

# *Oracle on Linux Measurement & Tuning*

Barton Robinson

**Velocity Software**

Barton@VelocitySoftware.com

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

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

## UKLug 2009



Storage opportunities

Performance measurement and tuning

- Swapping to VDISK analysis
- Storage Analysis and Tuning (XIP, CMM)

XIP

CMM

PLEASE NOTE, Author uses Velocity Software's ESALPS (zVPS) for ALL analysis and demonstrations.



# *Linux Storage: Performance*

Storage is current largest issue and has been for three years.

- Other platforms increase storage to fix performance problems
- “z” objective is to share storage effectively
- Many times, smaller (**virtual**) servers run faster
- Oracle is VERY Virtual Friendly

Current research is in CMM



# *Linux Storage Tuning Summary*

Minimize the virtual machine size

Use VDISK for swap

- Allocate 2 vdisks for swap disks
- Prioritize the disks!
- Change any “real” swap disks to VM paging packs

Use XIP to reduce storage

Watch for VRM (Velocity Resource Manager) to use CMM



## VMRM Does not have feed back or Linux data

- Often crashes Linux servers
- Does not differentiate between Oracle or WAS servers
- Recent experiment: Started databased load, VMRM took more storage away from server

## CMMA

- Project involving kernel, z/VM, microcode
- No validated positive experience (reported crashes)
- Driver withdrawn by Novell in SLES11,
  - (replaced by CMMA-lite)

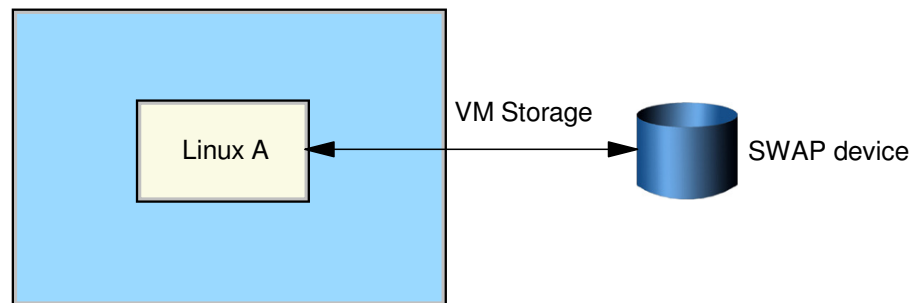
## CMM1

- Very positive results (but can crash a server)
- Requires intelligence, knowledge and feedback



# VM Storage Overview, Paging Hierarchy

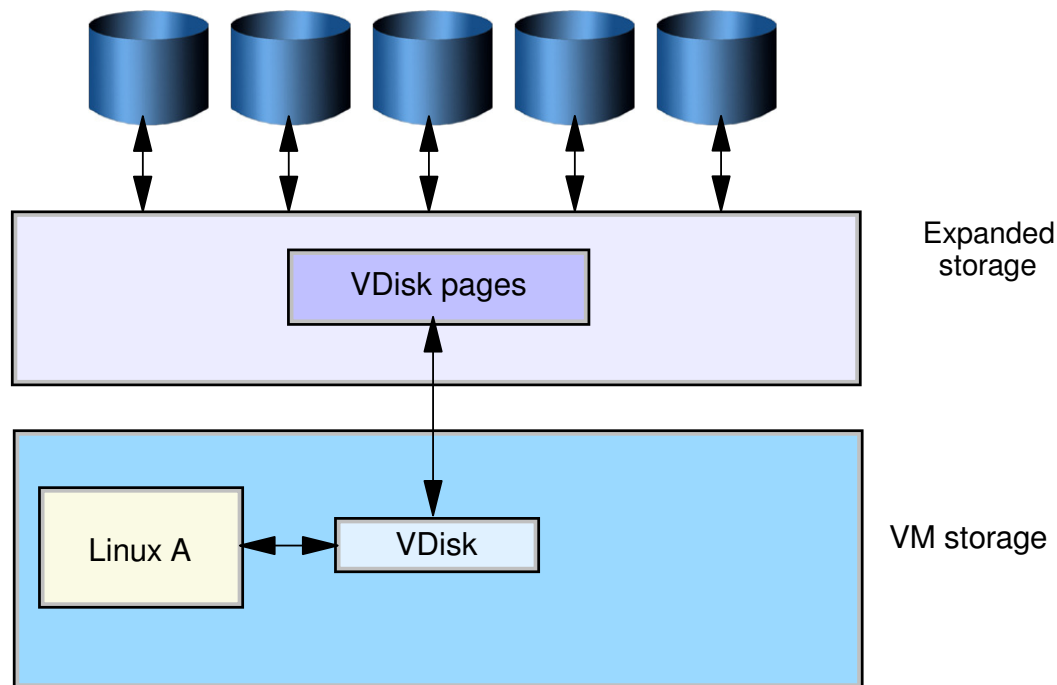
Linux storage/SWAP



# *z/Linux Page / Swap Hierarchy*

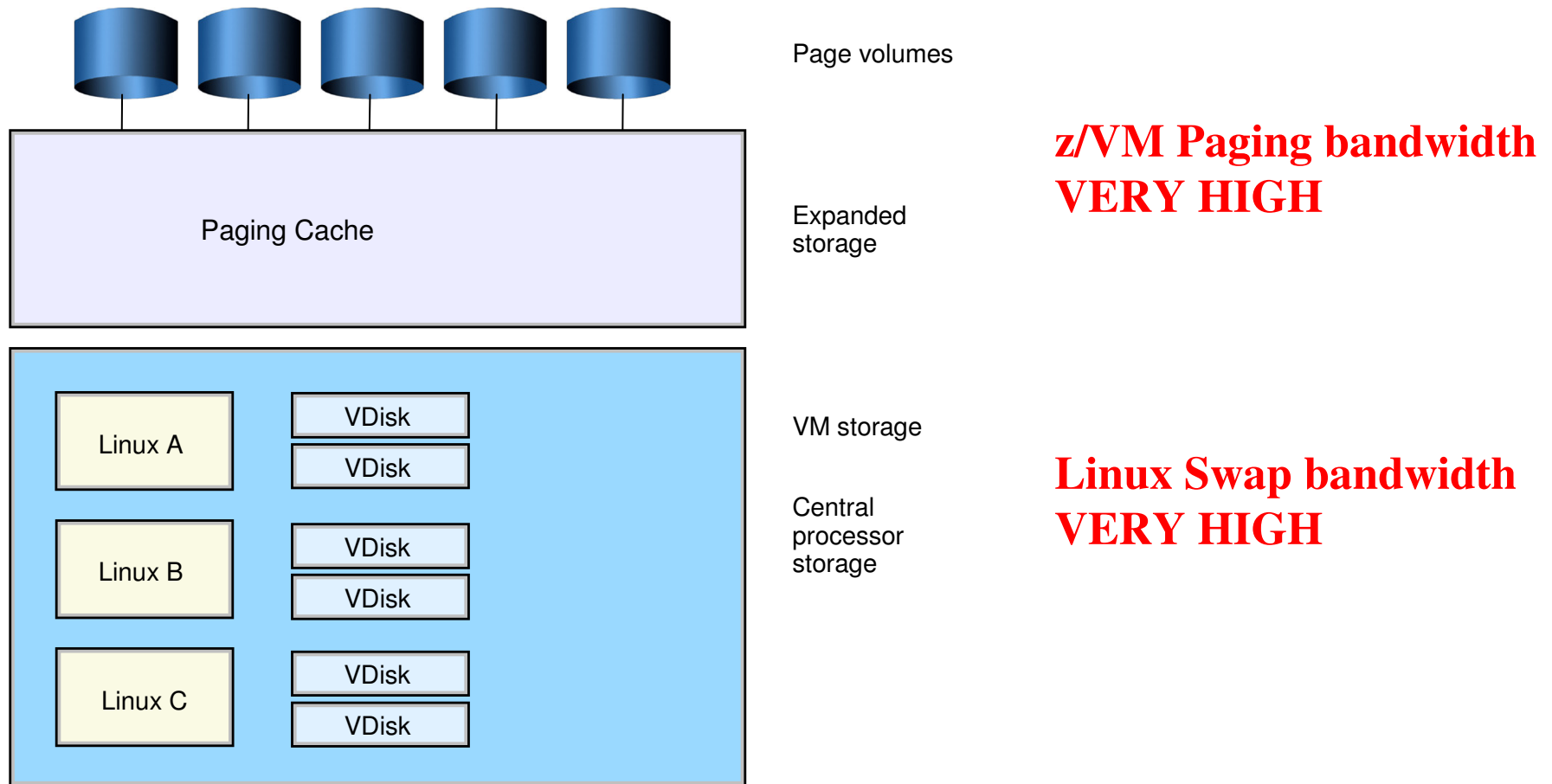
- Utilize features of z/VM – Virtual Disk
- Linux not limited in swap rate,
- z/VM manages storage, high band width

Linux storage SWAP to Vdisk



# z/VM Paging Hierarchy

z/VM paging bandwidth very high, multi-level





# 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.          ESAMON V2.2  03/15 12:14-
<--pages-->  DASD      X-
Resi- Lock-   Page Store
dent   ed   Slots  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
```



# *Tuning Objectives*

## Maximize **SYSTEM** Throughput

- Storage is always a challenge
- Must trade off Linux storage size and VM System Paging
- (or buy LOTS more storage)

## Storage (ram) Opportunities

- Minimize server size (**CMM**)
- Swap to VDISK
- Shared NSS, DCSS (XIP)



# Storage Issues

## z/VM Paging

- Over commitment of storage causes paging
- Database server goes idle, page it out
- **Over commitment of storage reduces cost**
- Paging is common **(manageable)** performance problem

## 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

## Cache

- Linux avoids I/O by using cache
- Linux will cache gigabytes of data if allowed
- I/O is asynchronous, I/O not as bad as swap to disk
- Oracle SGA MUST fit in cache



# Linux Server Sizing

Objective is to improve “SYSTEM” throughput.

- Reducing z/VM paging improves performance of all servers
- A larger server will page more, and be delayed more
- The next Linux page will be the LRU page – and most likely to be paged out by z/VM
  - **(should you give more storage to the slow server?)**

Tradeoffs:

- **z/VM Paging – Affects ALL servers**
- Linux Swapping – Impacts server swapping minimally if Vdisk
- Linux Data I/O – has less impact



# 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

Report: ESAUCD2		LINUX UCD Memory Analysis Report								TEST MAP	
Node/ Time/ Date	-----Storage Sizes (in MegaBytes)----->										
	<---Real Storage--->			<-----SWAP Storage----->				Total	<---Storage in Use-->		
	Total	Avail	Used	Total	Avail	Used	MIN	Avail	Shared	Buffer	Cache
20:58:35											
LNXdap	122.4	4.6	117.8	511.4	501.2	10.2	15.6	505.8	0	17.1	49.6
LNXnfs	193.1	4.6	188.5	511.4	511.0	0.4	15.6	515.6	0	29.6	55.7
LNXzero	122.8	3.4	119.3	444.2	436.1	8.1	15.6	439.5	0	19.6	43.2
LNXdna2	499.6	182.9	316.8	317.3	317.3	0	15.6	500.1	0	25.7	164.5
LNXdna3	499.6	25.0	474.6	511.4	511.4	0	15.6	536.4	0	38.7	315.0
LN Xtux	502.2	6.7	495.5	571.1	571.1	0	15.6	577.8	0	108.9	180.8
LNXPRbt0	499.6	22.9	476.7	511.4	511.4	0	15.6	534.3	0	94.6	241.5
LNXPRbt1	499.6	27.6	472.0	511.4	511.4	0	15.6	539.0	0	25.2	299.9
LNXPRbt2	287.4	18.5	268.9	511.4	511.4	0	15.6	529.9	0	30.7	106.3
LNXPRci1	499.6	10.1	489.5	511.4	358.6	152.9	15.6	368.7	0	20.6	269.4
LNXPRci2	499.6	21.3	478.4	511.4	449.8	61.7	15.6	471.0	0	17.7	164.5
LNXPRot1	499.6	8.5	491.1	511.4	394.6	116.8	15.6	403.1	0	39.0	164.5
LNXPRot3	704.0	12.1	691.8	511.4	511.4	0	15.6	523.6	0	28.9	239.9
LNXPRot5	499.6	4.0	495.6	511.4	451.3	60.1	15.6	455.3	0	4.4	426.5
LNXPRrg1	499.6	15.1	484.5	511.4	431.8	79.6	15.6	446.9	0	22.1	104.1
LNXPRrg2	499.6	24.6	475.0	511.4	465.3	46.1	15.6	489.9	0	23.1	127.1
LNXPRmk1	499.6	24.0	475.6	511.4	453.2	58.2	15.6	477.3	0	8.5	156.1
LNXPRmk2	499.6	27.2	472.4	511.4	465.2	46.3	15.6	492.4	0	13.6	136.3
LNXPRmx1	499.6	36.0	463.6	511.4	465.4	46.0	15.6	501.4	0	14.2	141.6
LNXPRic1	499.6	31.6	468.0	511.4	462.5	48.9	15.6	494.1	0	20.6	184.6
LNXPRic5	248.1	5.6	242.5	511.4	437.8	73.6	15.6	443.4	0	2.4	201.0
LNXPRic6	248.1	5.7	242.4	511.4	467.7	43.7	15.6	473.5	0	2.3	194.8
LNXPRic2	499.6	27.6	472.0	511.4	511.4	0	15.6	539.0	0	38.7	213.9
LNXPRiv1	499.6	16.0	483.6	511.4	316.7	194.7	15.6	332.7	0	2.8	281.7
LNXPRmx1	499.6	29.7	470.0	511.4	511.4	0	15.6	541.1	0	15.3	151.6
LNXPRmx2	499.6	27.8	471.8	511.4	459.2	52.3	15.6	487.0	0	14.6	143.1
LNXPRbq1	499.6	11.6	488.1	511.4	453.2	58.2	15.6	464.8	0	16.3	92.5
LNXPRsd1	499.6	23.7	475.9	1023	1023	0	15.6	1047	0	3.9	411.0
LNXPRkf1	499.6	161.8	337.8	511.4	511.4	0	15.6	673.2	0	22.7	178.9
LNXPRot2	751.1	13.5	737.6	511.4	511.4	0	15.6	524.9	0	47.3	235.8
LNXPRa8	502.3	21.5	480.8	507.7	507.7	0	15.6	529.2	0	18.9	292.3



## Linux

- Kernel – Fixed at boot, includes Page management structures
- Available - small
- Buffer – write buffer, small
- Cache – includes programs, Oracle SGA, data
- Overhead / anonymous – page tables, working storage

## Linux Swap

- Disk
- Virtual disk

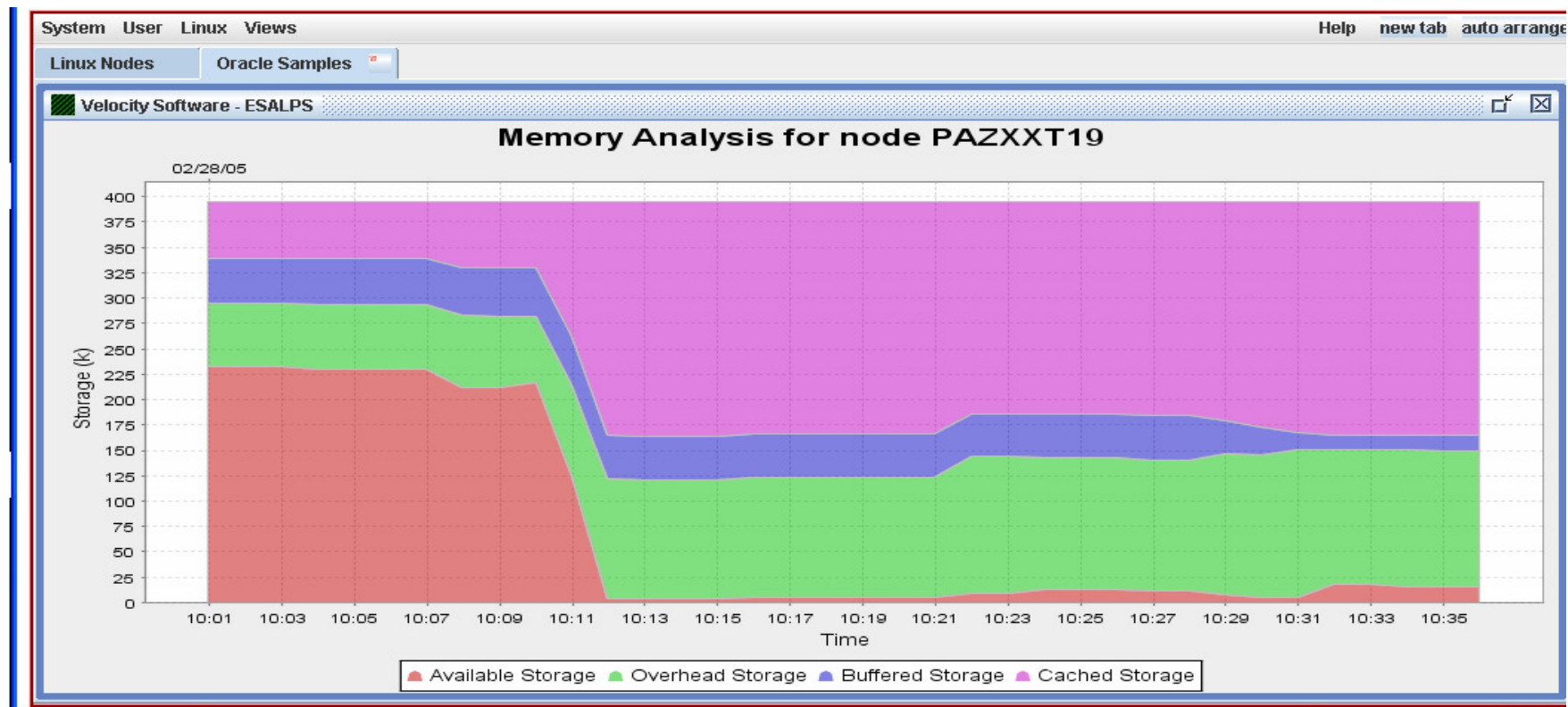
## Opportunities

- XIP/DCSS
- NSS
- Collaborative Memory Management (CMM 1)



# Linux Storage

Following picture shows available storage as oracle starts





# Understanding Linux Storage

## Linux Storage management effective

- Programs loaded once, read/only
- Modified pages become owned by process
- Shared storage (overlap) difficult to analyze
- For oracle processes, difficult to know how much storage is shared

Report: ESALNXC      LINUX Process Configuration Report

Node/ Name	<-Process Ident->			<-----Process----		<Storage(k)>		Proc
	ID	PPID	GRP	Path	parms	Size	RSS	TYPE
PAZXXT19								
init	1	0	0	init	[3]	696	288	4
snmpd	2574	1	2573	snmpd	-a	9788	5652	4
nscd	2867	1	2867	/usr/sbi	"	42928	916	4
oracle	9729	1	9729	ora_dbw0	"	284K	28364	4
oracle	9731	1	9731	ora_lgwr	"	296K	7112	4
<b>oracle</b>	<b>9755</b>	<b>1</b>	<b>9755</b>	<b>ora_j000</b>	<b>"</b>	<b>283K</b>	<b>55964</b>	<b>4 --&gt; largest process</b>
oracle	9761	1	9761	ora_j003	"	282K	17880	4
oracle	13956	1	13956	oracleor	(DESCRIP	282K	22004	4
oracle	13960	1	13960	oracle	"	282K	22072	4



## Costs **(Two xip experiments had higher costs)**

- Addressability – Page management structures
  - Use “mem=” to increase addressability
- DCSS Storage
  - Evaluate resident storage

## Benefits

- Reduced Linux Storage
- Impacts the “page cache” requirement

## Some measurements showed costs can outweigh benefits!

- Increase in kernel size, VM working set size
- Increase in overhead



# Testing Linux Storage “mem=”

## Measure Linux overhead (SLES9/2.6 Kernel)

- Assumption is DCSS in high storage somewhere
- Kernel size (page structure tables)
- Linux reported “total storage” excludes kernel

SUSELNX1 (256mb, 31-bit)

- mem=1gb

SUSELNX2 (256mb, 31-bit)

- mem=256mb

```
Report: ESAUCD2          LINUX EXAMPLE          Linux Test
-----
Node/      <-----Mega Bytes----->
Time/      <--Real Storage--> . <----Storage in Use-
Date       Total   Avail Used   . Shared Buffer Cache
-----
10:26:00   .
SUSELNX1   239.3 175.8 63.5 .      0   18.6 26.1
SUSELNX2   247.3 182.8 64.5 .      0   18.5 26.2
```



# Testing Linux Storage “mem=”

“kernel size” is VM size minus “total”

- SUSELNX1: 16.7mb
- SUSELNX2: 8.7mb
- Linux kernel size increased 8mb (31-bit Linux)

Linux reported “total” size decreases

- Addressability increased from 256 to 1024
- 9mb is accepted cost per GB (for 31-bit linux)
- 31-bit kernel is then 7-8mb

```
Report: ESAUCD2          LINUX EXAMPLE          Linux Test
-----
Node/      <-----Mega Bytes----->
Time/      <--Real Storage--> . <----Storage in Use-
Date       Total  Avail Used  . Shared Buffer Cache
-----
10:26:00   .
SUSELNX1   239.3 175.8 63.5 .      0   18.6 26.1
SUSELNX2   247.3 182.8 64.5 .      0   18.5 26.2
```



# Testing VM's working set "mem=256M"

## Force storage contention,

- only referenced pages resident (low point 1150 pages)

Report: ESAUSR2			User Resource			Linux Test					
	<---CPU time-->			<-Pages-->		<-----Paging (pages)----->					
UserID /Class	<(seconds)>			T:V <Resident>		<---Allocated---		<---I/O---			
	Total	Virt	Rat	Totl	Activ	Total	ExStg	Disk	Read	Write	
-----											
14:47:00											
SUSELNX1	2.48	2.03	1.2	16K	16222	0	0	0	0	0	==> All storage resident
14:48:00											
SUSELNX1	2.47	2.03	1.2	16K	16222	0	0	0	0	0	
14:49:00											
SUSELNX1	3.05	2.28	1.3	5584	5584	11037	907	10130	1021	10145	==> contention starts
14:50:00											
SUSELNX1	4.98	4.15	1.2	1879	1879	14937	21	14916	2916	5033	==> Page stealing
14:51:00											
SUSELNX1	2.68	1.92	1.4	1189	1189	15658	197	15461	2974	2028	
14:52:00											
SUSELNX1	5.63	5.01	1.1	1196	1196	15754	366	15388	3082	2271	
14:53:00											
SUSELNX1	3.89	3.13	1.2	1160	1160	15819	410	15409	2431	1907	====> Minimum storage requirement (SNMP ONLY)
14:54:00											
SUSELNX1	3.33	2.63	1.3	1461	1461	15498	195	15303	143	52	====> init,kblockd,pdflush
14:55:00											
SUSELNX1	3.33	2.67	1.2	1331	1331	15630	362	15268	37	0	==> storage starts to drop
14:56:00											
SUSELNX1	3.70	2.94	1.3	3910	3910	15405	144	15261	2361	0	==> Start "top" - cost 10MB
14:57:00											
SUSELNX1	5.04	4.40	1.1	4135	4135	15056	136	14920	217	0	



# Testing VM's working set "mem=256M"

## Process activity during measurement

Report: ESALNXP LINUX HOST Process Statistics Report

node/ Name	<-Process ID	Ident-> PPID	GRP	<-----CPU Tot	Percents-----> sys	user	syst	usrt	nice	<Stg (k)> Size	RSS
---------------	-----------------	-----------------	-----	------------------	-----------------------	------	------	------	------	-------------------	-----

14:54:00

SUSELNX1	0	0	0	10.0	8.1	1.8	0	0	0	135K	53K
init	1	0	0	0.5	0.4	0.0	0	0	0	628	100
kblockd/	6	4	0	0.1	0.1	0	0	0	0	0	0
pdflush	1509	4	0	0.4	0.4	0	0	0	0	0	0
snmpd	1817	1	1816	4.0	3.0	1.0	0	0	0	7060	4016
slpd	1832	1	1832	0.2	0.2	0.0	0	0	0	3588	1244

==> kernal processes

14:55:00

SUSELNX1	0	0	0	6.3	4.0	2.3	0	0	0	135K	53K
snmpd	1817	1	1816	2.6	1.6	1.0	0	0	0	7060	4016

==> storage requirements drop

14:56:00

SUSELNX1	0	0	0	9.7	6.2	3.5	0	0	0	135K	53K
snmpd	1817	1	1816	3.8	2.4	1.4	0	0	0	7060	4016
sshd	2073	1868	2073	0.3	0.3	0.0	0	0	0	8392	2576

14:57:00

SUSELNX1	0	0	0	13.3	8.8	4.5	0	0	0	137K	54K
kjournal	277	1	1	0.2	0.2	0	0	0	0	0	0
snmpd	1817	1	1816	2.4	1.7	0.8	0	0	0	7060	4016
sshd	2073	1868	2073	0.6	0.4	0.2	0	0	0	8392	2580
bash	2076	2073	2076	0.6	0.5	0.1	0	0	0	3204	1952
top	2095	2076	2095	2.0	1.1	0.9	0	0	0	2132	1100

==> top starts



# Testing “mem=1024M” impact on VM pages

## Force storage contention,

- only referenced pages resident (Low point **2984** pages)

Report: ESAUSR2			User Resource			Linux Test				
-----										
	<---CPU time-->			<-----Ma		<-----Paging (pages)----->				
UserID	<(seconds)> T:V			<Resident>		<---Allocated---			<---I/O---	
/Class	Total	Virt	Rat	Totl	Activ	Total	ExStg	Disk	Read	Write
-----										
16:24:00										
SUSELNX1	2.49	2.03	1.2	16K	15876	0	0	0	0	0 ➡ Unconstrained
16:26:00										
SUSELNX1	2.51	2.08	1.2	10K	10430	5446	5371	75	0	75 ➡ Constrained, page stealing
16:27:00										
SUSELNX1	3.33	2.86	1.2	3725	3725	12481	9137	3344	330	3269
16:28:00										
SUSELNX1	2.29	1.85	1.2	2984	2984	13345	9135	4210	123	866 ➡ Minimum storage requirement
16:29:00										
SUSELNX1	2.27	1.84	1.2	2997	2997	13339	9129	4210	65	0
16:30:00										
SUSELNX1	2.26	1.83	1.2	3109	3109	13339	9129	4210	54	0
16:33:00										
SUSELNX1	2.26	1.83	1.2	3169	3169	13339	9129	4210	0	0
16:35:00										
SUSELNX1	2.35	1.89	1.2	5354	5354	13391	9181	4210	1673	0 ➡ top started - cost
16:36:00										
SUSELNX1	4.02	3.44	1.2	6011	6011	13840	9630	4210	1106	0 ➡ cost 11MB



# Testing “mem=” impact on VM pages

## Increasing “mem=” from 256M to 1024M

- Increased kernel storage 8mb
- Increased VM working set 2000 pages (8mb)
- **NOTE, 64-bit Linux has DOUBLE COST**

## If using XIP in DCSS

- Location of DCSS can create hidden cost
- If dcsc is 1000M to 1024M instead of at 256M,
  - added cost is 8mb REAL storage for 31-bit Linux
  - Added cost is 16mb REAL storage for 64-bit Linux
- If savings from DCSS is 8mb, then no savings using xip
- Page table requirement eliminated in 2.6.26 kernel





## Choose processes for DCSS/XIP

```
Report: ESALNXC          LINUX Process Conf Report
-----
Node/      <-Process  Ident->  <-----Pr  <Storage(k)
Name       ID      PPID    GRP   Path      Size  RSS
-----
SUSELNX1
init        1        0        0  init [3]   628    99
*           4        1        0
khelper     5        4        0  khelper    0      0
kblockd/    6        4        0  kblockd/   0      0
pdflush    1516     4        0    0      0
kjournal    277     1        1  pdflush    0      0
rc          557     1      557  /sbin/sy  2616   144
S14hwsca   1921    557    557  top        2616   256
hwbootsc   1929   1921    557  sshd: ro   2612   208
hwscan     1953   1929    557  -bash      2880  1064
blogd      568     1      568  /sbin/kl   9824    76
syslogd    1798     1   1798  /sbin/kl   1640    728
snmpd      1817     1   1816  /sbin/re   7060  4016 -> candidate
resmgrd    1830     1   1830  /sbin/po   1496    516
portmap    1831     1   1831  /usr/sbi   1516    580
syslogd    1841     1   1841  /sbin/re   1640    196
klogd      1844     1   1844  /sbin/po   1596    224
sshd       1868     1   1868  /sbin/mi   5304   1972 -> candidate
sshd       2073   1868   2073  -bash      8392  2580 -> candidate
bash       2076   2073   2076  top        3204   1952 -> candidate
top        2095   2076   2095  sshd: ro   2132   1104 -> candidate
sshd       6524   1868   6524  -bash      8392  2576
bash       6527   6524   6527  pdflush    3208   1996 -> candidate
bash      10430   6527  10430    3208   1996 -> candidate
vi         10433   6527  10433    4100  2368
```



## Choose processes for DCSS/XIP

- Requires 13mb dcss, located at 64mb
- Measure the impact at process level
  - - saves about 4.7MB virtual storage

Report: ESALNXP      LINUX HOST Process Statistics Report

node/ Name	<-Process Ident->			<-----CPU Percents----->						<Stg (k)>	
	ID	PPID	GRP	Tot	sys	user	syst	usrt	nice	Size	RSS
SUSELNx1	0	0	0	10.6	6.2	4.4	0	0	0	131K	15K
kjournal	279	1	1	0.1	0.1	0	0	0	0	0	0
snmpd	1865	1	1864	1.3	0.7	0.6	0	0	0	7248	1948
sshd	2125	1923	2125	0.6	0.5	0.2	0	0	0	8392	740
bash	2128	2125	2128	0.2	0.2	0.0	0	0	0	3204	592
top	3171	2128	3171	3.3	1.7	1.5	0	0	0	2132	288

→ dropped 2.0 MB  
→ dropped 1.2 MB  
→ dropped 1.4 MB  
→ dropped .7 MB



# Testing “xip” impact on VM pages

Comparable minimal storage: 1160 pages

- XIP reduces to 498 pages – 2.5mb Real saving

```
Report: ESAUSR2      User Resource U      Linux Test
-----
      <---CPU time--> <-----Ma <-----Paging (pages)----->
UserID <(seconds)> T:V <Resident> <---Allocated---> <---I/O--->
/Class Total  Virt Rat  Totl Activ  Total ExStg  Disk  Read Write
SUSELNX1 4.21  3.63 1.2  16K 16186      0      0      0      0      0
16:29:00
SUSELNX1 5.04  4.46 1.1  16K 16186      0      0      0      0      0
16:30:00
SUSELNX1 2.90  2.31 1.3 1290 1290 15046 12465 2581   641 2759
16:31:00
SUSELNX1 5.59  5.04 1.1 1198 1198 15657 4912 10745 3548 10838
16:32:00
SUSELNX1 3.55  2.89 1.2   785   785 16417 1051 15366 2675 6475
16:33:00
SUSELNX1 5.90  5.35 1.1 1111 1111 16548 1206 15342 2547 1813
16:34:00
SUSELNX1 3.26  2.56 1.3   981   981 16667 1342 15325   35   15
16:35:00
SUSELNX1 4.64  3.96 1.2 1402 1402 16505 1232 15273 311   0
16:36:00
SUSELNX1 3.37  2.69 1.3   925   925 17015 1709 15306   89   89
16:37:00
SUSELNX1 4.68  3.99 1.2   738   738 17373 1993 15380 795 678
16:38:00
SUSELNX1 4.63  3.95 1.2   498   498 17639 2342 15297 155  48 --> Minimum storage requirement
```



# Testing “xip” impact on Oracle pages

Two Oracle servers 400mb each (Unconstrained system)

- PAZXXT20 with dcss, dcss 420mb to 512M
- PAZXXT21 without dcss, savings 30MB virtual storage

Report: ESALNXP      LINUX HOST Process Statistics Report

node/ Name	<-Process Ident->			<-----CPU Percents----->						<Stg (k)>	
	ID	PPID	GRP	Tot	sys	user	syst	usrt	nice	Size	RSS
14:42:00											
PAZXXT21	0	0	0	5.1	0.6	0.1	1.2	3.2	0	6M	682K
init	1	0	0	4.3	0	0	1.1	3.2	0	696	288
kswapd0	232	1	1	0.2	0.2	0	0	0	0	0	0
snmpd	2571	1	2570	0.4	0.3	0.1	0	0	0	9784	5644
nscd	2859	1	2859	0.0	0.0	0	0	0	0	43K	912
bash	3258	3257	3258	0.0	0.0	0	0.0	0	0	4788	2360
oracle	3288	1	3288	0.1	0.1	0.0	0	0	0	296K	15K
oracle	3290	1	3290	0.0	0.0	0	0	0	0	281K	15K
oracle	3313	1	3313	0.0	0	0.0	0	0	0	283K	60K
PAZXXT20											
init	1	0	0	11.9	0.0	0	2.6	9.3	0	696	288
pdflush	231	4	0	0.0	0.0	0	0	0	0	0	0
kswapd0	232	1	1	0.0	0.0	0	0	0	0	0	0
snmpd	2575	1	2574	0.2	0.2	0.0	0	0	0	9788	5648
bash	3265	3264	3265	0.0	0	0.0	0	0.0	0	4788	2360
oracle	3295	1	3295	0.2	0.1	0.1	0	0	0	296K	7088
oracle	3320	1	3320	0.0	0	0.0	0	0	0	282K	30K

--> Largest Oracle Process

--> Largest Oracle process



# New “xip=” impact analysis

## Two Oracle servers 400mb each

Report: ESALNXC      LINUX Process Configuration Report

Node/ Name	<-Process ID	Ident-> PPID	<-----Process---> GRP Path	<Storage(k)> Size	Proc RSS	TYPE
---------------	-----------------	-----------------	-------------------------------	----------------------	-------------	------

PAZXXT19

(DCSS)

init	1	0	0	init [3] "	696	288	4
snmpd	2574	1	2573	snmpd -a	9788	5652	4
nscd	2867	1	2867	/usr/sbi "	42928	916	4
oracle	9729	1	9729	ora_dbw0 "	284K	28364	4
oracle	9731	1	9731	ora_lgwr "	296K	7112	4
oracle	9755	1	9755	ora_j000 "	283K	55964	4
oracle	9761	1	9761	ora_j003 "	282K	17880	4
oracle	13956	1	13956	oracleor (DESCRIP	282K	22004	4
oracle	13960	1	13960	oracle "	282K	22072	4

→ largest process

PAZXXT20

(NO DCSS)

init	1	0	0	init [3] "	696	288	4
kswapd0	232	1	1	kswapd0 "	0	0	4
snmpd	2552	1	2551	snmpd -a	9784	5704	4
nscd	2869	1	2869	/usr/sbi "	42928	912	4
oracle	3272	1	3272	ora_dbw0 "	284K	33264	4
oracle	3274	1	3274	ora_lgwr "	296K	14716	4
oracle	3294	1	3294	ora_mmon "	285K	58836	4
oracle	3298	1	3298	ora_j000 "	282K	64696	4

→ Largest Process, (8.5 mb larger)



# Using CMM: Overview

## CMM Overview:

- Requires CMM driver, included with SLES9, SLES10
- Make sure the virtual machine has is enabled for IUCV
  - #CP SET SMSG IUCV

CMM must be loaded prior to use.

- `modprobe cmm sender=VRM`
- Or line in `/etc/zipl.conf` with (followed by doing a ZIPL):
  - `cmm.sender=VRM`
- NOTE: MAKE SURE USERID IS IN CAPITALS

Check to see if loaded:

```
linux9:~ # lsmod
Module                Size  Used by
cmm                   20108  0
msgiucv               13836  1 cmm
iucv                  31032  1 msgiucv
```



# Using CMM: Setting Balloon Size

Command to take away storage from Linux:

- `smsg suselnx2 CMM SHRINK 10000`

Verify it

```
linux9s:~ # cat /proc/sys/vm/cmm_pages
10000
```

Give all the pages back:

- `smsg suselnx2 CMM SHRINK 0000`

Verify it:

```
linux9s:~ # cat /proc/sys/vm/cmm_pages
0
```



# *Using CMM: Setting Balloon Size*

11:39, cmm loaded,  
11:43, take away 10,240 pages (40mb)  
12:38, take away 10,240 pages (40mb)  
12:45, give them back  
12:46, start up memory stresser





# Using CMM: Setting Balloon Size

- Set CMM balloon to 20000, 40000 pages,
- Set CMM balloon to zero pages

Screen: ESAUSR2 Velocity Software, Inc.

3 of 3 User Resource Utilization

<-----Paging (pages)----->						
<---Allocated---> <---I/O--->						
Time	UserID /Class	Total	ExStg	Disk	Read	Write
-----	-----	-----	-----	-----	-----	-----
13:15:00	SUSELNX2	2517	2517	0	0	0
13:00:00	SUSELNX2	2617	2617	0	0	0 (set to zero)
12:45:00	SUSELNX2	1929	1929	0	0	0 (-20000 pages)
12:30:00	SUSELNX2	22845	4160	18685	35937	14443
12:15:00	SUSELNX2	28969	2640	26329	129	0
12:00:00	SUSELNX2	28969	2640	26329	0	0
11:45:00	SUSELNX2	30205	2640	27565	0	0
11:30:00	SUSELNX2	50452	1975	48477	21379	427 (-20000 pages)



# Using CMM: Setting Balloon Size

- Set CMM balloon to 10000 pages,
- Set CMM balloon to zero pages

Screen: ESAUSR2 Velocity Software, Inc.

1 of 3 User Resource Utilization

Time	UserID /Class	<---CPU time-->			<---Main		
		<(seconds)>	T:V	Rat	<Resident>		
-----	-----	Total	Virt	Rat	Total	Activ	
-----	-----	-----	-----	---	-----	-----	
13:15:00	SUSELNX2	44.22	36.73	1.2	77161	77161	
13:00:00	SUSELNX2	276	265	1.0	68721	68721	(zero pages)
12:45:00	SUSELNX2	357	343	1.0	45664	45664	(-40000 pages)
12:30:00	SUSELNX2	250	233	1.1	44758	44758	
12:15:00	SUSELNX2	43.94	36.94	1.2	34877	34877	
12:00:00	SUSELNX2	32.44	25.82	1.3	34791	34791	
11:45:00	SUSELNX2	30.49	23.98	1.3	34774	34774	
11:30:00	SUSELNX2	125	116	1.1	37992	35716	(-20000 pages)



# *Performance Summary*

**Oracle in storage constrained environment**

- **Evaluate working sets**
- **Oracle SGA must fit in page cache**

**Velocity Resource Manager – CMM-1**

**XIP has great potential benefit**

**CMM on SP9 SP4 requires update**

- **PMR: CMM - 10405,7TD,00**



# Performance Resources

⌘ <http://LinuxVM.com>, <http://VelocitySoftware.com>

• **zLinux & z/VM Performance Workshops**

• <http://VelocitySoftware.com/workshop.html>

