Linux Storage Measurement and Analysis

Barton Robinson,
Velocity Software
Barton@VelocitySoftware.com

VELOCITY SOFTWARE

PROVEN PERFORMANCE

Topics

- **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 ALLWAYS 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, mq**

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**
- **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

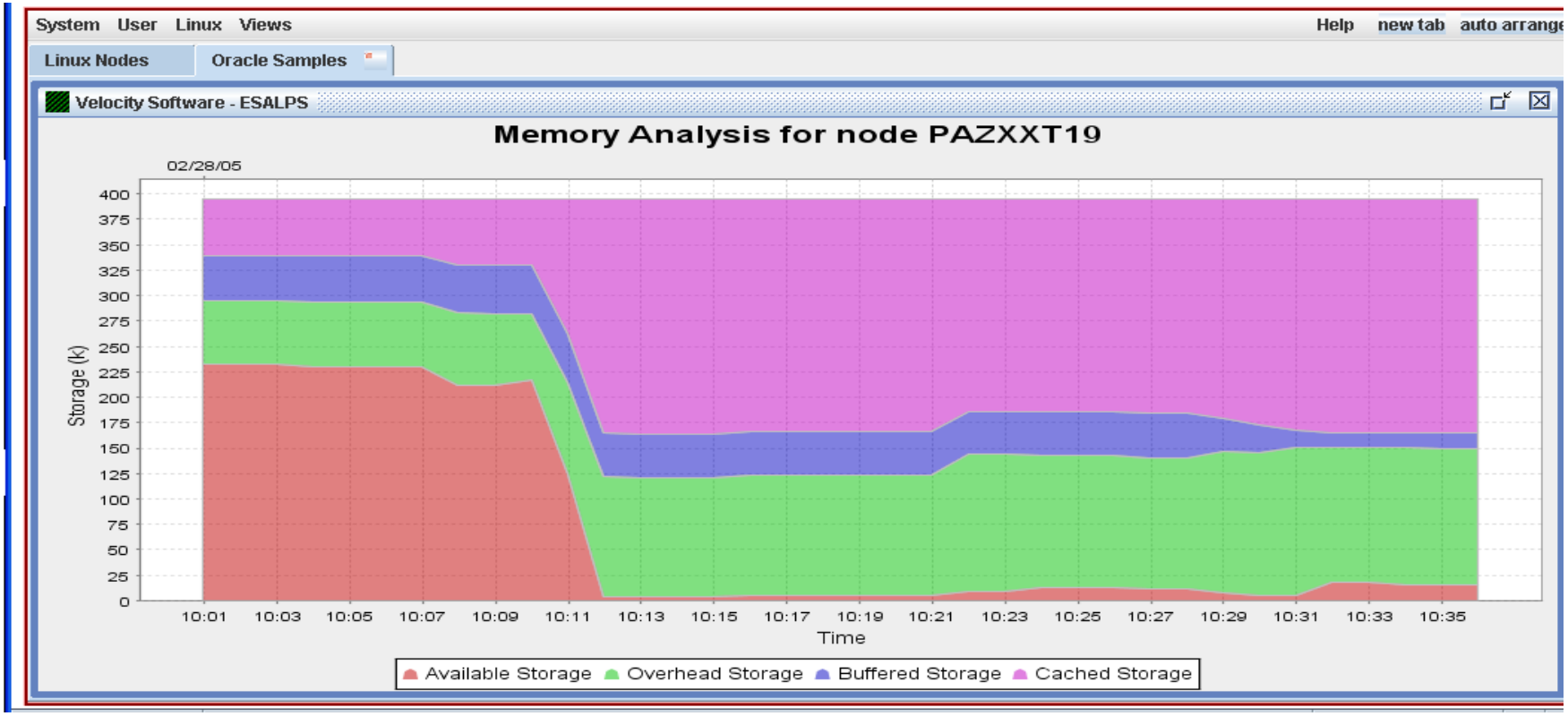
- **Understand storage**
- **Determine available metrics, expose them**
- **Understand and exploit metrics**
- **Evaluate value, accuracy**
  - **(Performance research)**
- **Understand deficiencies and start over**
- **Velocity data sources and naming convention:**
  - “ucd”, “hst”, “tcp” comes from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

Linux Storage

- **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



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											
LNXldap	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
LNXtux	502.2	6.7	495.5	571.1	571.1	0	15.6	577.8	**0**	**108.9**	180.8
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
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

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
-------------------------------------------------------------------------------
Node/    <-------------------------Storage Sizes (in MegaBytes)------------------->
Time/    <--Real Storage--> <-----SWAP Storage---> Total <----Storage in Use----->
Date     Total  Avail Used  Total Avail Used  MIN   Avail CMM    Buffer Cache Ovrhd
-------- ----- ----- ----- ----- ----- ----- ----- ------ ------ ----- ----- ------
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
```

Opportunities

XIP: Execute in Place, shared binaries

VMRM Does not have feed back or Linux data

- Often crashes 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

Measuring Tuning Impacts: XIP in DCSS

- **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 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
-------------------------------------------------------------------
         <---CPU time--> <-Pages-->  <------Paging (pages)------->
UserID   <(seconds)> T:V <Resident>  <---Allocated---> <---I/O--->
/Class   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
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/     <-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  ==➔ storage requirements drop
 snmpd     1817     1  1816  2.6  1.6  1.0    0    0    0 7060 4016
-------------------------------------------------------------------
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=” impact on VM pages

- **Increasing “mem=“ from 256M to 1024M**
  - Increased kernal storage 8mb
  - Increased VM working set 2000 pages (8mb)
  - **NOTE, 64-bit Linux has DOUBLE COST**
  - **(Page table entries: 32 bytes per page for 31 bit, 64 bytes for 64 bit)**
- **If using XIP in DCSS**
  - Location of DCSS can create hidden cost in PTE
  - If dcss 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/        <-Process Ident-> <-----Pr <Storage(k)
 Name        ID    PPID   GRP  Path      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/     <-Process Ident-> <-----CPU Percents----------->  <Stg (k)>
 Name      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   → 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
-------------------------------------------------------------------
         <---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
```

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
smsgiucv               13836  1 cmm
iucv                   31032  1 smsgiucv
```

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
                  <------Paging (pages)------->
          UserID  <---Allocated---> <---I/O--->
Time      /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
                    <---CPU time--> <---Main
          UserID    <(seconds)> T:V <Resident>
Time      /Class    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)
```

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)

- SwapCached Both in swap and ram
- Active Recently referenced
- **Inactive Not recently referenced**
- ActiveAnon Anonymous storage NOT file backed
- **InactiveAnon**
- 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      LINUX RAM/Storage Analysis Report            Velocity SoftwaZMAP 4.2.0 08/03/14   Page
Monitor initialized: 08/03/14 at 13:00:00 on 2828 serial 414C7    First record an4 13:00:00
--------------------------------------------------------------------------------------------------------------
Node/    <--------Memory in megabytes-------------> <-Kernel(MB)-> <-Buffers(MB)->
                  <---Cache----><---Anonymous---> Stack<-Slab-->           Write Page <----Huge pages---->
Time     Total Free Size Actv Swap Total Actv Inact Size Size SRec Size Dirty Back Tbls Totl Free Rsvd Size
-------- ----- ---- ---- ---- ---- ----- ---- ----- ---- ---- ---- ---- ----- ---- ---- ---- ---- ---- ----
13:33:00
roblx1    8040 3573 1246  212    0  2198 2196 771.5 18.3  110 38.8  148   0.7    0 582K    0    0    0 1024
--------------------------------------------------------------------------------------------------------------
```

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
  - One clock cycle vs “30” clock cycles for 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” Address Translation still 4k page
  - TLB not impacted (unless run dedicated LPAR)
  - **Linux page tables reduced**
  - (almost like old days, VSE mapped “V=R”

Validate huge page support being used!!!

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/     <------Memory in mb>
                     <---Cache Page <--Huge pages->
Time      Total Free Size Actv Tbls 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
-----------------------------------------------
```

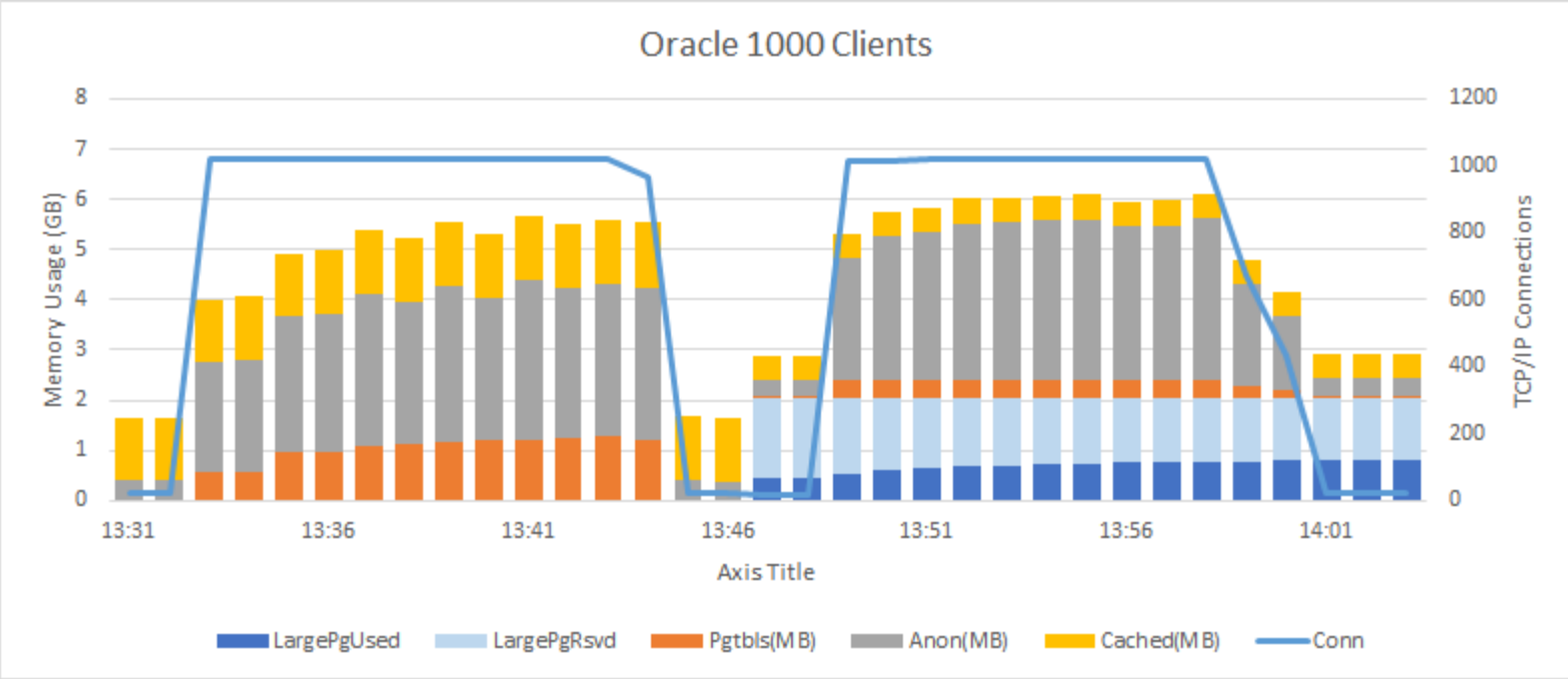
Validate huge page support being used!!!

Second Benchmark showing value of huge pages to pagetables

```
Report: ESALNXR      LINUX RAM/Storage Analysis Report            Velocity SoftwwareZMAP 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/    <--------Memory in megabytes------------> <-Kernel(MB)-> <-Buffers(MB)-->
                  <---Cache----><---Anonymous---> Stack<-Slab-->                 <Page <----Huge pages----
Time     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
roblx1    8040 3573 1246  212    0  2198 2196 771.5 18.3  110 38.8  148   0.7     0 582K    0    0    0 1024
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
roblx1    8040 4835  483  213    0 341.3  339   0.1  3.5 59.2 34.3  149   8.3     0  16K 2050 1587 1586 1024
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)

New metrics at Process Level

- RSS, Size - Same
- Locked: Locked memory size (mlock)
- Peak: peak RSS (high water mark)
- Data: size of data, stack
- Stack: size of stack
- EXEC: size of executable (text)
- Lib: shared library code size
- **PTBL: page table entries (linux 2.6.10) - Use to evaluate HUGE PAGES**
- **Swap: Swapped out**

```
Report: ESALNXP      LINUX HOST Process Statistics Report          Velocity Software Corporate   ZMAP 4.2.0 08/03/
Monitor initialized: 08/03/14 at 13:00:00 on 2828 serial 414C7     First record analyzed: 08/03/14 13:00:00
-------------------------------------------------------------------------------------------------------------------
node/     <-Process Ident-> Nice PRTY <------CPU Percents----> <---------------Storage Metrics (MB)----------> <-F
 Name      ID    PPID   GRP  Valu Valu  Tot  sys user syst usrt  Size RSS Peak Swap Data Stk  EXEC Lib Lck PTbl min
--------- ----- ----- ----- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- --- --- ---- ---
13:33:00
roblx1        0     0     0    0    0 24.7 2.58 19.8 1.34 0.95 844K  11K 825K    0 2013 55.9  66K  5K 8.6  239 43K
 nscd      2767     1  2767    0   20 0.05 0.05    0    0    0  115    1  113    0 14.2  0.1  0.1 2.1   0 0.03   0
 bash      3182  3174  3182    0   20 0.02    0 0.02    0    0    6    3  6.3    0 1.58  0.1  0.6 2.2   0 0.02   1
 tnslsnr   3424     1  3424    0   20 3.50 1.50 0.90 0.95 0.15   62   17 60.2    0 8.00  0.1  0.7  49   0 0.10  2K
 perl      3864     1  3864    0   20 0.52    0 0.02 0.07 0.42   18   14 17.9    0 12.0  0.1  1.4 2.7   0 0.04   8
```

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
-------------------------------------------------------------------
node/     <-Proc <---------------Storage Metrics (MB)---------->
 Name     ID      Size RSS Peak Swap Data Stk  EXEC Lib Lck PTbl
--------- -----  ---- ---- ---- ---- ---- ---- ---- --- --- ----
 oracle   43146  2303  265 2249    0 3.07  0.1  181  13   0 0.96
 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
 oracle   43160  2303  101 2249    0 3.07  0.1  181  13   0 0.84
 oracle   43190  2318  101 2263    0 3.07  0.2  181  13   0 1.04
 oracle   43193  2303   28 2249    0 3.07  0.1  181  13   0 0.44
 oracle   43229  2308  108 2254    0 3.20  0.2  181  14   0 1.04
 oracle   43231  2308  117 2254    0 3.20  0.2  181  13   0 1.20
 oracle   43235  2306  124 2252    0 3.20  0.2  181  13   0 0.96
 oracle   43271  2308  100 2256    0 3.20  0.2  181  14   0 1.10
 oracle   45550  2303   23 2249    0 3.07  0.1  181  13   0 0.43


 oracle   51439  2304   18 2250    0 4.26  0.1  181  14   0 0.31
 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
 oracle   51447  2303   16 2249    0 3.07  0.1  181  13   0 0.31
 oracle   51451  2303   22 2250    0 3.07  0.1  181  14   0 0.32
 oracle   51453  2314   23 2259    0 3.07  0.1  181  13   0 0.32
 oracle   51455  2303   16 2249    0 3.07  0.1  181  13   0 0.31
 oracle   51457  2310   23 2256    0 8.95  0.1  181  13   0 0.31
 oracle   51459  2318   17 2263    0 3.07  0.1  181  13   0 0.32
```

Tuning options

Linux controls starting to make sense

Control pdflush (to write dirty pages to disk)

- **/proc/sys/vm/dirty_ratio** (default 40) (percent cache)
- **/proc/sys/vm/dirty_background_ratio** (default 10): (pct active)

Swappiness

- Zero avoids swapping for cache space, applications not swapped
- 100 favors making disk cache larger
- Large disk cache, larger amount of dirty pages

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
Monitor initialized: 08/06/13 at 18:28:15 on 2096 serial 34B42     First reco
-------------------------------------------------------------------------------
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)

```
Report: ESAJVM       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

See the new Velocity Software newsletter!

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
-------------------------------------------------------------------
          <-------Virtual Server Storage (Pages)----> <Resident>
UserID    Size  Alloc Resi- UFO   <--IBR--> <AgeList> <Unreferd>
/Class                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 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 growing
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