

Linux on z/VM Configuration Guidelines

Barton@VelocitySoftware.com

[HTTP://VelocitySoftware.com](http://VelocitySoftware.com)

[HTTP://LinuxVM.com](http://LinuxVM.com)

**“If you can’t Measure it,
I am Just Not Interested™”**



Copyright 2008 Velocity Software, Inc. All Rights Reserved. Other products and company names mentioned herein may be trademarks of their respective owners.

Configuring z/VM for Linux on zSeries

- Must configure z/VM – many defaults incorrect **or out of date**
- Linux must be configured for shared resource environment
- Many actions not intuitive
- “Best Practices”

Infrastructure unknowns for “new” installations

- How to manage performance / capacity planning?
- Is chargeback important?
- Operational support for 1,000 servers?
- What are the limits of a configuration and how to measure
- How to share resources to INCREASE the ROI

Measurement and Tuning for z/VM IS Required

- **Start with Proper Configurations**

General Storage Options

Linux Options

- Storage Sizes
- Swapping for Linux
- Linux virtual processors
- Network

z/VM Configuration

- Network, I/O, FTP Topics
- MDC
- Paging and Spooling for z/VM
- DASD/Cache/Channels
- z/VM System parameters

Infrastructure

- Linux infrastructure – monitoring availability and performance

z/VM is shared resource environment

- Over-committing storage improves costs per server
- Over-allocating storage reduces servers that can be supported

Storage requirements of Linux very high

- Linux designed for dedicated storage, references all storage
- **Linux is LRU, competing with VM's reference pattern**
- High percent of referenced pages – what can z/VM page out?

Linux does not drop from queue

- 100 timer pops per second was 1st problem, fixed.
- **Current release of IBM JDK (WAS) polls 10 ms**

6.3 Considerations?

- **More page space....**

z/VM Paging

- Over commitment of storage causes paging
- **Over commitment of storage reduces cost**
- Paging is common **(manageable)** performance problem
- (6.3, paging rates goes up)

Linux Swapping

- Swapping result of over commitment of Linux storage
- Swapping to vdisk very fast, uses storage when it happens
- Swapping to dasd very slow, always noticeable

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
- Swap historically was slow SCSI device so storage 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 - Unlike “Real” memory (pre 6.3)
- Experiment with Linux server swapped 40,000 per second (z800)

Tailoring Linux Storage

Linux data shows
Real storage
Swap storage
“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/      <-----Storage Size (MB)----->
Time/     <--Real Storage--> <-----SWAP Storage--Storage 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
ORA1205a  12062  81.8 11980  4942 4562 380.1 162.6 8115 3702  0
ORA1301b  12062  79.0 11983  4942 4760 181.7 731.4 9952 1299  0
ORA1401a  12062 334.7 11727  4942 4454 487.7 181.5 7234 4312  0
ORA1402a  12062 528.2 11533  4942 3777 1165 133.3 6976 4424  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
WAS2a124  2502.6  14.1 2488  1130 1106 24.0 271.0 264.8 1953 48.0
WAS2a128  2502.6  17.8 2485  1130 1106 23.4 263.1 251.9 1970 48.4
WAS2a402  5016.4  37.7 4979  1130 907.0 222.9 15.8 418.3 4545 0.0
    
```

Reducing virtual storage size may cause swap

- Linux does not swap until out of storage

Swapping to disk

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

Linux Swapping to Vdisk

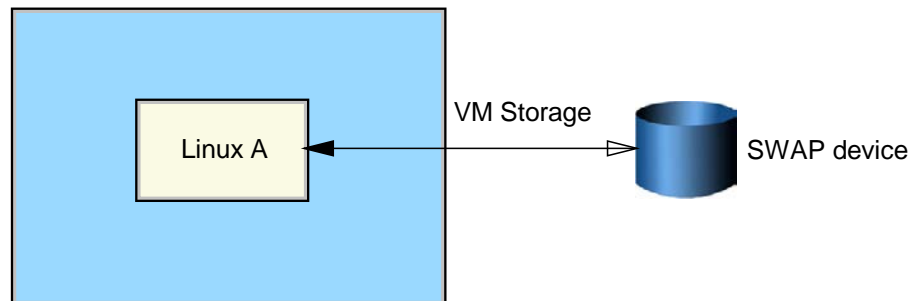
- Not a performance degradation
- 40,000 (z800) / second is FAST

Swap Guideline:

- Define 2 virtual disks, prioritized swap
- Use DIAG driver instead of FBA - Reduces I/O by factor of 8

VM Storage Overview, Paging Hierarchy

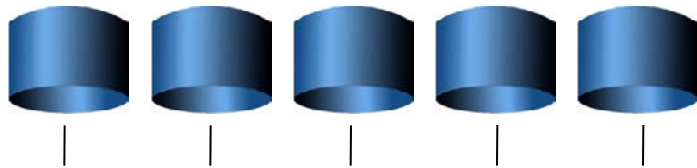
Linux storage/SWAP



z/VM Paging Hierarchy

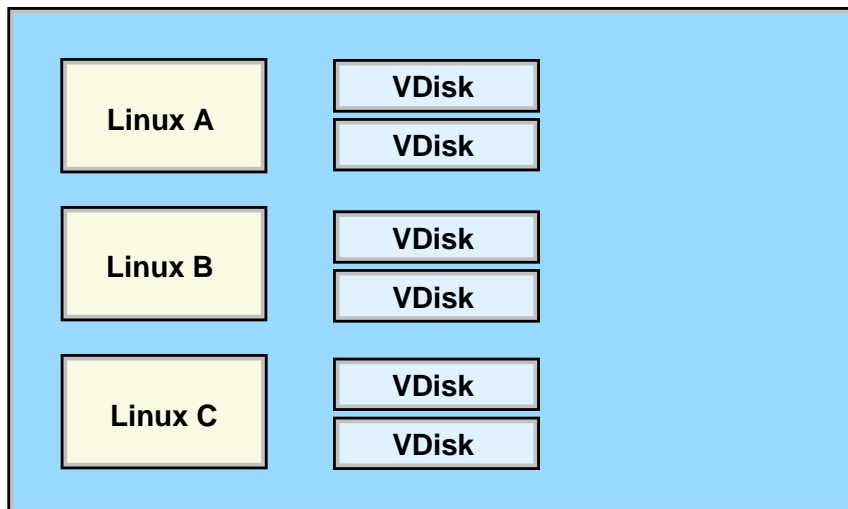
Utilize features of z/VM – Virtual Disk

- Linux not limited in swap rate,
- z/VM supports high paging band width over many exposures



Page volumes

**z/VM Paging bandwidth
VERY HIGH**



Central
processor
storage

**Linux Swap bandwidth
VERY HIGH**

Linux Storage Case Study

First case study:

- Process took hours, system paged significantly
- Reduced size of Linux Virtual Machine, 128mb to 24mb
- Defined 100MB Swap disk
- **Linux reduces storage requirement**
- Process took minutes

Virtual Disk paged out when not in use

- This works!!! Paging greatly reduced, Linux performance greatly improved!!!

This research critical to using Collaborative Memory Mgmt (CMM)

LINUX Swapping to VDISK

Change 128MB Server to 24MB with 100MB Swap Reduction of Overall Storage Requirements of 100MB

- Unused VDISK is paged out

Screen: ESAVDSK Velocity Software, Inc.

Time	Owner	Space Name	<--pages--> Resi- dent	Lock- ed	DASD Page Slots	X- Store Blks
12:15:01	LINUX001	VDISK\$LINUX001\$0202\$0009	36	0	50	0
12:16:01	LINUX001	VDISK\$LINUX001\$0202\$0009	36	0	50	0
12:17:01	LINUX001	VDISK\$LINUX001\$0202\$0009	173	0	50	0
12:18:01	LINUX001	VDISK\$LINUX001\$0202\$0009	293	0	35	0
12:19:01	LINUX001	VDISK\$LINUX001\$0202\$0009	293	0	35	0
...						
12:39:01	LINUX001	VDISK\$LINUX001\$0202\$0009	259	0	35	0
12:40:01	LINUX001	VDISK\$LINUX001\$0202\$0009	259	0	35	0
12:41:01	LINUX001	VDISK\$LINUX001\$0202\$0009	207	0	86	0
12:42:01	LINUX001	VDISK\$LINUX001\$0202\$0009	207	0	86	0
12:43:01	LINUX001	VDISK\$LINUX001\$0202\$0009	13	0	280	0
12:44:01	LINUX001	VDISK\$LINUX001\$0202\$0009	13	0	280	0
12:45:01	LINUX001	VDISK\$LINUX001\$0202\$0009	13	0	280	0

Virtual Storage vs Virtual Disk tradeoffs

Virtual Disk I/O 838K / 900 seconds

- About 900 - 1,000 per second
- (NOTE MDISK HIT RATE!!!!)

Report: ESAUSR3 User Resource Utilization - Part 2 Domino Redbook ESAMAP 3.4.0
Monitor initialized: on 2066 serial 71CE3 First record analyzed: 08/21/03 12:00:00

```
-----  
UserID      DASD MDisk Virt Cache I/O    <---Virtual Device--->  
/Class      I/O   I/O  Hits  Disk  Hit Prty <----I/O Requests---->  
-----  
08/21/03  
12:15:00   613K    0  248K 838K  74.8      0  1510    0  321    0  
**Top User Analysis**  
LINUXA     610K    0  246K 838K  74.8      0    1    0    0    0  
-----  
12:30:00   615K    0  250K 822K  74.6      0  1487    0  324    0  
**Top User Analysis**  
LINUXA     613K    0  248K 822K  74.6      0    0    0    0    0  
-----  
12:45:00   631K    0  260K 884K  75.5      0  1634    0  321    1  
**Top User Analysis**  
LINUXA     628K    0  258K 884K  75.5      0    0    0    0    0  
-----
```

Cost of Swap daemon

- at 1000 swaps per second:
- about 10% (on z800)

Report: ESAHSTA		LINUX HOST Application Report				Domino	Redbook	ESAMAP	
Node/ Date Time	Process/ Application name	<-Application Process Counts----->				<-----Processor-----> <---Utilization--->			
		Total	active	Running	ResWait	Loaded	Percent	seconds	Avg

08/21/03 12:15:00 LINUXA	java	15.0	15.0	2.0	13.0	0	10.3	92.6	0.7
	kswapd	1.0	1.0	0	1.0	0	9.1	82.2	9.1
	router	11.0	11.0	0	11.0	0	10.6	95.4	1.0
	server	67.0	67.0	1.0	63.0	3.0	63.2	568.5	0.9
	snmpd	1.0	1.0	1.0	0	0	3.3	29.3	3.3
	update	3.0	3.0	1.0	2.0	0	10.2	91.7	3.4

12:30:00 LINUXA	java	17.0	17.0	2.0	15.0	0	9.5	85.9	0.6
	kswapd	1.0	1.0	0	1.0	0	8.8	79.5	8.8
	router	12.0	12.0	2.0	9.0	1.0	11.0	99.3	0.9
	server	61.0	61.0	4.0	55.0	2.0	62.7	563.9	1.0
	snmpd	1.0	1.0	1.0	0	0	3.2	28.8	3.2
	update	4.0	4.0	0	4.0	0	12.0	107.8	3.0

12:45:00 LINUXA	java	16.0	16.0	0	16.0	0	10.3	92.4	0.6
	kswapd	1.0	1.0	0	1.0	0	9.5	85.6	9.5
	router	10.0	10.0	0	10.0	0	11.1	99.6	1.1
	server	67.0	67.0	9.0	53.0	5.0	64.3	578.6	1.0
	snmpd	1.0	1.0	1.0	0	0	2.4	21.9	2.4
	update	5.0	5.0	0	5.0	0	13.0	116.9	2.6

VDisk for swap rules:

- Two small virtual disks for swap, prioritized

Breaking the rules increases storage:

Note vdisk large? WHY???

Report: ESASTR1

Monitor initialized: 032094 serial 9E14C

First record analyzed: 03/05/08

```
-----
```

Time	Users <-----Pages		System		User	NSS/DCSS	<-AddSpace>		VDISK
	Loggd	System	<Available>	ExSpc			Resdnt	Resident	
	On	Storage	<2gb	>2gb					
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
03/05/08									
02:15:00	28	1310719	802	4377	1124	967698	2950	230K 10866	229K
02:30:00	28	1310719	784	4635	1123	967458	2952	230K 10866	229K
02:45:00	28	1310719	806	3129	1124	967570	2950	230K 10867	229K
03:00:00	28	1310719	815	3669	1124	967954	2949	229K 10868	228K
03:15:00	28	1310719	874	3991	1124	967710	2953	230K 10868	229K

VDISK Case Study

VDisk for swap best practice: Two small disks, prioritized

- Two disks per server, goodness
- Should be 1 small swap disk, plus 2nd large disks, goodness
- Prioritized backward though, badness....

```
*****
```

Owner	Space Name	<--Size--> AddSpc VDSK Pages Blks	<--pages--> Resi- Lock- dent ed	-----> Stg-> T Migr	DASD Page Slots	X- 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
LINUX6	VDISK\$LINUX6\$\$\$0101\$004B	65791 8738	3.0 0	0	571	0
LINUX6	VDISK\$LINUX6\$\$\$0112\$004C	524K 69905	771 0	0	5666	603
.						
System Totals:		5901K 39321	233K 0	0.7	669K	38631

Additional Storage Performance

Named Saved System

- Fast IPL, shared kernel storage
- Saves 1mb per server, **difficult to implement**

DCSS with XIP File System

- Load all programs into shared DCSS,
- Saves 20-100mb/server, easy to implement
- **Used VERY SELDOM**

CMM: Collaborative memory management

- Dynamically manage storage size
- Saves GB/server, requires feedback
- Used in different forms frequently

How many Virtual Processors?

- Linux is multiprocessor capable
- Global lock is large issue on older Linux
 - One processor acquires lock
 - Other processors attempt to spin
 - On 390 – spin converted to Diagnose 44 (now 9C)
- Problem easily detected
 - High Diagnose -> Instruction Simulation -> SIE
 - High TV ratio
 - **Guideline: Minimize virtual processors**

How many Virtual Processors

Report: ESACPUA

CPU Utilization Analysis

Time	CPU	<CPU percents>			<--Internal (per second)-->			SIGP		
		Totl Util	Ovrhead	Diag	Inst	SIE	Fast		Page	Rate
		Util	Usr	Sys	nose	Sim	intrcp	path	fault	/sec
16:01:00	0	66.6	12	25	80K	82K	83275	2108	0.1	350
	1	67.6	12	25	89K	91K	91879	1051	0	332
	2	62.3	12	24	83K	85K	85768	1219	0.1	383
	3	62.7	11	25	77K	78K	79354	776	0	293
	4	63.6	12	24	84K	85K	86175	1047	0.0	329
	5	63.1	11	26	82K	84K	85064	1188	0.0	297
	6	64.1	11	22	83K	84K	84874	1079	0.0	304
	7	57.3	10	22	73K	75K	75481	1044	0.0	323
	8	62.7	10	26	53K	57K	58761	1421	0.1	267
System:		570	101	218	704K	723K	730630	11K	0.2	2879

- CPU Performance typical of many Linux Apps:
 - High Diagnose 44 -> Instruction Simulation -> SIE
 - z/VM 5.2 modified logic

How many Virtual Processors

Report: ESADIAG

Diagnose Rate Report

Date Count /Time	CPU	<--Total-->		<-----Diagnose							
		<Diags/Sec>		DIAG: Rate		DIAG:Rate		DIAG: Rate		DIAG: Rate	
		User	IBM								
10:45:00	0	0	1954	0000:	0.0	0008:	0.9	000C:	0.1	0024:	0.0
				0068:	0.0	007C:	0	0098:	0	009C:	1733
	1	0	2593	0000:	0.0	0008:	0.9	000C:	0.1	0024:	0.0
				0068:	0.0	007C:	0.0	0098:	0	009C:	2403
	2	0	1891	0000:	0.0	0008:	2.4	000C:	0.2	0024:	0.0
				0068:	0.0	007C:	0	0098:	0	009C:	1654
	3	0	2174	0000:	0.0	0008:	0.6	000C:	0.0	0024:	0.0
				0068:	0.0	007C:	0	0098:	0	009C:	1977
	14	0	1473	0000:	0.0	0008:	0.5	000C:	0.1	0024:	0.0
				0068:	0.0	007C:	0	0098:	0	009C:	1351
System:		0	26540	0000:	0.1	0008:	11.5	000C:	1.1	0024:	0.1
				0068:	0.2	007C:	0.0	0098:	0.0	009C:	24K

- CPU Performance typical of many Linux Apps:
 - High Diagnose 9C -> Instruction Simulation -> SIE
 - Still a problem if too many VCPU

How many Virtual Processors

Report: ESACPUA

CPU Utilization Analysis

```
-----  
      <-----Load----->      <CPU percents><--Internal (per  
      <--Usrs--> Tran          Totl Ovrhead Diag Inst      SIE  
Time    Actv In Q /sec CPU Util  Usr Sys  nose   Sim intrcp  
-----  
10:45:00    65  132  1.7  0  90.7 1.8 2.3 1954 3124 9134.7  
              1  91.7 1.7 2.2 2593 3787 9724.0  
              2  91.4 1.7 2.3 1891 3059 8805.9  
              3  91.9 1.7 1.9 2174 3380 8843.5  
              4  91.9 1.6 1.9 2156 3245 8627.6  
             12  79.5 1.8 2.4 1375 2430 7065.5  
             13  78.9 1.7 2.1 1851 2857 7179.6  
             14  75.1 1.6 2.0 1473 2402 6483.7  
              ----  ---  ---  ----  ----  ----  
  
System:                1285   25   31   27K   43K 116734
```

- CPU overhead much better with Diag9C
 - High Diagnose 9C -> Instruction Simulation -> SIE
 - Still a problem if too VCPU is over configured

FTP Benchmarks: Results NOT intuitive

Compare Linux Asynchronous I/O vs synchronous I/O

- Asynchronous is default
- Synchronous writes data without buffering
- DASD response time
 - Asynchronous: 50ms (6 I/O / second, 512k / IO),
 - Synchronous: 1.5ms (300 I/O / second, 4k / IO)
- Which is better throughput?

Guideline: Use Asynchronous - default

- DASD Response time rot don't work

CP algorithms VERY poor at sizing MDC Storage Control the size of MDC!

Report: ESAMDC Minidisk Cache Analysis . ESAMAP 3.6.1 02/08/07 Pg 2660
 Monitor initialized: 02/07/07 at 00:00:05 on 2084 serial 447AA First record analyzed: 02/07/07 00:00:05

Time	<---Load--->			<IO per><Insertions>						<-----Main Storage MDC-->					<-Expanded Storage MDC----->					<External>				
	<-Users-> Actv	In	Q	<second> /sec	Hit Pct	rds	hits	Usr Max	Per Min	Not Ald	<-Sizes (MB)--> Avg MIN MAX			</Second> Obj Stls		<-Sizes (MB)--> Avg MIN MAX			<Per Second > Obj Rds Wrts		<I/O rate> Stls Pages DASD			
12:20:00	26	18.7	2.2	63	33	20.4	8K	7.5	0	2K	0	8K	2K	0.1	180	1K	0	3K	1K	55	0	0.1	253	261
12:35:00	26	19.1	2.1	63	8.5	5.4	10K	5.8	0	2K	0	8K	2K	0.0	69.9	1K	0	3K	1K	10	0	0.0	53	185
12:50:00	26	18.3	2.0	69	6.0	4.2	11K	4.7	0	1K	0	8K	2K	0.0	43.6	1K	0	3K	1K	12	0	0.0	33	167
13:05:00	27	19.5	2.2	38	29	11.0	12K	5.2	0.4	2K	0	8K	2K	1.2	1062	1K	0	3K	2K	63	0.0	1.3	571	406
13:26:00	31	17.4	1.7	28	28	8.0	14K	12	0.7	4K	0	8K	4K	2.8	1324	272	0	3K	2K	3.7	0.0	4.5	1090	356
13:41:00	25	19.9	2.9	69	60	41.5	14K	7.5	0	3K	0	8K	3K	0.5	483	727	0	3K	2K	2.0	0	0.2	742	422

Guidelines:

SET MDC STORAGE 128M 128M

SET MDC XSTORE 0M 0M | OFF

Overcommitting real storage is good, reduces cost

- Back up is Paging storage

If 40GB main storage

- Overcommit factor of 2 - How much paging storage needed?
- VM installations often very underconfigured
- **Guideline: Paging storage should still be 2 times requirement**

Number of paging devices? Number of channels?

- ROT not valid

Lack of page space planning is top reason for first installation
z/VM outage

As of z/VM 6.3, “pre-write” can fill up page space. ALERT!

Expanded Storage required for paging performance

- True LRU, with page migration
- Page the correct pages
- Page rates to disk drop when converting real storage only system to real+expanded

How much expanded?

- Enough for 30 second window
- **Guideline: Enough so STEAL does not page to disk**
- 20% a good target, usually enough – NO arbitrary LIMITS!!!
- Measure on ESABLKP
- **MEASURE BLOCK SIZE!!!!**

Original problem documented in 1992

- If VTAM has (recommended) REL SHARE 10000, looping user consumed CPU
- If VTAM had ABS 5%, looping user constrained
- Velocity recommended ABS shares for critical servers

Creating EXCESS SHARE

- Setting SHARE to 10000 (compare 100 servers at REL 100)
- Linux servers that are idle, but inqueue servers count
- NOTE: VMRM often used SET REL 10000

Starting with 3 looping users REL 100.

- They all get equal share of the resources
- this is as we expected.

```
Screen: ESAUSP2 Velocity Software-Test VSIVM4 ESAMON 3.778
1 of 3 User Percent Utilization CLASS * USER
<-----Main Storage----->
UserID <Processor> <Resident-> Lock <-WSSize-->
Time /Class Total Virt Total Actv -ed Total Actv
-----
00:11:00 ROBLNX1 32.39 32.38 15862 15862 11 15536 15536
ROBLX2 32.12 32.11 66136 66136 259 78478 78478
ROBLX1 32.02 32.01 38219 38219 176 37790 37790
ROB2LV 0.01 0.00 2246 2246 0 2246 2246
```

We now give ROBLX2 a REL 200

- because that is a more important service
- (nothing with virtual 2-way).
- Not as expected, it gets the excess share

Screen: ESAUSP2 Velocity Software-Test VSIVM4

1 of 3 User Percent Utilization

CLASS * USER

```
<-----Main Storage----->
```

Time	UserID /Class	<Processor> Total	<Resident-> Virt	Lock -ed	<-WSSize--> Total	Actv	Actv	
00:14:00	ROBLX2	68.71	68.68	66211	66211	258	78478	78478
	ROBLX1	14.00	14.00	38245	38245	256	37790	37790
	ROBLNX1	13.99	13.99	15879	15879	11	15536	15536
	ROB2LV	0.01	0.00	2246	2246	0	2246	2246

Now for the experiment

- we reduce the relative share for all idle users down to 1
- (using the allocated share computation below and showing how much allocated / consumed share is).
- This ELIMINATES “EXCESS” bucket

Screen: ESAUSP2 Velocity Software-Test VSIVM4 ESAMON 3.778

1 of 3 User Percent Utilization CLASS * USER

<-----Main Storage----->								
Time	UserID /Class	<Processor> Total	<Resident-> Virt	Lock	<-WSSize--> Total	Actv	Actv	Actv
00:20:00	ROBLX2	48.39	48.37	67141	67141	292	80047	80047
	ROBLNX1	24.19	24.19	16168	16168	11	15536	15536
	ROBLX1	24.19	24.18	39006	39006	241	37790	37790
	ROB2LV	0.01	0.00	2246	2246	0	2246	2246

And when we set ROBLNX1 to REL 300

- it works again: 48% 32% and 16%
- exactly like the REL 300, 200 and 100 we set.

```
Screen: ESAUSP2 Velocity Software-Test VSIVM4 ESAMON 3.778
1 of 3 User Percent Utilization CLASS * USER
<-----Main Storage----->
UserID <Processor> <Resident-> Lock <-WSSize-->
Time /Class Total Virt Total Actv -ed Total Actv
-----
00:23:00 ROBLNX1 48.15 48.14 16170 16170 11 15536 15536
ROBLX2 32.86 32.86 67190 67190 211 80047 80047
ROBLX1 16.44 16.43 39016 39016 193 37790 37790
ROB2LV 0.01 0.00 1680 1680 0 1680 1680
```

SET SHARE

- Use RELATIVE 100 for single virtual CPU
- Use RELATIVE 200 for two virtual CPU
- Use ABSOLUTE for shared or critical resource servers

SET SRM STORBUF – allow overcommit

- SET SRM STORBUF 350 300 300

SET SRM LDUBUF – DO NOT allow overcommit

- You can NOT run paging devices at more than 100% busy!!!
- SET SRM LDUBUF 100 80 60 (or lower)
- Broken in z/VM 6.3, SET SRM LDUBUF 200 200 200

SET QUICKDSP – no function as of z/VM 6.3

- **Use for only absolutely critical servers**

Infrastructure Requirements

Requirements:

- Performance management
- Capacity planning
- Chargeback
- Operations

Shared resource environment:

- Avoid unnecessary work
- Avoid “waking up Linux”

Availability Monitoring – necessary?

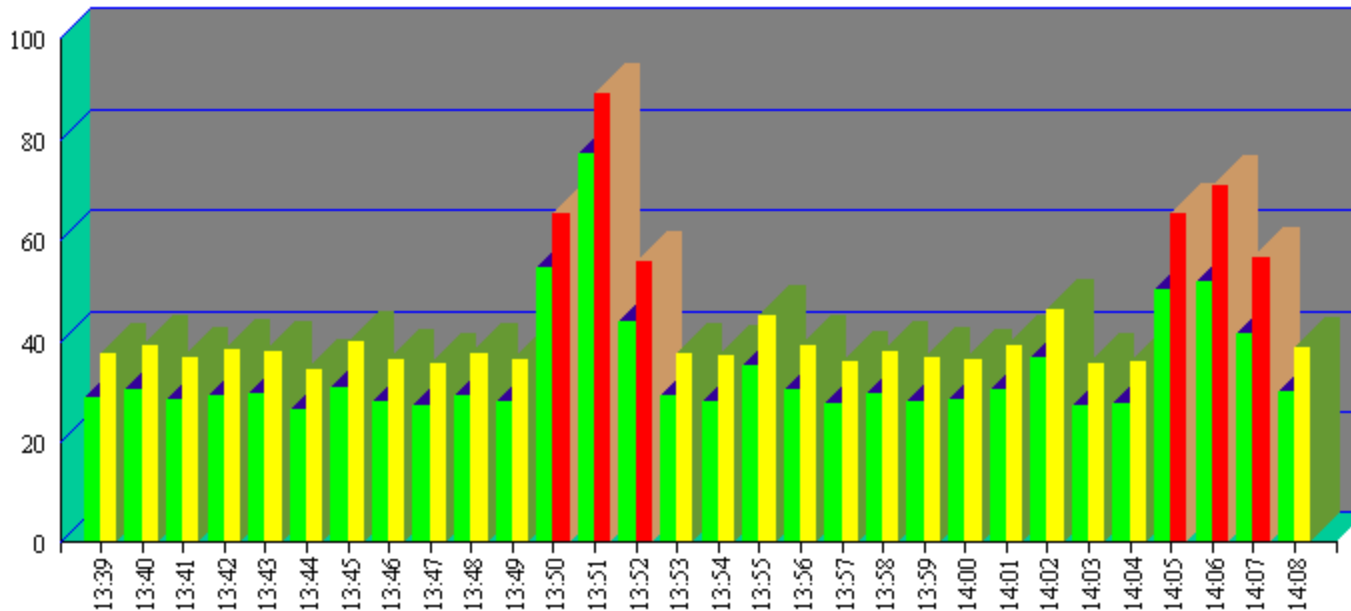
High Availability – cost? (DB2, RAC)

How many different monitoring tools do you need?

Measure your infrastructure and determine scalability!

Infrastructure: SOP Valid?

Virtual and Total Cpu Utilization



Question:

- Why always hit every 15 minutes?

SOP: Standard Operating Procedure???

Detect and alert looping processes

Report: ESAHST1 LINUX HOST Software Analysis Report
Monitor initialized: on 2066 serial 71CE3

```
-----  
Node/      <-----Software Program-----> <CPU Seconds> CPU  StgSize  
Time      Name      ID      Type  Status  Total Intrval Pct  (Bytes)  
-----  
08:32:00  
LINUXA  
init       1      Applic ResWait  0.9    0.0  0.0    61440  
kjournal   95     Applic ResWait  2.5    0.0  0.0     0  
db2fmd     596    Applic ResWait  0.3    0.0  0.0   573440  
sshd      1081   Applic ResWait  0.4    0.0  0.0   204800  
event     10787  Applic ResWait  19.5   0.0  0.0   11188K  
snmpd     10861  Applic Running 193.4  4.2  7.1    1492K  
adminp    11452  Applic ResWait  58.5   0.0  0.1   13848K  
server    11525  Applic ResWait  1.0    0.1  0.1   35720K  
server    11533  Applic ResWait  4.3    0.0  0.0   35720K  
server    11537  Applic Running 44697  58.3  99.2  35720K  
java      13024  Applic ResWait  0.0    0.0  0.0    6632K  
java      24016  Applic ResWait  1.9    0.0  0.0    6632K  
java      24024  Applic ResWait  4.9    0.0  0.0    6632K  
server    24192  Applic ResWait  19.0   0.1  0.1   35720K  
java      26352  Applic ResWait  0.4    0.0  0.0    7320K  
sshd      26477  Applic ResWait  0.2    0.0  0.1    2028K
```

Show process by ID

- Status
- Total CPU
- Percent CPU
- Storage

(Non-velocity mib)

Performance Instrumentation

Performance Instrumentation

- Cost of instrumentation often excessive
- “Native Linux” tools will not detect many problems
- Agents may take 5-10% of a processor (**Per server**)

Cost of instrumentation should be < .1% (of ONE CPU) per server

- **Performance instrumentation should not change performance**

Active agents vs Passive agents

- Active agent wakes up at constant interval and records data
- Passive agent only responds to external request

Linux Configuration Summary

Virtual machine size

- Minimize until some swap

Swapping

- Swap to virtual disk
- Define 2 virtual disks,
 - One to meet the average requirement
 - Second one for overflow - Insurance
- Use DIAG driver instead of FBA
 - Reduces I/O by factor of 8

Virtual processors

- Minimize to meet the workload/application requirement

Infrastructure costs

- Minimize – shared resource architecture

DASD Channels

- ECKD “Measurable” by channel hardware
- FCP/SCSI measurable from inside each linux

Paging

- How much paging is required to support 2 times over commitment of 40GB z/VM system?
- At least 80 GB.

MDC

- Caches data – read-ahead, often used data
- Default too high
- SET MDC STORAGE 128M 128M