

# Linux Storage Measurement and Analysis August, 2015

Barton Robinson,  
Velocity Software  
[Barton@VelocitySoftware.com](mailto:Barton@VelocitySoftware.com)



- **Performance measurement and tuning**
  - The process
  - Understanding Linux Storage
  - Storage Analysis and Tuning
- **Storage opportunities**
- **z/VM 6.3 Topics – Expanded Storage?**
- **Oracle, JVM Storage**
- **NOTE: Reports presented come from zVPS**
  - Linux Data collected by zVPS using snmp
  - Historical data maintained at one minute granularity

# Linux Storage: Performance

- **Storage (ram) is ALWAYS a challenge.**
  - Our real storage is expensive
  - Other platforms increase ram to fix performance problems with inexpensive ram
  - “z” objective is to share storage effectively (**overcommit**)
  - Smaller (**virtual**) servers run faster
- **Current research:**
  - **Linux storage tuning**
  - **Linux system and process storage metrics**
  - **z/VM 6.3**
  - **Application subsystems: Oracle, jvm**

# Linux Storage Tuning Guidelines

- **Distributed admins always ask for too much**
- **Minimize the virtual machine size **Until it starts to swap****
  - **Good for static workload**
- **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 – “old”**
- **Use CMM to reduce storage dynamically**
  - If you take too much, server starts to swap
  - If swap fills up, “bad things happen”
  - Question is feedback and reaction time

# Tuning Objectives

- **Maximize SYSTEM Throughput**
  - HIGH CPU UTILIZATION: (target peak 90-95% on z)
    - CPU (and associated licenses) most expensive resource
    - Objective is to support enough concurrent work to use the CPUs
    - Concurrent work requires storage, storage costs money too
  - **Assembler based applications take a LOT less storage than Java**
  - Must trade off Linux storage size and VM System Paging
  - (or buy LOTS more storage)

# Measurement Objectives

- **Available Metrics:**
  - snmpd – standard with Linux, same as “top”, VERY inexpensive
  - Velocity Software snmp – additional mib supplied by VSI
- **Storage by server**
  - Is the server sized correctly?
- **Storage by process**
  - Are there storage cancers?
  - Is there some processes that use too much?
- **Storage by Application**
  - Chargeback, Capacity planning requirements
- **Where is the opportunity for improvement?**

# The “maturing” Performance Metrics Process

- **Understand storage**
- **Determine available metrics, expose them**
- **Understand and exploit metrics**
- **Evaluate value, accuracy**
  - **(Performance research)**
- **Understand deficiencies and start over**
- **Velocity data sources and naming convention:**
  - “ucd”, “hst”, “tcp” comes from standard linux snmp
  - “lnx” comes from Velocity Software snmp mib
  - “ora”, “jvm” comes from “new” VSI mib
  - Other reports comes from CP Monitor

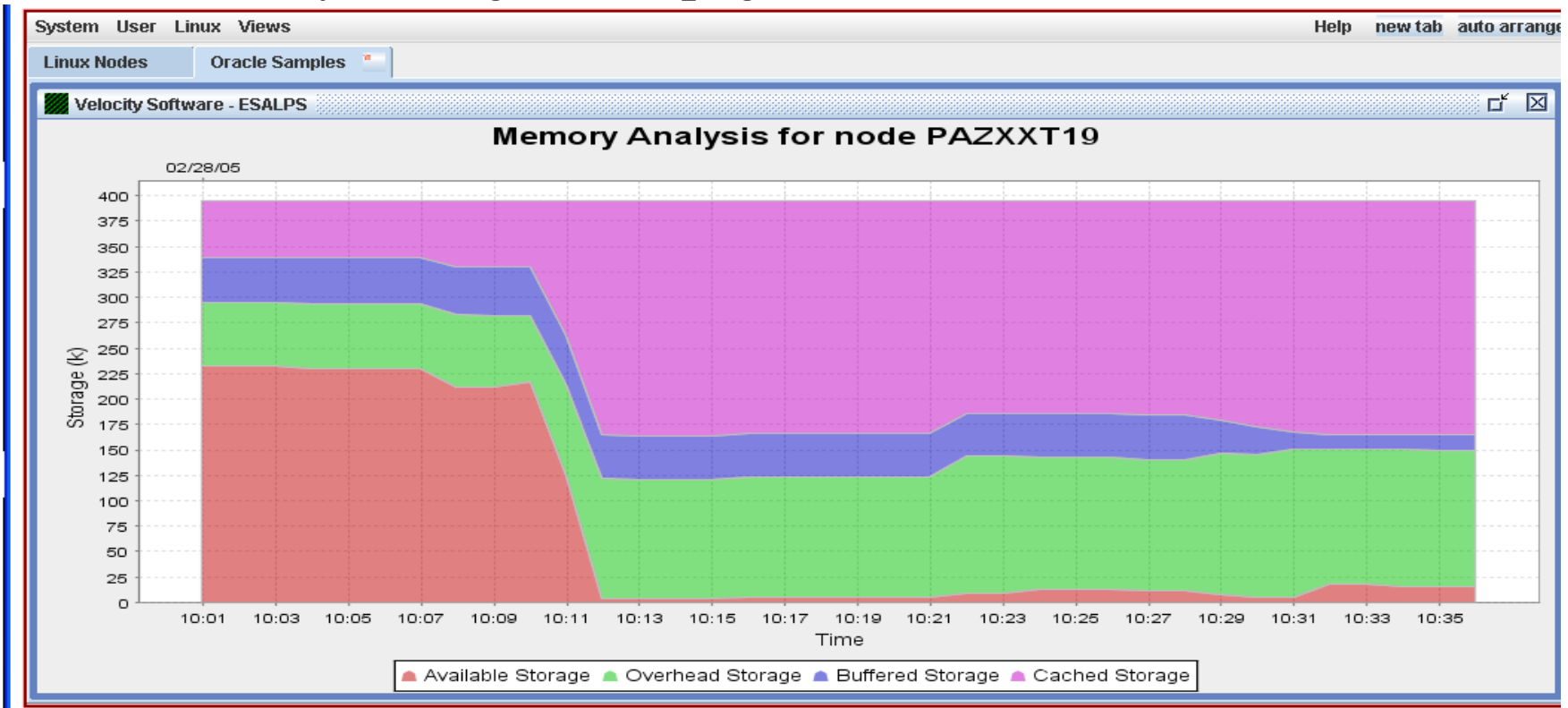
# Linux Storage - Traditional

- **Linux – high level**
  - Kernel – Fixed at boot, includes Page management structures
  - Available - small
  - Buffer – write buffer, small (20mb target?)
  - Cache – includes programs, Oracle SGA, data
  - Overhead / anonymous – page tables, working storage
- **Linux Swap – we want to swap a “little”!**
  - Disk – VERY SLOW
  - Virtual disk – VERY FAST



# Linux Storage at Oracle initialization

- **As Oracle starts:**
  - Available goes down
  - Write buffer contracts
  - Anonymous grows – page tables?



# Storage Structure

- **Linux has page structure tables identifying real pages**
  - Fixed at boot
- **Page tables maintained for each process**
  - 1000's of large processes results in many page tables
  - Page tables require “real storage” (can be LARGE amounts)
- **Process Page table tree**
  - Pgd – page global directory (some entries null)
  - Pmd – page middle directory (some entries null)
  - PTE directory – Page table Entry
  - PGD/PMD collapsed to segment table on “z”
  - PTE directories only needed for referenced pages

# Measuring Linux Storage (UCD mib) – zVPS Report

- **Linux snmp data shows same data as “top”**
  - Real storage, swap storage, buffer, cache
- **Some Swapping is “good”. If not swapping shrink linux server**

**Report: ESAUCD2**                      LINUX UCD Memory Analysis Report                      TEST MAP

```

-----
Node/      <-----Storage Sizes (in MegaBytes)----->
Time/     <--Real Storage--> <-----SWAP Storage----> Total <--Storage in Use-->
Date      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
LN XPRci1 499.6  10.1 489.5 511.4 358.6 152.9 15.6 368.7      0 20.6 269.4
LN XPRci2 499.6  21.3 478.4 511.4 449.8  61.7 15.6 471.0      0 17.7 164.5
LN XPRot1 499.6   8.5 491.1 511.4 394.6 116.8 15.6 403.1      0 39.0 164.5
LN XPRot3 704.0  12.1 691.8 511.4 511.4   0 15.6 523.6      0 28.9 239.9
LN XPRic2 499.6  27.6 472.0 511.4 511.4   0 15.6 539.0      0 38.7 213.9
LN XPRiv1 499.6  16.0 483.6 511.4 316.7 194.7 15.6 332.7      0  2.8 281.7
LN XPRmx1 499.6  29.7 470.0 511.4 511.4   0 15.6 541.1      0 15.3 151.6
LN XPRmx2 499.6  27.8 471.8 511.4 459.2  52.3 15.6 487.0      0 14.6 143.1
LN XPRbq1 499.6  11.6 488.1 511.4 453.2  58.2 15.6 464.8      0 16.3  92.5
LN XPRsd1 499.6  23.7 475.9 1023 1023   0 15.6 1047      0  3.9 411.0
LN XPRkf1 499.6 161.8 337.8 511.4 511.4   0 15.6 673.2      0 22.7 178.9
LN XPRot2 751.1  13.5 737.6 511.4 511.4   0 15.6 524.9      0  47.3 235.8
LN XPRa8  502.3  21.5 480.8 507.7 507.7   0 15.6 529.2      0 18.9 292.3
    
```

# Measuring Linux Storage (UCD mib) – zVPS Report

- Linux snmp data shows same data as “top” (real storage, swap)
- Some Swapping is “good”. If not swapping shrink linux server

Report: ESAUCD2            LINUX UCD Memory Analysis Report  
Monitor initialized: 01/28/15 at 11:18:00 on 2827 se

```
-----  
Node/      <-----Storage Sizes (in  
Time/      <--Real Storage--> <-----SWAP Storage----->  
Date       Total  Avail Used  Total Avail Used  MIN  
-----  
ELNX4000   11931  9131  2799    0    0    0  15.6  
ELNX4002   11931  7140  4791    0    0    0  15.6  
ELNX4003   11931  9147  2784    0    0    0  15.6  
ELNX4013   11931  6344  5587    0    0    0  15.6  
ELNX4032   11931  6353  5578    0    0    0  15.6  
ELNX4033   11931  6365  5566    0    0    0  15.6  
ELNXT006   491.6  50.2  441.4  256.1  240.6  15.5  15.6  
ELNXT025   491.6  60.8  430.7  511.5  497.9  13.6  15.6  
SLNXD197   996.2   8.3  988.0  511.5  510.8   0.7  15.6  
SLNXD199  4015.6 217.8  3798  511.5  510.8   0.7  15.6  
SLNXD201  4015.6 219.6  3796  511.5  510.8   0.7  15.6  
SLNXD217  2503.7 217.9  2286  1151  1151   0.7  15.6  
SLNXD219  2503.7  73.6  2430  1151  1149   2.7  15.6  
SLNXD231  3006.7  92.5  2914  1279  1260  18.9  15.6
```

# Measuring Linux Storage (UCD mib) – zVPS Report

- Linux snmp data shows same data as “top” (buffer, cache, anonymous)
- Some Swapping is “good”. If not swapping shrink linux server

Report:	ESAUCD2		LINUX		Velocity Software Corporate				ZMA
Node/	<-----Bytes)----->								
Time/	<--Real Storage-->			<-----Storage in Use----->				Error	
Date	Total	Avail	Used	CMM	Buffer	Cache	Ovrhd	Shared	Message
EZVC4000	11931	9131	2799	0	378.1	1558	863.8	0	Swap not
EZVC4002	11931	7140	4791	0	412.1	2026	2352	0	Swap not
EZVC4003	11931	9147	2784	0	382.9	1558	842.8	0	Swap not
EZVC4013	11931	6344	5587	0	461.8	4434	690.8	0	Swap not
EZVC4014	5895.1	209.2	5686	0	469.9	3942	1274	0	Swap not
EZVC4015	5895.1	132.4	5763	0	465.0	4014	1284	0	Swap not
EZVC4016	5895.1	152.7	5742	0	463.0	4005	1275	0	Swap not
EZVC4017	5895.1	127.4	5768	0	465.3	4048	1255	0	Swap not
EZVJT006	491.6	50.2	441.4	0	23.7	100.5	317.3	0	
EZVJT025	491.6	60.8	430.7	0	23.3	76.6	330.9	0	
SZVAD197	996.2	8.3	988.0	0	148.2	328.0	511.8	0	
SZVAD199	4015.6	217.8	3798	0	417.7	2609	771.4	0	
SZVAD201	4015.6	219.6	3796	0	251.2	3261	283.3	0	
SZVAD217	2503.7	217.9	2286	0	522.2	1001	762.5	0	
SZVAD219	2503.7	73.6	2430	0	228.3	1076	1125	0	
SZVAD231	3006.7	92.5	2914	0	440.5	838.4	1635	50.5	
SZVAD233	3258.7	205.8	3053	0	467.5	554.6	2031	46.9	

# Measuring Application Storage or other servers

- Node Groups – Virtualized systems and applications
  - Group by ESX

Report: **ESAUCD2** LINUX UCD Memory Analysis Report Velocity Software Corporate

```

-----Storage Sizes (in MegaBytes)-----
Node/      <---Real Storage---> <---SWAP Storage---> Total <---Storage in Use--->
Time/      Total  Avail Used  Total Avail Used  MIN  Avail  CMM  Buffer Cache Ovrhd
Date
-----
00:15:00
***Node Groups***
REDHAT    3477.6 104.4  3373 14609 13490  1119  93.7 13595      0  644.8 1272 1456
SUSE      11098 873.8 10224 25733 24337  1395 193.7 25211      0 1867.6 4367 3989
SOLARIS 2535.1 215.6  2320 1792 888.4 903.6 31.2 720.3      0    0    0 2320
VMWARE 1985.9 135.8  1850 5636 5454 182.7  46.9 5589      0  448.7 779.2 622.1
*** Nodes *****
redhat01  496.9  12.5 484.3  1024  1024    0.1  15.6  1036      0   56.4 338.1  89.8
redhat5   499.2  17.5 481.7  4095  3561 533.2  15.6  3579      0   63.8  24.9 393.0
redhat5x  497.1  18.2 478.9  4095  3874 220.5  15.6  3892      0   96.9  55.7 326.3
linux93   1011.4 21.5 989.9  1914  1336 578.2  15.6  1357      0   61.2  91.4 837.3
redhat56  497.0  17.4 479.7  1176 820.1 355.9  15.6 837.5      0   71.3  40.7 367.6
linux64   2013.1 15.8  1997    0    0    0  15.6  15.8      0   71.4 1819 107.0
rhel55v   498.5  22.5 476.1  2047  2047    0.1  15.6  2070      0 133.0 139.3 203.8
rhel64v   996.4  82.8 913.7  2047  2047    0  15.6  2130      0 194.7 536.2 182.7
redhat6x  994.8  31.0 963.8 125.0 125.0    0  15.6 156.0      0 187.3 625.3 151.2
sles11x   494.2 189.5 304.7  4219  4219    0  15.6  4408      0 139.7 109.7  55.3
sles11    494.7  75.0 419.7  4211  4211    0  15.6  4286      0 141.2 213.5  64.9
sles10    493.0  24.6 468.4  4219  4219    0  15.6  4243      0 141.5 276.7  50.1
SUSEAPPS  997.4  41.6 955.8   62.5  23.8 38.7  15.6  65.4      0 140.4 106.2 709.2
solarisa 1535.6 166.1 1369 768.0 768.0    0  15.6 700.2      0    0    0 1369
solarisb  999.6  49.5 950.1  1024 120.4 903.6 15.6  20.2      0    0    0 950.1
    
```

# Understanding Linux Process 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/          <-Process Ident-> <-----Process----> <Storage(k)> Proc  
Name           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  
oracle       9755    1   9755 ora_j000  "          283K 55964 4 -> largest  
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
```

# Understanding Linux Process Storage

- Linux Storage management (ESALNXP)

- Look at “totals” per node

Report: ESALNXP

```
-----  
node/      <-----CPU Percents-----> <-----  
Name       Tot  sys user syst usrt  Size RSS  
-----  
SLNXD263  0.35 0.18 0.17   0   0 4164  750  
SLNXD265  0.35 0.17 0.18   0   0 4138  743  
SLNXD267  0.35 0.15 0.15 0.02 0.03 4070  808  
SLNXD269  0.62 0.22 0.40   0   0 4189  836  
SLNXD277  0.52 0.23 0.28   0   0 5977 1431
```

Older

Report: ESALNXP

```
-----  
node/      <-----Storage Metrics (MB)----->  
Name       Size RSS Peak Swap Data Stk  EXEC Lib Lck PTbl  
-----  
SLNXD263  4164  750 4325 11.7 2198  4.5  9.0  2K 453 2.70  
SLNXD265  4138  743 4222 3.83 2167  4.5  9.0  2K 453 2.59  
SLNXD267  4070  808 4210 23.1 2228  4.5  9.0  2K 453 2.65  
SLNXD269  4189  836 4333   0 2312  4.6  9.6  2K 453 2.75  
SLNXD277  5977 1431 6042   0 3884  4.6  9.5  2K 517 4.09
```

Newer



# Understanding Linux Process Storage

- Linux Storage management (ESALNXP)

- Look at “interesting processes” per node

Report: ESALNXP

```
-----  
node/      <-----Storage Metrics (MB)----->  
Name       Size RSS Peak Swap Data Stk  EXEC Lib Lck PTbl  
-----  
SLNXD267  4070  808 4210 23.1 2228  4.5  9.0  2K 453 2.65  
events/0   0      0    0    0    0    0    0    0  0  0  
events/1   0      0    0    0    0    0    0    0  0  0  
bdi-defa   0      0    0    0    0    0    0    0  0  0  
tlmagent   43     3  41.7 0.26 34.1  0.1  0.8  6.4  0 0.03  
java       376    46   367 18.4  327  0.1  0.0  38   0 0.19  
java       913   632  903 0.25  807  0.1  0.0  80   0 1.39  
snmpd      129     7   136   0  4.29  0.1  0.0 122   0 0.05  
  
SLNXD277  5977 1431 6042   0 3884  4.6  9.5  2K 517 4.09  
events/0   0      0    0    0    0    0    0    0  0  0  
jbd2/dm-   0      0    0    0    0    0    0    0  0  0  
snmpd      129     6   126   0  4.26  0.1  0.0 122   0 0.05  
cybAgent   662    53   650   0  627  0.1  0.5  18   0 0.18  
java       879   530   860   0  735  0.1  0.0 119   0 1.17  
java       772   296   805   0  634  0.1  0.0 119   0 0.68  
java       546   338   543   0  409  0.1  0.0 119   0 0.78
```

## **XIP: Execute in Place, shared binaries – “old” VMRM Does not have feed back or Linux data**

- History of crashing Linux servers
- Does not differentiate between Oracle or WAS servers (SGA?)
- Takes too much from one server, not enough from another

## **CMMA – too complicated, withdrawn**

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

## **CMM1**

- Very positive results (but can crash a server if used with no feedback)
- Requires intelligence, knowledge and feedback

## **Huge Pages**

# Testing Process 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
-----
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, Measure from z/VM

- only referenced pages resident (low point 1150 pages)

Report: ESAUSR2      User Resource      Linux Test

```

-----
UserID      <---CPU time--> <-Pages--> <-----Paging (pages)----->
/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 requirement
14:54:00
SUSELNX1   3.33  2.63 1.3 1461 1461 15498   195 15303 143 52 ==> init,kblockd,pdf flush
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" + 10MB

```

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

## Process activity during measurement

Report: ESALNXP

LINUX HOST Process Statistics Report

```
-----
node/      <-Process Ident-> <-----CPU Percents-----> <Stg (k)>
Name      ID    PPID   GRP   Tot  sys user syst usrt nice  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 ==> kernal processes
  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
-----
14:55:00
SUSELNX1   0     0     0  6.3  4.0  2.3   0   0   0 135K  53K ==> requirements drop
  snmpd    1817   1  1816  2.6  1.6  1.0   0   0   0 7060 4016
-----
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=” 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 in PTE
  - If dcsc is 1000M to 1024M instead of at 256M,
    - 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

# Implementing XIP

- Choose processes for DCSS/XIP, use “size” for sizing, RSS for “opportunity”

Report: ESALNXC LINUX Process Conf Report

Node/ Name	<-Process ID	Ident-> PPID	<-----Pr GRP	Path	<Storage(k) Size	RSS	
SUSELNX1							
init	1	0	0	init [3]	628	99	
*	4	1	0		0	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	

# Implementing XIP

- Choose processes for DCSS/XIP
  - Impact measured at process level
  - Requires 13mb dcss, located at 64mb
  - Measure the impact at process level
    - - saves about 5.3 MB 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	
kjournald	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	→ dropped 2.0 MB
sshd	2125	1923	2125	0.6	0.5	0.2	0	0	0	8392	740	→ dropped 1.2 MB
bash	2128	2125	2128	0.2	0.2	0.0	0	0	0	3204	592	→ dropped 1.4 MB
top	3171	2128	3171	3.3	1.7	1.5	0	0	0	2132	288	→ dropped .7 MB

```
-----
```



# Testing “xip” impact on z/VM pages

## Comparable minimal storage: 1160 pages

- XIP reduces to 498 pages – 2.5mb Real saving

```
Report: ESAUSR2      User Resource U      Linux Test
-----
```

UserID	<---CPU time-->			<-----Ma		<-----Paging (pages)----->				
	<(seconds)>	T:V	<Resident>	<---Allocated---	<---I/O---	Total	ExStg	Disk	Read	Write
/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

# 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 in /etc/zipl.conf with (followed by doing a mkinitrd,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 20,240 pages (80mb)

12:38, take away 20,240 pages (80mb)

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

Time	UserID /Class	<-----Paging (pages)----->			<---I/O--->		
		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		
		Total	Virt	T:V Rat	Total	Activ <Resident>	
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)

# System Storage Metrics Version 2

- **Linux maturing, providing more metrics**
- **zVPS 4.2 will expose new metrics:**
  - 40 new System Storage Metrics
    - **/proc/meminfo**
  - 10 new process storage metrics
- **z/VM 6.3 – storage is changed**
  - ESASTR3 – Storage IBR Analysis
  - ESAUSTR – User storage
- **Other New metrics – Look at the consumers**
  - Oracle Metrics
  - JVM Metrics

# System Storage metrics (zVPS version 4.2)

Linux now provides 40 new system level metrics, (ESALNXR)

- CACHE, Size, Active, swapped (**Inactive - Not recently referenced** )
- Anonymous (NOT file backed) Active, Inactive

Report: ESALNXR            LINUX RAM/Storage Analysis Rep

```
-----
Node/      <-----Memory in megabytes----->
           <---Cache---><---Anonymous--->
Time       Total Free Size Actv Swap Total Actv Inact
-----
ELNX4000  11931 9131 1558 1081    0 506.1  506  0.0
ELNX4002  11931 7140 2026 1332    0  1918 1918   0.0
ELNX4033  11931 6365 4359 2132    0 324.7  325   0.0
SLNXD263   2001  18.0  681  768  0.7 533.3  253 279.5
SLNXD265   2001  34.7  493  745  0.0 489.0  228 259.4
SLNXD267   2001  127  335  495  0.7 657.6  252 404.5
SLNXD269   3885  51.7 1936 1421    0 659.6  659   0.0
SLNXD277   5895  273 3177 2549    0  1018 1017   0.0
SLNXC107   7911  72.8 7318  788  4.4 189.0  368 4642
SLNXC255   7911  121 6622  639  0.1 787.1 2551 2559
SLNXC257   7911 1524 5130  559  0.4 644.6  786 3979
SLNXC259   8040 1461 5816 1457    0 213.5  184 1598
```



# System Storage metrics (zVPS version 4.2)

Linux now provides 40 new system level metrics, (ESALNXR)

- ActiveFile page cache active
- HugePages
- **PageTable Page tables in use, (4K page table)**
- WriteBack Write operations in progress, should be small or zero - **ALERT?**

Report: ESALNXR Velocity SoftwarMAP

```

-----
Node/      <----- <-Kernel(MB)-> <-Buffers(MB)->
           Stack<-Slab-->                               Page
Time      Total  Size Size SRec Size Dirty Back Tbls
-----
ELNX4000  11931   4.8  284  272  378   0.1   0  4392
ELNX4002  11931  11.2  344  322  412   0.1   0   12K
ELNX4033  11931   4.8  376  358  433   0.0   0  3476
SLNXD263   2001   6.3  87.1  67.9  416   0.1   0   32K
SLNXD265   2001   6.3  190  170  530    0    0   32K
SLNXD267   2001   5.5  214  194  403   0.0   0   32K
SLNXD269   3885   6.7  421  395  572   0.0   0   28K
SLNXD277   5895   8.5  541  509  662   0.1   0   29K
SLNXC107   7911   7.3  95.3  69.9  5.7   0.1   0   47K
SLNXC255   7911  15.1  103  72.2  9.3   0.2   0   78K
SLNXC257   7911  13.0  102  73.0  252   0.1   0   75K
    
```

# HUGE Page Concepts, Support (for “z”)

- **Standard pages are 4k, virtual storage address translated**
  - Segment table / page tables used for “address translation”
- **Translation Lookaside Buffer (TLB)**
  - “relatively small”,
  - bypasses expensive address translation
  - Large page support: 1 TLB entry per mb vs 256 per mb
  - Significant CPU savings at 200 cycles per address translation
- **Huge Page support different on other platforms**
  - 1 mb on system ‘Z’ (or 2gb)
  - 2 mb for Linux on “x”
  - Supported on z/OS, TPF, Linux (not z/VM)

# Huge Page Opportunities (for “z”)

- **Per Martin Schwidefsky (Linux listserv, 11/28/2009)**
  - There will be considerable improvement if the oracle workload uses i) many processes and ii) large shared segments. For each process that maps 1GB of shared memory you save 2MB in page tables.
- **Linux supports Huge Pages**
  - Useful for large databases with multiple connections
  - Fixed, pre-defined, **dedicated storage**, not swappable
  - z/VM still treats pages like 4k pages, so z/VM pageable
- **So where are the savings?**
  - “Real – z/VM” Address Translation still 4k page
  - TLB (hardware level) not impacted (unless dedicated LPAR)
  - **Linux page tables reduced**
  - (almost like old days, VSE mapped “V=R”)

# Validate huge page support in use!!! (ESALNXR)

## First Benchmark showing value of huge pages to pagetables

- Saw no difference in performance???
- (right, huge pages WERE being used in both tests)
- **Value in having data to validate tuning changes**
- **Read the hugepage rules – Oracle will not use hugepages if not enough available**
- **“used Huge Pages” should match SGA size (256mb in this case)**

Node/ Time	<-----Memory in mb>				Page Tbls	<--Huge pages-->		
	Total	Free	Size	Actv		Totl	Free	Rsvd
-----								
06:15:00								
oratst	4015	1188	1749	1027	46K	257	68.0	68.0
oratst	4015	1188	1749	1027	46K	257	68.0	68.0
oratst	4015	1121	1758	1050	48K	257	68.0	68.0
oratst	4015	1112	1760	1057	48K	257	68.0	68.0
-----								
06:25:00								
oratst	4015	1064	1809	1101	48K	257	68.0	68.0
oratst	4015	1023	1810	1107	49K	257	68.0	68.0
oratst	4015	1022	1811	1114	49K	257	68.0	68.0
-----								

# Validate huge page support being used!!!

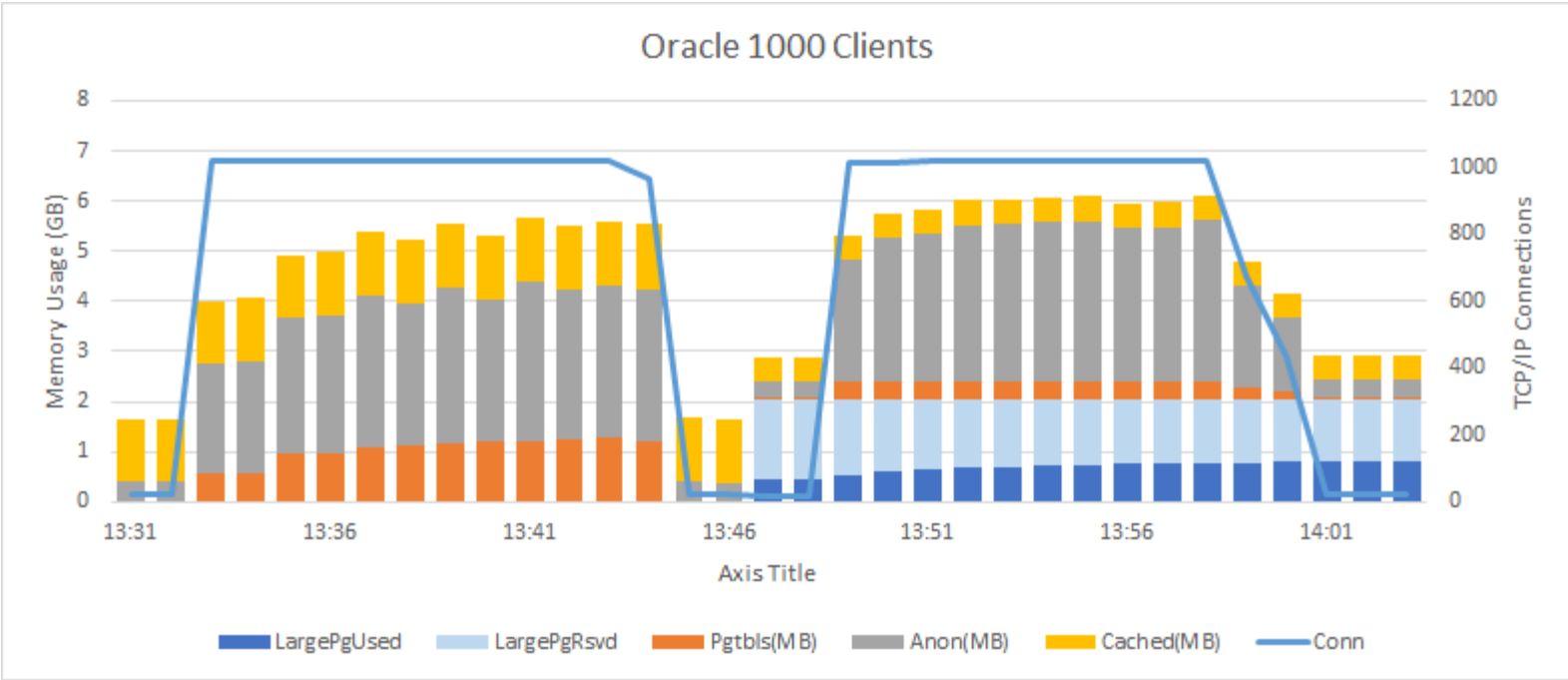
## Second Benchmark showing value of huge pages to pagetables

Report: ESALNXR      LINUX RAM/Storage Analysis Report      Velocity SoftwareZMAP 4.2.0 08/03/14      Pag  
 Monitor initialized: 08/03/14 at 13:00:00 on 2828 serial 414C7      First record aanal4 13:00:00      -

Node/ Time	<-----Memory in megabytes----->					<-Kernel(MB)->					<-Buffers(MB)-->			<Page <-----Huge pages----->					
	Total	Free	Size	Actv	Swap	Total	Actv	Inact	Size	Size	SRec	Size	Dirty	Bacck	Tbls	Totl	Free	Rsvd	Size
roblx1	8040	6084	1245	211	0	367.5	366	771.5	3.5	60.9	35.1	148	0.0	0	28K	0	0	0	1024
roblx1	8040	6083	1245	211	0	367.5	366	771.5	3.5	60.9	35.1	148	0.0	0	28K	0	0	0	1024
roblx1	8040	6084	1245	211	0	367.5	366	771.5	3.5	60.1	35.1	148	0.1	0	28K	0	0	0	1024
<b>roblx1</b>	<b>8040</b>	<b>3573</b>	<b>1246</b>	<b>212</b>	<b>0</b>	<b>2198</b>	<b>2196</b>	<b>771.5</b>	<b>18.3</b>	<b>110</b>	<b>38.8</b>	<b>148</b>	<b>0.7</b>	<b>0</b>	<b>582K</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1024</b>
roblx1	8040	3529	1246	212	0	2228	2227	771.5	18.4	111	39.0	148	0.7	0	595K	0	0	0	1024
roblx1	8040	2649	1250	212	0	2721	2720	775.3	19.1	118	40.6	148	0.0	0	974K	0	0	0	1024
roblx1	8040	2580	1254	212	0	2750	2748	779.3	19.2	120	40.7	148	0.1	0	1.0M	0	0	0	1024
roblx1	8040	2196	1265	212	0	3005	3004	790.9	19.2	120	40.6	148	0.1	0	1.1M	0	0	0	1024
roblx1	8040	2354	1268	212	0	2822	2820	793.6	19.2	122	40.9	149	0.1	0	1.2M	0	0	0	1024
roblx1	8040	2028	1273	212	0	3106	3104	798.0	19.2	121	40.8	149	0.2	0	1.2M	0	0	0	1024
roblx1	8040	2247	1277	212	0	2850	2849	802.4	19.2	120	40.7	149	0.1	0	1.2M	0	0	0	1024
roblx1	8040	1884	1280	212	0	3188	3186	805.5	19.2	120	40.8	149	0.0	0	1.2M	0	0	0	1024
roblx1	8040	2053	1283	212	0	2988	2987	808.2	19.2	121	40.7	149	0.0	0	1.3M	0	0	0	1024
roblx1	8040	1967	1287	212	0	3048	3047	812.1	19.2	120	40.8	149	0.1	0	1.3M	0	0	0	1024
roblx1	8040	2039	1288	212	0	3053	3052	813.3	18.3	119	40.7	149	0.1	0	1.2M	0	0	0	1024
roblx1	8040	6031	1290	212	0	367.9	367	815.5	3.6	62.9	35.4	149	0.0	0	32K	0	0	0	1024
roblx1	8040	6069	1294	212	0	334.2	333	818.9	3.5	62.0	35.3	149	0.1	0	26K	0	0	0	1024
<b>roblx1</b>	<b>8040</b>	<b>4835</b>	<b>483</b>	<b>213</b>	<b>0</b>	<b>341.3</b>	<b>339</b>	<b>0.1</b>	<b>3.5</b>	<b>59.2</b>	<b>34.3</b>	<b>149</b>	<b>8.3</b>	<b>0</b>	<b>16K</b>	<b>2050</b>	<b>1587</b>	<b>1586</b>	<b>1024</b>
roblx1	8040	4839	483	213	0	339.5	337	0.1	3.5	57.9	34.3	149	0.0	0	16K	2050	1587	1586	1024
roblx1	8040	2254	484	213	0	2450	2447	0.1	19.1	113	38.0	149	0.1	0	348K	2050	1509	1508	1024
roblx1	8040	1823	485	213	0	2874	2872	0.1	19.1	117	38.9	149	0.0	0	352K	2050	1423	1422	1024
roblx1	8040	1755	485	213	0	2940	2938	0.1	19.1	118	39.2	149	0.1	0	352K	2050	1400	1399	1024
roblx1	8040	1561	485	213	0	3133	3130	0.1	19.1	119	39.3	149	0.0	0	353K	2050	1361	1360	1024
roblx1	8040	1531	485	213	0	3162	3159	0.1	19.1	119	39.8	149	0.0	0	353K	2050	1345	1344	1024
roblx1	8040	1495	485	213	0	3196	3193	0.1	19.1	120	39.8	149	0.3	0	354K	2050	1314	1313	1024
roblx1	8040	1484	485	213	0	3206	3203	0.1	19.2	120	39.9	149	0.1	0	355K	2050	1303	1302	1024
roblx1	8040	1616	485	213	0	3075	3073	0.1	19.1	119	39.8	149	0.0	0	353K	2050	1291	1290	1024

# Huge Page Opportunities (for “z”)

- **GOOD Measured experiment (8/2014)**



# Process Storage metrics (zVPS version 4.2)

## Benchmark process analysis (2G SGA, oversized)

Report: ESALNXP Velocity Software Corporate ZMAP 4.2.0 08/  
Monitor initiali First record analyzed: 08/03/14 13:00:00

```
-----<br>node/ <-Proc <-----Storage Metrics (MB)-----><br>Name ID Size RSS Peak Swap Data Stk EXEC Lib Lck PTbl<br>-----</pre>
```

### (WITHOUT HUGE PAGES)

oracle	43146	2303	265	2249	0	3.07	0.1	181	13	0	<b>0.96</b>
oracle	43148	2310	81	2256	0	8.95	0.1	181	13	0	1.06
oracle	43152	2303	57	2249	0	3.07	0.1	181	13	0	0.69
oracle	43158	2308	141	2254	0	3.20	0.3	181	14	0	1.21

### (WITH HUGE PAGES)

oracle	51439	2304	18	2250	0	4.26	0.1	181	14	0	<b>0.31</b>
oracle	51441	2303	16	2249	0	3.07	0.1	181	13	0	0.30
oracle	51443	2303	16	2249	0	3.07	0.1	181	13	0	0.30

# Application Feature – Oracle Metrics (SGA/PGA)

Oracle Reports: (10.2, 11.2, 12C supported) (zVPS 4.2)

ESAORAG: General Storage Areas (SGA, PGA)

Report: ESAORAG                      Oracle SGA/PGA Analysis Report                      Velocity S

```

-----
Node/          <-----Shared Global Area (SGA)----->
Date    <--DataBase ----> Fixed Redo Buffer    <-----Pool sizes----->
Time    Name      Instance Size Buffr Cache Free Shrd Large Java Stream ShrIO
-----
18:30:00
oracle  orcl      orcl      133K  135K    0    0 1543K  268K 268K    0    0
s11r20ra db01     db01     143K  274K 1711K    0 1812K  268K 268K 268416  0
    
```

```

<-----PGA Data----->
<-Target-> InUse Alloc Free
Parm Auto          -able
-----
1778K 90752 1239K    0 881K
1007K 1832K    0 4544 1074K
    
```



# Application Feature - Java/WebSphere Metrics

## Java/WebSphere Storage (zVPS 4.2)

<b>Report: ESAJVM</b>	Java Subsystem Analysis Report							Velocity Sof		
-----										
Node/	<JavaClass>		Memory	<-----Heap data----->						
Date	<-----Application----->		<--Loaded-->	pending	<-----sizes----->					
Time	Name	Type	Curr	/Sec	Final	Init	Used	Commit	Max	
-----										
13:06:00										
S11R20RA	WAS Server1	JVM	15287	0	0	52.4M	100M	107.5M	268M	
	WAS Server2longerna	JVM	15312	0	0	52.4M	85.4M	103.3M	268M	

# Analyzing Oracle Storage

## Oracle 12 has higher dependency on Java, see red paper

Report: ESALNXP            LINUX HOST Process Statistics Report

```
-----  
node/      <-Process Ident-> Nice PRTY <-----CPU Percents----->  
Name       ID      PPID   GRP  Valu Valu  Tot  sys user syst usrt  
-----  
08:45:00  
PAZXXT10   0       0      0    0    0 55.9 7.47 46.1 0.77 1.56  
init       1       1      1    0    20 2.30 0.00  0 0.74 1.55  
ora_vktm   2940    1     2940  0    -2 1.29 0.72 0.57  0  0  
Xvnc       3508    1     3501  0    20 0.86 0.31 0.56  0  0  
snmpd      15678   14338 15678 -10   10 0.56 0.31 0.24  0  0  
java      17178 17162 17160  0  20 27.7 1.71 26.0  0  0  
oracle_1   17261   1     17261  0    20 1.02 0.15 0.87  0  0  
oracle_1   17263   1     17263  0    20 1.07 0.16 0.91  0  0  
oracle_1   17265   1     17265  0    20 0.88 0.14 0.75  0  0
```

## 6.3 Thoughts

Page space utilization IS higher than previous

- Many pages duplicated in real storage AND on disk
- **FULLY FUNCTIONAL ALERTS NEEDED**

Customers with Expanded Storage can reconfigure

- Prior, most large customers have found that 20% **(really)** of Storage configured as expanded works best
- LRU requirement because Linux (java) apps poll and don't drop from queue
- NEW 6.3 storage algorithm is much closer to LRU than previous "steal" algorithm

# Analyzing z/VM 6.3 Storage

## z/VM 6.3 has IBR (Invalid But Resident) to replace ExStore

### New Report - ESAUSTR

Report: ESAUSTR User Storage Analysis

```

-----
UserID      <-----Virtual Server Storage (Pages)-----> <Resident>
/Class      Size  Alloc Resi- UFO  <--IBR--> <AgeList> <Unreferd>
            -----> dent Activ TOT  >2gb <2gb >2gb <2gb >2gb
            -----> -----> -----> -----> -----> ----->
15:29:00    265M  169M  134M  133M  1756  1029  217  1.2M  142  331K

***User Class Analysis***
Servers      117K  18119  2060  454.0  287  256  0  1319  0  120
ZVPS        49152  5409  2305  6.0  18.0  18.0  0  2281  0  13.0
Linux       14336  1980  96  3.0  6.0  6.0  0  87.0  0  0
TheUsers    265M  169M  134M  133M  1240  618  217  1.2M  142  331K

***Top User Analysis***
LNXJT007    52.4M  39.2M  23M  22.1M  16.0  16.0  57.0  499K  0  241K
LNXJT009    52.4M  43.4M  36M  35.6M  166  20.0  0  27K  0  2220
LNXJT010    52.4M  27.2M  24M  23.6M  188  50.0  90.0  160K  89.0  55K
LNXJT013    52.4M  36.5M  30M  30.2M  212  92.0  1.0  175K  0  8116
LNXJT011    52.4M  22.1M  22M  21.6M  14.0  14.0  53.0  167K  53.0  15K
LNXJT017    786K  116K  19796  260.0  128  45.0  1.0  19K  0  2978
LNXCT002    524K  304K  53250  1471  10.0  10.0  8.0  52K  0  2765
    
```

# 6.3 (bad) Case Study

Customer backed of 6.3 because of “very bad performance”

## System Storage analysis

Report: ESASTR1 Mais Velocity Software Corpora  
 Monitor initialized: 075 on 2828 serial 1 First record analyzed: 07

Time	Users		-----MegaBytes-----				User Resdnt	NSS/DCSS Resident	<-AddSpace>	
	Loggd On	System Storage	Frame Table	<Available> <2gb >2gb					System	User
12:40:00	72	10240	80	0	0	9637	10	316	1	
12:41:00	72	10240	80	0	0	9636	10	316	1	
12:42:00	72	10240	80	0	0	9638	10	315	1	
12:43:00	72	10240	80	1	2	9639	10	311	1	
12:44:00	72	10240	80	2	11	9622	10	318	1	

## User Analysis

Report: ESAUSTR User Storage Analysis V  
 Monitor initialized: 07/22/15 at 12:38:55 on 2828 serial 1F567 F

UserID /Class	<-----Virtual Server Storage (MB)----->		<Resident> Mbyt				
	Size	Alloc	Resident	UFO	<AgeList>	<Unreferd>	
			Activ	TOT	<2gb >2gb	<2gb >2gb	
12:40:00	6556	25148	9637	623.1	8816	<b>1836 6979</b>	43.3 155 <b>1855</b> 7092
DB2-PROD	128.0	12494	5943	142.1	5736	1190 4545	17.1 47.9 1197 4593
DB2-COPY	128.0	10971	3493	349.9	3070	642 2429	13.8 59.6 655 2488

# 6.3 Thoughts

## VDISK Allocation:

- Vdisks are not backed until “touched”
- Define many vdisks, and no overhead until “touched”
- Multiple vdisks for swap, small to large, prioritized!

## Paging Devices

- There will be some vendor hardware differences
- One installation going to SSD because 6.3 died....

# Storage Conclusions?

Storage metrics getting better, changing

Storage algorithms are changing

Tuning guidelines changing

Research in progress

Or , just use lots of money to buy enough storage

Lots of opportunity to measure and manage storage

ROI should be high