-----SWAP Memory--Memory in Use--------->
Date      Total  Avail Used  Total Avail Used   Buffer Cache Ovrhd Shared
--------  ------ ----- ----- ----- ----- -----  ------ ----- ----- ------
07:24:00
ORAap042 8041.5 475.9  7566  1130  1130   0.1   183.5  1512  5870      0
ORAap044  13069  7131  5939  6888  6888     0   233.0  3913  1793      0
ORAap046 8041.5  2091  5951  1130  1130   0.1   260.9  3423  2267      0
ORAap048 8041.5  2291  5751  1130  1130     0   224.8  3347  2179      0
ORAap050 8041.5 529.3  7512  1130  1130   0.1   186.9  1577  5749      0
ORAap052  10046 642.8  9403  8172  8172     0   226.5  3958  5218      0
ORAap054 8041.5  1235  6807  3036  2878 158.3   139.9 319.3  6348      0
ORAap056 8041.5 818.5  7223  5604  5592  12.2   156.4 968.3  6098      0
ORA1101b  12062  64.0 11997  4942  4758 183.6   727.5 10024  1246      0
ORA1201a  12062 218.9 11843  4942  4438 503.7   152.4  7170  4520      0
ORA1202a  12062  1668 10394  4942  4399 543.3   137.3  6435  3822      0
ORA1203a  12062  94.0 11968  4942  4443 498.5   168.6  7582  4216      0
ORA1204a  12062  90.9 11971  4942  3754  1188    70.9  8088  3811      0
ORA1403a  12062 462.1 11599  4942  4420 521.8   180.6  6783  4636      0
ORA1404a  12062 439.3 11622  4942  4442 499.9   103.4  6853  4666      0
ORA1405a  12062 442.5 11619  4942  4471 471.1   127.0  6593  4899      0
WAS2a016 2502.6  89.6  2413  1130  1106  24.2   203.0 243.0  1967   48.0
WAS2a020 2502.6  29.9  2473  1130  1106  24.1   254.3 238.8  1980   47.9
WAS2a024 5520.4  2635  2885  1130  1130     0   776.4 613.3  1496   50.3
WAS2a054 2502.6  22.0  2481  1130  1106  23.4   247.9 274.1  1959   48.5
WAS2a058 2502.6  22.4  2480  1130  1106  23.5   244.5 254.9  1981   48.5
WAS2a062 6528.3  3687  2841  1130  1130     0   762.0 591.8  1487   50.3
WAS2a114 2502.6  17.7  2485  1130  1106  23.6   219.6 267.6  1998   48.4
WAS2a118 2502.6  17.6  2485  1130  1106  23.6   260.5 264.1  1960   48.2
```

Linux Swapping

Linux Swapping

Reducing virtual Memory size may cause swap

- Linux does not swap until out of Memory

Swapping to disk VERY VERY SLOW

- Other platforms increase Memory size because disk is slow
- **Swap to disk if you want to penalize a server**
- Max swap rate maybe 200/second on a very good day

Linux Swapping to VDISK

- Not a performance degradation
- 40,000 / second is FAST

Redefine Linux Swap disks as z/VM Page packs

Linux Swapping

Swap Guideline:

- **Define 2 virtual disks, prioritized swap**
- **First one “smaller”, second one 2GB (insurance)**
- More (vDISK) swap devices for SAP as needed (they are essentially free)
- Use DIAG driver instead of FBA - Reduces I/O by factor of 8

VDISK Case Study (ESAASPC, ESAVDSK)

VDISK for swap best practice: Two disks, prioritized – DOUBLE CHECK!

- Two disks per server, goodness
- Should be 1 small swap disk, plus 2nd large disk, goodness
- Prioritized backward though, badness….
- (Address space names have server, virtual address and index)

Owner	Space Name	DASD AddSpc Pages	X-VDSK Blks	Resi-dent	Lock-ed	T	Stg-> Migr	Page Slots	Store Blks
Average:									
LINUX1	VDISK$LINUX1$$$0101$0041	65791	8738	3.0	0		0	568	0
LINUX1	VDISK$LINUX1$$$0112$0042	524K	69905	170	0		0.0	61212	11
LINUX2	VDISK$LINUX2$$$0101$0043	65791	8738	3.0	0		0	571	0
LINUX2	VDISK$LINUX2$$$0112$0044	524K	69905	85K	0		0.4	346K	2047
LINUX3	VDISK$LINUX3$$$0101$0045	65791	8738	3.0	0		0	571	0
LINUX3	VDISK$LINUX3$$$0112$0046	524K	69905	2.0	0		0	5767	0
LINUX4	VDISK$LINUX4$$$0101$0047	65791	8738	3.0	0		0	571	0
LINUX4	VDISK$LINUX4$$$0112$0048	524K	69905	147K	0		0.3	223K	35967
LINUX5	VDISK$LINUX5$$$0101$0049	65791	8738	3.0	0		0	568	0
LINUX5	VDISK$LINUX5$$$0112$004A	524K	69905	2.0	0		0	4321	0
System Totals:		5901K	39321	233K	0		0.7	669K	38631

Linux Summary

Linux

- CMM When valid
- Minimize RAM
- Use VDISK (two) for swap

Linux Process/Application Memory

Linux System Memory Details Reporting

Linux Process Memory details – VSI MIB, ESALNXP

Report: **ESALNXP** LINUX HOST Process Statistics Report Velocity Software Co
Monitor initialized: 04/15/21:00 on 8562 serial 040F78 First record analyze

node/ Name	<-Process Ident-> ID	PPID	GRP	N<------CPU Percents----> V	Tot	sys	user	syst	usrt	<--------------- Memory Size	RSS	Peak	Swap	Data
18:30:00														
mongo01	0	0	0		14.8	1.18	13.2	0.03	0.31	7248	1544	113K	727	78K
mongod	10889	1	10887		5.75	0.60	5.15	0	0	2653	1307	40K	429	37K
java	51013	8515	8515		4.94	0.31	4.62	0	0	1665	155	16K	0	14K
java	51596	8515	8515		3.61	0.20	3.41	0	0	1743	186	8985	0	8053
opensuse	0	0	0		10.0	8.75	1.26	0.00	0.01	33K	5900	537K	0	38K
gsd-colo	1909	1791	1776		1.13	0.00	1.13	0	0	706	84	11K	0	1773
VBoxHead	24298	24280	24298		8.61	8.61	0	0	0	5824	4237	87K	0	2089
REDHAT6X	0	0	0		0.72	0.34	0.27	0.07	0.05	16K	1205	227K	641	14K
rhel7v	0	0	0		2.46	0.41	1.69	0.25	0.11	43K	1643	676K	20K	252K
java	2028	1	1321		1.22	0.04	1.18	0	0	3848	865	58K	2054	55K
sles11v3	0	0	0		0.65	0.19	0.46	0	0	6526	117	105K	9009	27K
sles12	0	0	0		4.60	0.72	3.84	0.03	0.02	76K	5518	1.0M	2918	178K
ora_mmon	2596	1	2596		3.61	0.32	3.29	0	0	896	403	11K	16.3	1155
sles12v	0	0	0		0.52	0.16	0.32	0.01	0.03	15K	379	239K	10K	144K

Totals by node

Processes > thresh

Includes swap > 0

(Re)-Sizing Memory for "MongoDB" Server

Sizing Objective: Best performance at lowest cost

- Minimize memory requirements
- Minimize swap
- Minimize paging

First, evaluate existing z/VM Memory

- Resident
- Paged out
- VDISK resident

```
Screen: ESAUSPG  Velocity Software - VSIVC1      ESAMON 5.14
1 of 2  User Memory Analysis                    CLASS * USE

                    <-Memory Occupancy (MB)-->
          UserID    <---Main Memory--->  Paged <-Page I/O-->
Time      /Class     Total  >2GB   <2GB    Out  Writes  Reads
-------- -------- ------ ------ ------ ------ ------ ------
19:10:00 MONGO8PR 933.89 680.07 253.82 452.43      3      5
         MONGO8S1 936.31 704.77 231.54 341.56      8     58
         MONGO8S2 468.02 348.38 119.64 528.86     57    114
         MONGO804 1610.4 1208.2 402.22 995.58     26     29
         MONG505A 159.35 115.25  44.10 348.14      4      8
```

Profiling MongoDB Virtual Machine Memory

Report: **ESAMNGP** MONGODB RAM Profile Report Velocity Soft
Monitor initialized: 12/12/25 at 00:00:20 on 3931 serial 08B2A8 First record

Server Name	Current Connects	VMRES (mb)	<Linux System Ram/ Anonymous (mb)> Size	Cache	**Avail**	Activ	**Inact**	**Swap**	<Process RAM (MB)> SIZE	RSS	Data
00:02:00											
IGT000FC	55	8684	30195	11259	15487	36	2050	254	2981	28885	5410
IGT000FE	75	8061	30195	13702	3826	122	11124	497	13.5	101.9	20.5
IGT001A0	3195	423K	480K	173K	174K	25424	106K	1083	132K	249K	215K
IGT001A1	0	351K	480K	354K	116K	1	7718	4	8719	45250	11198
IGT001A3	78	475K	480K	413K	2892	117	64172	119	65160	99K	67824
IGT001B4	63	9928	60939	11070	45919	84	1941	352	3053	35474	5643
IGT0010B	0	8221	30195	11021	15688	24	2092	186	106.4	953.8	136.6
IGT0010F	51	10272	30195	10886	16700	1	1221	1	837.4	12626	998.4
IGT00100	69	19454	30195	27042	488	277	1239	1421	2301	29223	6016
IGT00101	66	8737	30195	11941	14909	27	1940	191	13.5	101.9	20.5
IGT00104	109	11904	30195	15077	11668	65	2004	303	13.6	166.1	20.8
IGT00105	66	8072	30195	10800	15949	82	1987	347	0	0	0
IGT00107	496	15833	30195	19181	7230	34	2356	186	3306	29823	6142
IGT00109	61	8002	30195	10758	16090	139	1855	537	54.2	794.6	93.6
IGT0011B	369	19094	60939	23801	33745	112	2295	445	3318	37156	6342
IGT0011D	138	15651	60939	24695	32485	141	2639	550	3725	34872	6530
IGT0011F	825	32702	60939	26825	14524	121	18413	465	19548	54782	23074

VSI MIB

Why swap?

Mongo takes all

MongoDB Swapping Analysis

```
Report: ESALNXP  cess Statistics Report        Velocity Software Co
Monitor initializ:00:20 on 3931 serial 08B2A8  First record analyze
-------------------------------------------------------------------
node/     <Proces <------CPU Percents----> <---------------Memory
 Name           ID  Tot  sys user syst usrt  Size RSS Peak Swap Data
--------- ------- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
00:02:00
IGT001A0         0 19.6 3.92 14.7 0.78 0.22 255K 135K 260K 1086 221K
 splunkd      5058 0.18 0.05 0.13    0    0 3079  107 3079 47.0  232
 platform    11364 4.88 0.30 4.58    0    0  364  123  314    0 50.7
 BESClien    84395 1.93 0.87 1.07    0    0  626   35  629 0.11 70.5
 mongod-g   114056 8.97 1.17 7.80    0    0 217K 133K 219K 1000 217K
```

MongoDB takes all Memory at startup

Other processes (BigFix, Splunk) start, compete for Memory

MongoDB process swaps out, impacting performance

Swap is not to vDISK but will happen by “design”

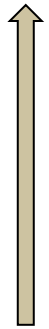
Should Mongo Memory requirements be parameterized?

Solution: Use CMM at startup, then start MongoDB, then pop CMM?

Sizing Memory for "MongoDB" Server: Linux Perspective

Sizing Objective: Best performance at lowest cost

- Buffer and cache were the real targets, this also works for java servers



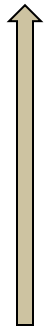
```
Screen: ESAUCD2  Velocity Software - VSIVC1      ESAMON 5.140 06/02 19:
2 of 3  LINUX UCD Memory Analysis Report         CLASS * NODE MONGO8 85

          Node/     <Real Memory (MB)> <-----Memory in Use (MB)----->
Time      Group     Total  Avail   Used   CMM Buffer Cache Ovrhd Shared
--------  --------  ------ -----  ------ ----- ------ ----- ----- ------
20:24:00  MONGO8S2   974.9 610.8   364.2     0    1.0 108.8 254.4    8.1
20:23:00  MONGO8S2   974.9 611.7   363.2     0    0.9 108.0 254.3    8.1
20:22:00  MONGO8S2   974.9 612.5   362.4     0    0.8 107.2 254.4    8.1
20:21:00  MONGO8S2   974.9 613.3   361.6     0    0.7 106.5 254.4    8.1
20:20:00  MONGO8S2   974.9 628.5   346.4     0    0.6  92.3 253.6    8.0
20:19:00  MONGO8S2   974.9 630.3   344.6     0    0.4  90.3 253.9    8.0
20:18:00  MONGO8S2   974.9 634.6   340.4     0    0.3  86.6 253.4    8.0
20:17:00  MONGO8S2   974.9  20.1   954.8 620.0    0.2  81.8 872.8    8.0
20:15:00  MONGO8S2   974.9 334.1   640.8     0   64.0 270.1 306.7   10.1
20:14:00  MONGO8S2   974.9 325.8   649.1   9.8   64.0 269.7 315.4   10.1
20:13:00  MONGO8S2   974.9 334.9   640.0     0   64.0 269.6 306.4   10.1
20:12:00  MONGO8S2   974.9 334.4   640.5   1.0   64.0 269.5 307.0   10.1
```

Sizing Memory for "MongoDB" Server: Linux Perspective

Sizing Objective: Best performance at lowest cost

- VDISK resident + userid resident: dropped 60mb
- Swap space increased 200mb (pushed out anonymous inactive?)
- Server has 600 MB Available, ready to go

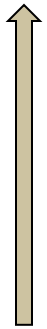


```
Screen: ESAUCD2  Velocity Software - VSIVC1     ESAMON 5.
Node/                <Real Memory  (MB)> <--SWAP Memory (MB
Time      Group     Total  Avail   Used  Total Avail Used
-------- -------- ------ ----- ------ ----- ----- ----- -
20:21:00 MONGO8S2  974.9 613.3  361.6 371.9  56.3 315.5
20:20:00 MONGO8S2  974.9 628.5  346.4 371.9  55.6 316.3
20:19:00 MONGO8S2  974.9 630.3  344.6 371.9  55.1 316.8
20:18:00 MONGO8S2  974.9 634.6  340.4 371.9  53.8 318.0   ←0 mb cmm
20:17:00 MONGO8S2  974.9  20.1  954.8 371.9  53.3 318.5   ←600mb cmm
20:15:00 MONGO8S2  974.9 334.1  640.8 371.9 186.3 185.6
20:14:00 MONGO8S2  974.9 325.8  649.1 371.9 186.3 185.6
20:13:00 MONGO8S2  974.9 334.9  640.0 371.9 186.3 185.6
```

Sizing Memory for "MongoDB" Server

Sizing Objective: Best performance at lowest cost

- VDISK resident + userid resident: dropped 60mb
- But wait, vDISK inactive
- Final savings 170mb



```
Screen: ESAUSPG  Velocity Software - VSIVC1      ESAMON 5.140 0
2 of 2  User Memory Analysis                    CLASS * USER M

                      <-Memory Occupancy (MB)--> <Address Spaces>
          UserID      <---Main Memory--->  Paged <MegaB Resident>
Time      /Class       Total  >2GB   <2GB    Out  VirtDisk AddSpce
--------  --------  ------ ------ ------ ------ -------- -------
21:23:00  MONGO8S2  353.74 265.55  88.20 323.64        3       0
21:22:00  MONGO8S2  353.54 265.38  88.16 323.66        7       0
21:21:00  MONGO8S2  353.25 265.20  88.05 323.66        7       0

20:20:00  MONGO8S2  309.19 230.91  78.28 322.95      177       0
20:19:00  MONGO8S2  307.59 229.54  78.04 322.95      177       0
20:18:00  MONGO8S2  306.66 228.67  77.99 322.95      177       0
20:17:00  MONGO8S2  294.79 219.50  75.29 322.95      177       0
20:16:00  MONGO8S2  260.25 193.63  66.63 323.15      138       0
20:15:00  MONGO8S2  526.03 392.79 133.24 526.49        0       0
20:14:00  MONGO8S2  525.27 392.05 133.21 526.49        0       0
20:13:00  MONGO8S2  533.95 399.55 134.39 526.55        0       0
```

Memory Summary

z/VM Memory map available: ESASTR1

Virtual Machine Memory: ESAUSPG

Linux Memory: ESAUCD2

Linux Memory detailed: ESALNXR

Process Memory: ESALNXP

CMM can be your friend