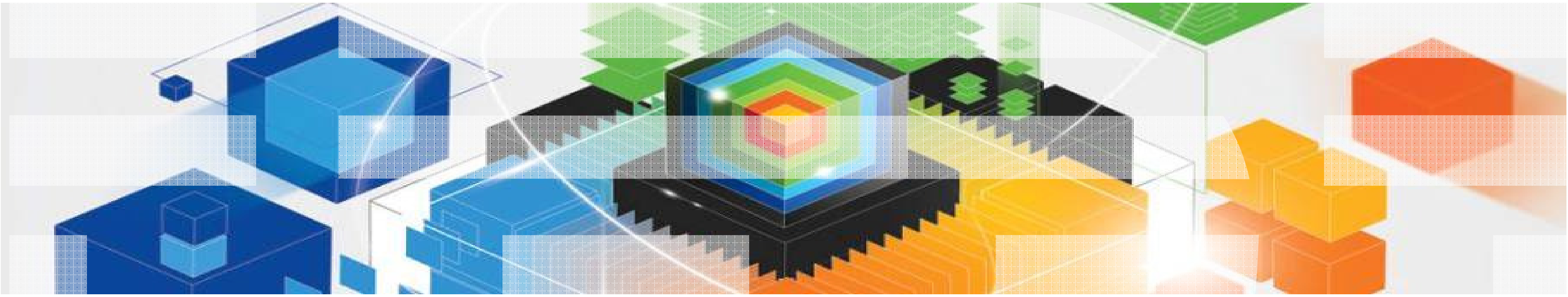
Oracle on IBM z Systems CPU Reduction Tips
April 30, 2015

Mike Zechman - Database Consultant, Convertible Technology, Inc.

David Simpson - Oracle Technical Specialist, IBM (simpson.dave@us.ibm.com)



Copyright and Trademark Information

- For IBM – can be found at http://www.ibm.com/legal/us/en/copytrade.shtml
- For Oracle – can be found at

  http://www.oracle.com/us/legal/index.html

- Any performance results/observations in this presentation are purely for education and planning purposes. No Test results should be construed as indicative of any particular customer workload or benchmark result.

Introduction

- **Goal: To share what we’ve learned in our pursuit to reduce CPU cycle usage over the last 4 – 5 years**
  - Beneficial for Virtual Environment
  - Shares CPU and Memory
  - Can be used in any environment
  - My Customers - Service Provider(s)
  - History of why and how we got to this point
  - Last several years CPU Reduction Sessions on SIG Site
  - This session includes some items of previous years, but mostly includes what we have reviewed this past year

Agenda

- Why reduce CPU cycle usage?
- Targeting CPU Usage and Statistics
- CPU Resource Consumption Tips
- CPU Updates (IBM z13, Oracle 12c , Oracle Enterprise Manager)
- Comments/Questions

Why Reduce CPU Usage ?

- Goal: To share what we’ve learned in our pursuit to reduce CPU cycle usage
  - Money, saving, and not wasting $’s - SAVE MONEY!
  - Helps Performance
  - Helps Service Provider (data center)
  - Helps Customers
  - z/VM manages SUSE Linux Virtual Servers

Service Provider

- Reduce the number of required CPU's
- CPU Physical Cost factor
- Hardware costs
- Software cost/licensing
- Smaller Footprint in Data Center
- COST – Per CPU Potential hundred thousand plus per year
- $ per CPU includes overhead for Service Providers

Customer

- Customer billed by CPU cycle usage
- Customers share Software cost
- Reduce CPU Usage for databases and applications
- Keeps you honest and system tuned
- Some apps are not honest!
- Some are!
- Wide variety of simple and complex solutions

Targeting CPU Usage

- Make sure Oracle database is tuned/goals!
- Requires regular database monitoring.
- Monitor Processes at the Server level
- Target Processes at Server level to reduce CPU usage on the server
- Target Processes at the Database level to reduce CPU

Process CPU Usage

- Gather INDIVIDUAL PROCESS CPU usage statistics (all processes)
- Variety of tools to gather statistics
- Some Free, some not
- We use Velocity software at the z/VM level for process CPU
- Keep CPU usage for each process at 5 minute intervals
- Summarize Top 5 or 10 CPU usage processes by each Server for an hour, every hour, for each day.
- Remainder of process - OTHER
- Load Statistics into Oracle Tables
- Created standard SQL’s to view statistics

Process Reports

- 11g SELECT PIVOT works great. (SUM, GROUP BY, ROLLUP)
- Target high CPU usage processes
- Surprised at what we FOUND!
- Compare production systems
- Compare test systems
- Compare OS and Database Upgrades

Some Processes

- Oracle was biggest consumer most of the time on Oracle Guests
- Other processes combined were anywhere between 20 to 90% of the CPU usage on a server
- Test Servers – Oracle was less
- Production Servers – Oracle was more
- Administrative Tasks Use CPU Cycles
- ZIP, GZIP, TAR, RMAN, TAR, Perl, JAVA, BPBKAR, auditd

CPU Resource Consumption Tips

- Oracle was biggest consumer
- Other processes combined were anywhere between 20 to 90% of the CPU usage on a server
- Test Servers – Oracle was less
- Production Servers – Oracle was more
- Administrative Tasks Use CPU Cycles

Reduce Linux RPM’s that are Installed For ORACLE Guests

- Helps reduces the Disk space & the Number of Linux services created.
- Reduces the software updates/bug/security updates that are required.
- Use the Oracle RPM checker
  - Requirements for Installing Oracle Database 12c on RHEL 6 on IBM: Linux on System z (s390x) (Doc ID 1574413.1)
  - Requirements for Installing Oracle Database 12c on SLES 11 on IBM: Linux on System z (s390x) (Doc ID 1574414.1)

- Oracle 12c database no longer requires the 31-bit s390 libraries
  - Oracle client still requires 31-bit libraries (not typically installed on DB server)

Oracle Grid Control Agent

- Oracle 10.2.0.5 monitoring agent is consuming high cpu

- Per Oracle, Oracle’s new 12c Agent consumes less CPU.

- 12c Release 4

- Per Oracle, 12.1.0.4 Agent must less CPU resource intensive

Oracle RMAN Backup Compression

Backup Compression	Backup Time	Compression Size Source DB - 1.29 GB	% Compression / Input MB/s
‘Basic’ 10gR2 (**BZIP2**) Compression	02:48 (168 s)	278.95 MB	78.9 % 7.89 MB/s
‘High’ 11gR2 (**BZIP2**) Compression	08:41 (521 s)	224.82 MB	83.0 % 2.54 MB/s
‘Medium’ (**ZLIB**) Compression	01:08 (68 s)	295.53 MB	77.6 % 19.46 MB/s
‘Low’ (**LZO**) Compression	00:28 (28 s)	357.03 MB	73.0 % 47.26 MB/s

- **RMAN Command -> CONFIGURE COMPRESSION ALGORITHM ‘Low‘**
- **Oracle Advanced Compression Feature required for Low, Medium, High**
- **Very High CPU observed with BZIP2**

Oracle's Backup's

- Biggest Challenge is to use same strategy for all platforms while reducing CPU
- Originally went to disk, then Veritas offload to Tape

Currently Being Tested

- RMAN Differential Incremental's (0 and 1)
- Oracle Block Change Tracking (BCT) – Without it full scan
- Archive REDO Logs
- Direct to NET BACKUP To DATA DOMAIN
- DATA DOMAIN DISK (Compression)
- Allows Mirroring of Backups on Disk

Oracle’s VKTM Process

- Oracle’s VKTM timer service centralizes time tracking and offloads multiple timer calls from other clients.

- **VKTM** is responsible for providing a wall-clock time and reference-time counter (updated every 20ms) **even when the database is idle for a long time (CPU Idle).**

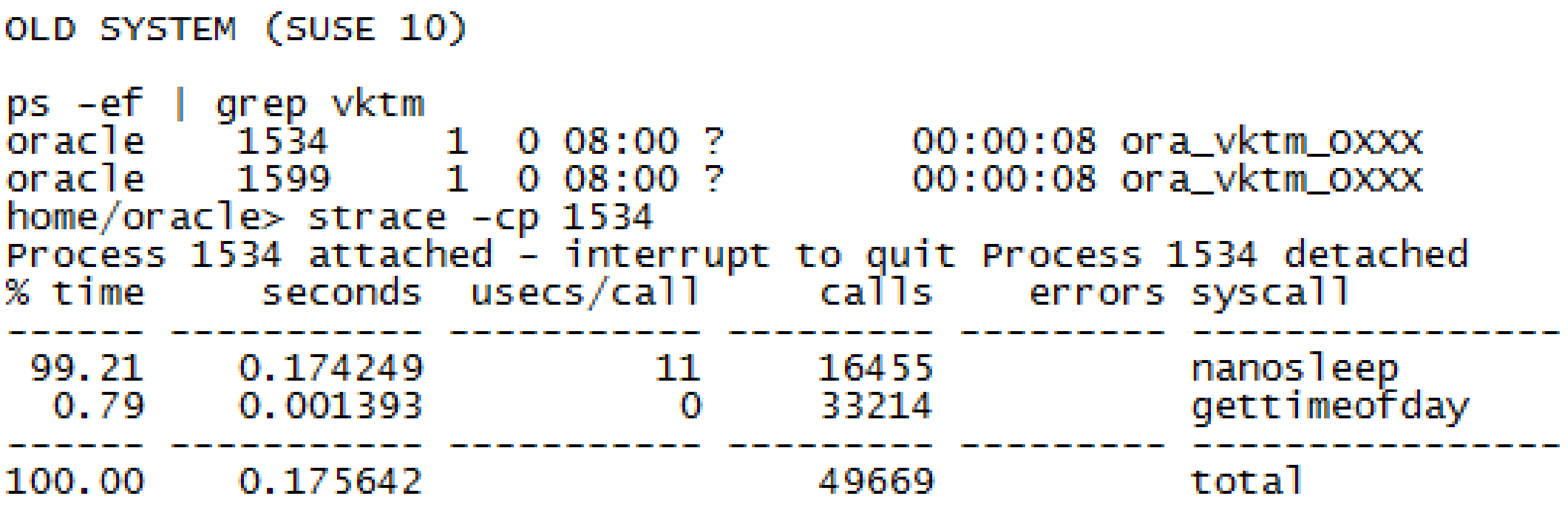
SUSE 10

kernel timer interrupt frequency is approx. 100 Hz

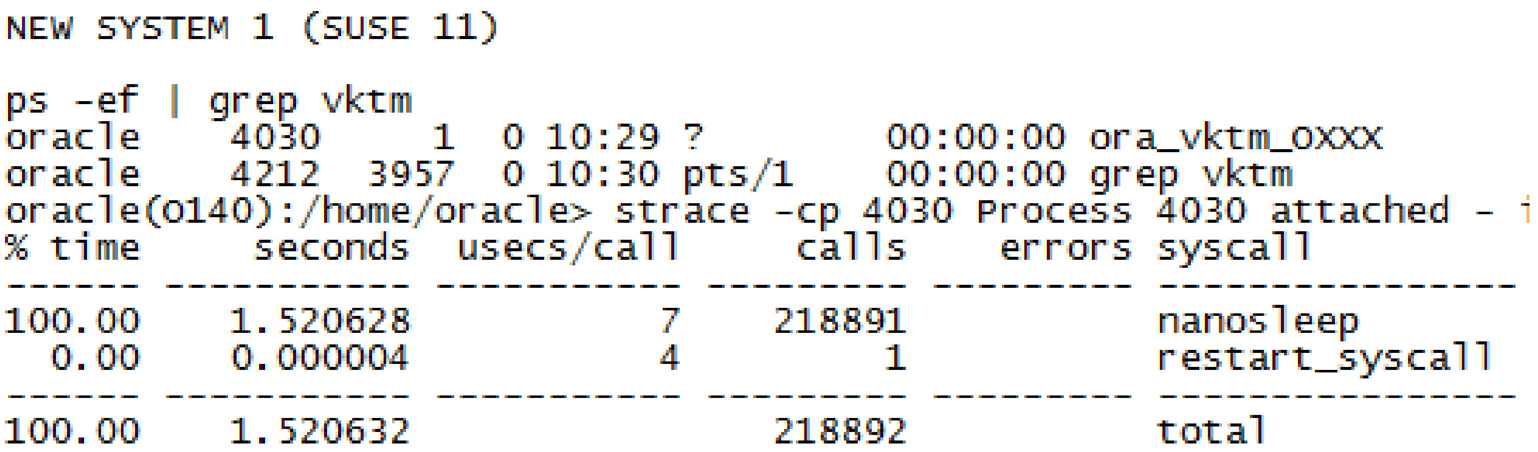
SUSE 11

kernel timer interrupt frequency is approx. 4000 Hz or higher

VKTM – OS Upgrade Reduces CPU Usage



```
OLD SYSTEM (SUSE 10)

ps -ef | grep vktm
oracle    1534     1  0 08:00 ?        00:00:08 ora_vktm_OXXX
oracle    1599     1  0 08:00 ?        00:00:08 ora_vktm_OXXX
home/oracle> strace -cp 1534
Process 1534 attached - interrupt to quit Process 1534 detached
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 99.21    0.174249          11     16455           nanosleep
  0.79    0.001393           0     33214           gettimeofday
------ ----------- ----------- --------- --------- ----------------
100.00    0.175642                 49669           total
```



```
NEW SYSTEM 1 (SUSE 11)

ps -ef | grep vktm
oracle    4030     1  0 10:29 ?        00:00:00 ora_vktm_OXXX
oracle    4212  3957  0 10:30 pts/1    00:00:00 grep vktm
oracle(o140):/home/oracle> strace -cp 4030 Process 4030 attached - i
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00    1.520628           7    218891           nanosleep
  0.00    0.000004           4         1           restart_syscall
------ ----------- ----------- --------- --------- ----------------
100.00    1.520632                218892           total
```

VKTM with Oracle 12c & 11gR2

Default Values 11gR2 & 12c:

```
_disable_highres_ticks  False
_timer_precision        10
```

```
% time     seconds  usecs/call   calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00     0.069437           1   125092    nanosleep
  0.00     0.000000           0        1     restart_syscall
------ ----------- ----------- --------- --------- ----------------
100.00    0.069437                125093                    total
```

VKTM Changes to Help Reduce CPU***:

```
_disable_highres_ticks  TRUE
_timer_precision        2000
```

***** Get Oracle support approval before using.**

```
% time  seconds  usecs/call     calls     errors syscall
------ ----------- ----------- --------- --------- ----------------
 99.81   0.002063           1      1496     nanosleep
  0.19   0.000004           4         1     restart_syscall
------ ----------- ----------- --------- --------- ----------------
100.00    0.002067                   1497                  total
```

Under Review

- Backups
- Auditing Storing
- Audit data review and storage
- FLASHBACK especially for test system recovery
- Security Hardening and Monitoring – We have been doing it for years, but some processes need tweaked/tuned.

Turning off Unneeded Services

- Keep the golden image as lean as possible in terms of processor usage, some of these services can be turned off with chkconfig command:

Red Hat 6.4+

```
# chkconfig iptables off
# chkconfig ip6tables off
# chkconfig auditd off
# chkconfig abrtd off
# chkconfig atd off
# chkconfig cups off
# chkconfig mdmonitor off
```

Sles 11 sp3+

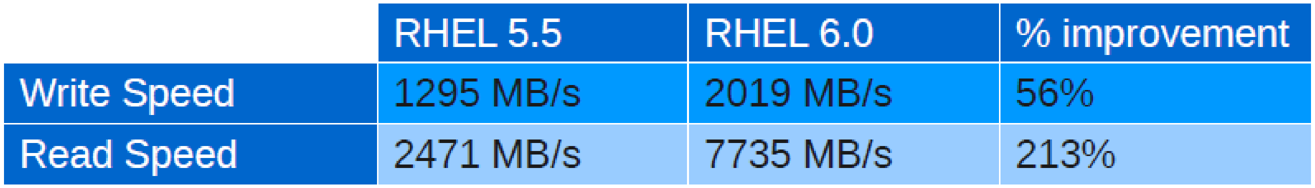
```
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd off
# chkconfig xinetd off
```

Source: The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3

OS Level Currency

- **Significant Performance & Security Improvements when upgrading OS Distribution levels:**

Red Hat Memory Performance:



	RHEL 5.5	RHEL 6.0	% improvement
Write Speed	1295 MB/s	2019 MB/s	56%
Read Speed	2471 MB/s	7735 MB/s	213%

Red Hat - A Performance Comparison Between RHEL 5 and RHEL 6 on System z

VDSO – Linux cpu Improvements

- **V**irtual **D**ynamically-linked **S**hared **O**bject (VDSO) is a shared library provided by the kernel. This allows normal programs to do certain system calls without the usual overhead of system calls like switching address spaces.

- Example by using the new VDSO implementation we have seen **six times** reduction in the number of function calls.

- Newer Linux distributions (RHEL 5.9 & 6.x, SLES 11) have this feature and it's enabled by default.

- Oracle calls Linux **gettimeofday()** hundreds of times a second for reporting statistics.

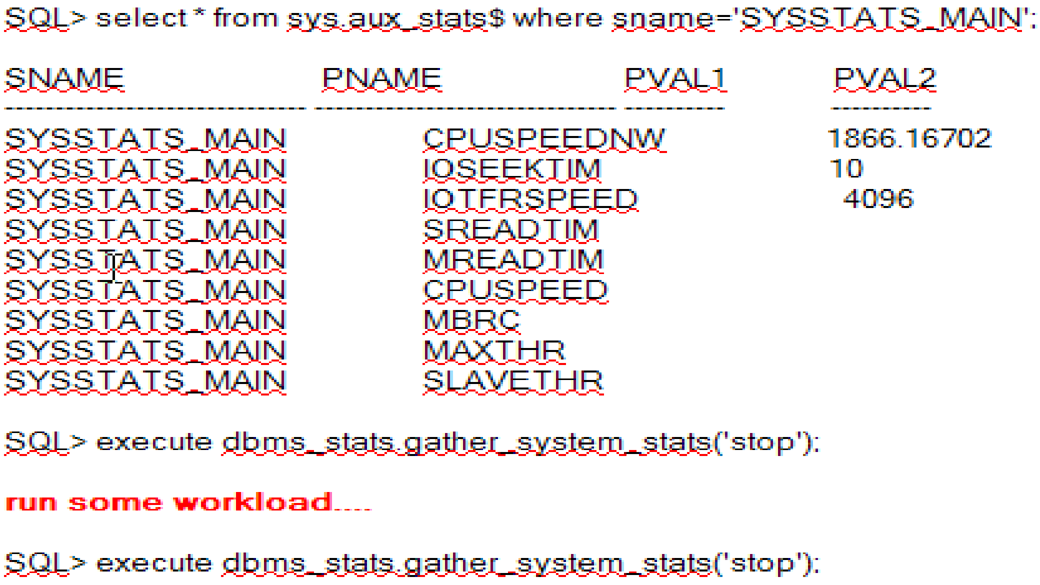
(Less Oracle Oracle products you install the less number of user calls)

- By upgrading Linux, VDSO reduces cpu costs, especially in virtualized environments

Oracle Optimizer Hints

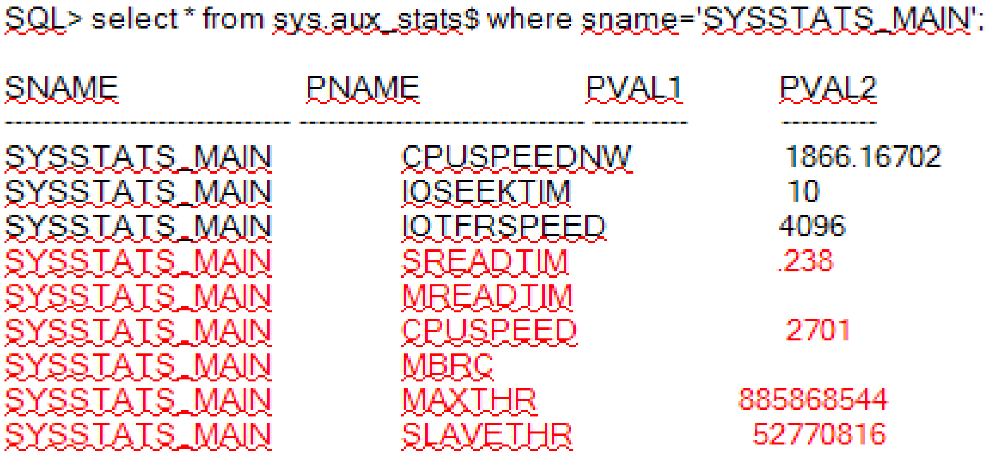
- Oracle calculates the cpu cost for a sql query plan with:
  - number cores (cpu_count)
  - optimizer_mode (all_rows, first_rows etc) and
  - the number of rows and Bytes in table.

Before updating System Statistics



```
SQL> select * from sys.aux_stats$ where sname='SYSSTATS_MAIN';

SNAME                          PNAME                          PVAL1      PVAL2
------------------------------ ------------------------------ ---------- ----------
SYSSTATS_MAIN                  CPUSPEEDNW                             1866.16702
SYSSTATS_MAIN                  IOSEEKTIM                              10
SYSSTATS_MAIN                  IOTFRSPEED                              4096
SYSSTATS_MAIN                  SREADTIM
SYSSTATS_MAIN                  MREADTIM
SYSSTATS_MAIN                  CPUSPEED
SYSSTATS_MAIN                  MBRC
SYSSTATS_MAIN                  MAXTHR
SYSSTATS_MAIN                  SLAVETHR

SQL> execute dbms_stats.gather_system_stats('stop');

run some workload....

SQL> execute dbms_stats.gather_system_stats('stop');
```

After updating System Statistics



```
SQL> select * from sys.aux_stats$ where sname='SYSSTATS_MAIN';

SNAME                          PNAME                          PVAL1      PVAL2
------------------------------ ------------------------------ ---------- ----------
SYSSTATS_MAIN                  CPUSPEEDNW                              1866.16702
SYSSTATS_MAIN                  IOSEEKTIM                               10
SYSSTATS_MAIN                  IOTFRSPEED                             4096
SYSSTATS_MAIN                  SREADTIM                               .238
SYSSTATS_MAIN                  MREADTIM
SYSSTATS_MAIN                  CPUSPEED                                2701
SYSSTATS_MAIN                  MBRC
SYSSTATS_MAIN                  MAXTHR                               885868544
SYSSTATS_MAIN                  SLAVETHR                              52770816
```

Oracle Optimize – Running Statics

exec DBMS_STATS.GATHER_SYSTEM_STATS('NOWORKLOAD');

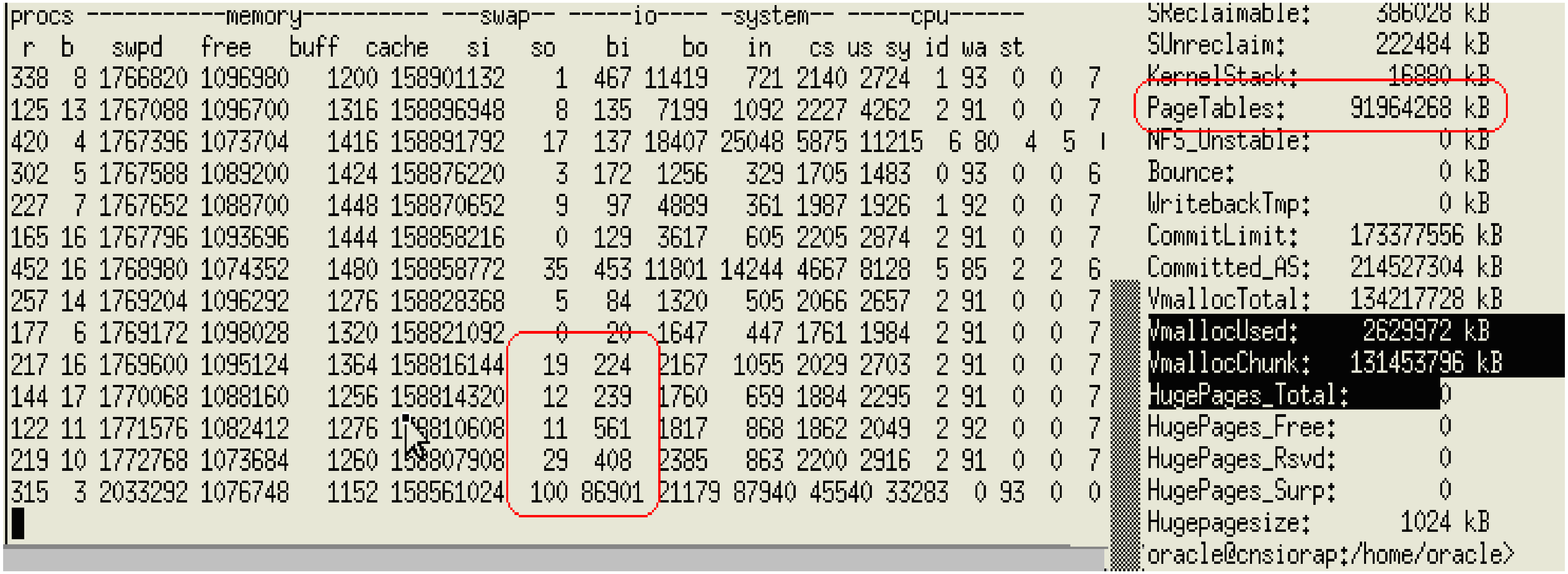
```
z9:
SNAME                          PNAME                                      PVAL1
------------------------------ ------------------------------             ----------
SYSSTATS_MAIN                  CPUSPEEDNW                                  533
Linux bogomips per cpu: 6510.00

z196:
SNAME                          PNAME                                      PVAL1
------------------------------ ------------------------------             ----------
SYSSTATS_MAIN                  CPUSPEEDNW                                  2335
Linux bogomips per cpu: 14367.00

zEC12:
SNAME                          PNAME                                       PVAL1
------------------------------ ------------------------------             ----------
SYSSTATS_MAIN                  CPUSPEEDNW                                  2613
Linux bogomips per cpu: 18115.00

z13:
SNAME                          PNAME                                      PVAL1
------------------------------ ------------------------------             ----------
SYSSTATS_MAIN                  CPUSPEEDNW                                  3123
Linux bogomips per cpu: 20325.00
```

Should be done for hardware upgrades on an idle system for each DB (dynamic)

Linux Huge Pages

- **Consider Using Linux Huge Pages for Oracle Database Memory**

  →In general 10-15% can be gained by the reduction in CPU usage as well as more memory for applications that would be otherwise consumed in Linux Page Tables…



```
procs -----------memory---------- ---swap-- -----io---- -system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
338  8 1766820 1096980   1200 158901132    1  467 11419   721 2140 2724  1 93  0  0  7
125 13 1767088 1096700   1316 158896948    8  135  7199  1092 2227 4262  2 91  0  0  7
420  4 1767396 1073704   1416 158891792   17  137 18407 25048 5875 11215  6 80  4  5  l
302  5 1767588 1089200   1424 158876220    3  172  1256   329 1705 1483  0 93  0  0  6
227  7 1767652 1088700   1448 158870652    9   97  4889   361 1987 1926  1 92  0  0  7
165 16 1767796 1093696   1444 158858216    0  129  3617   605 2205 2874  2 91  0  0  7
452 16 1768980 1074352   1480 158858772   35  453 11801 14244 4667 8128  5 85  2  2  6
257 14 1769204 1096292   1276 158828368    5   84  1320   505 2066 2657  2 91  0  0  7
177  6 1769172 1098028   1320 158821092    0   20  1647   447 1761 1984  2 91  0  0  7
217 16 1769600 1095124   1364 158816144   19  224  2167  1055 2029 2703  2 91  0  0  7
144 17 1770068 1088160   1256 158814320   12  239  1760   659 1884 2295  2 91  0  0  7
122 11 1771576 1082412   1276 158810608   11  561  1817   868 1862 2049  2 92  0  0  7
219 10 1772768 1073684   1260 158807908   29  408  2385   863 2200 2916  2 91  0  0  7
315  3 2033292 1076748   1152 158561024  100 86901 21179 87940 45540 33283  0 93  0  0
```

```
SReclaimable:     386028 kB
SUnreclaim:       222484 kB
KernelStack:       16880 kB
PageTables:     91964268 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:   173377556 kB
Committed_AS:  214527304 kB
VmallocTotal:  134217728 kB
VmallocUsed:     2629972 kB
VmallocChunk:  131453796 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       1024 kB
oracle@cnsiorap:/home/oracle>
```

Huge Page Considerations:

- Can not use **MEMORY_TARGET** with Huge Pages.
  - Set manually to **SGA_TARGET** not including the **PGA_AGGREGATE_TARGET**.

- Not swappable: Huge Pages are not swappable

- General guideline consider when combined Oracle SGA's are greater than **8 GB** (particularly if a lots of connections)

- Decreased page table overhead; more memory can be freed up for other uses. i.e. more Oracle SGA memory, and less physical I/O's (See also Oracle Note: **361468.1**)

Recommendation: Use Huge Pages under z/VM

- Under z/VM (which has 4K pages) it’s still recommended to use Huge Pages for SGA’s > 10GB particularly with many connections

- Saves Memory that would otherwise be used for pagetables

- Stability for user process spikes (avoiding swap)

- Less work to manage smaller number of pagetables

- ~10% improvement (ROT) for memory intensive databases

Oracle Upgrade 11.2.0.4 -> 12.1.0.1 - CPU Intensive Test

18.9% improvement in response time between 11.2.0.4 & 12.1 (cpu intensive test)

Oracle 11.2.0.4

Running Parallel Processes: 32

real 0m12.01s

user 0m0.20s

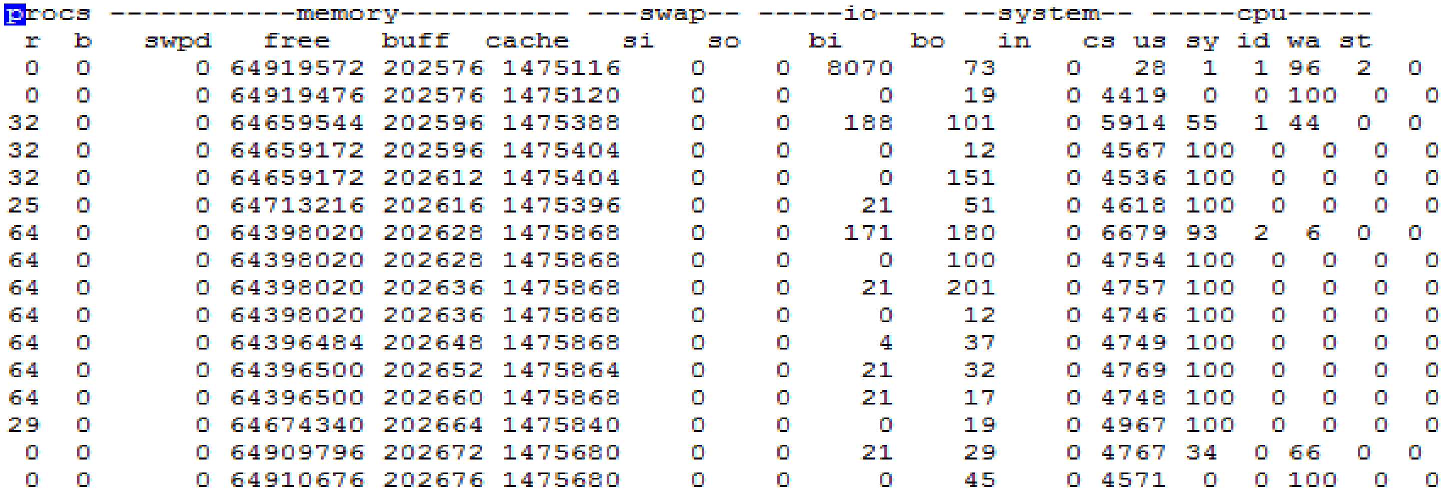
sys 0m0.13s

Running Parallel Processes: 64

real 0m23.84s

user 0m0.40s

sys 0m0.26s



```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 64919572 202576 1475116    0    0  8070    73    0   28  1  1 96  2  0
 0  0      0 64919476 202576 1475120    0    0     0    19    0 4419  0  0 100  0  0
32  0      0 64659544 202596 1475388    0    0   188   101    0 5914 55  1 44  0  0
32  0      0 64659172 202596 1475404    0    0     0    12    0 4567 100  0  0  0  0
32  0      0 64659172 202612 1475404    0    0     0   151    0 4536 100  0  0  0  0
25  0      0 64713216 202616 1475396    0    0    21    51    0 4618 100  0  0  0  0
64  0      0 64398020 202628 1475868    0    0   171   180    0 6679 93  2  6  0  0
64  0      0 64398020 202628 1475868    0    0     0   100    0 4754 100  0  0  0  0
64  0      0 64398020 202636 1475868    0    0    21   201    0 4757 100  0  0  0  0
64  0      0 64398020 202636 1475868    0    0     0    12    0 4746 100  0  0  0  0
64  0      0 64396484 202648 1475868    0    0     4    37    0 4749 100  0  0  0  0
64  0      0 64396500 202652 1475864    0    0    21    32    0 4769 100  0  0  0  0
64  0      0 64396500 202660 1475868    0    0    21    17    0 4748 100  0  0  0  0
29  0      0 64674340 202664 1475840    0    0     0    19    0 4967 100  0  0  0  0
 0  0      0 64909796 202672 1475680    0    0    21    29    0 4767 34  0 66  0  0
 0  0      0 64910676 202676 1475680    0    0     0    45    0 4571  0  0 100  0  0
```

Oracle 12.1.0.1

Running Parallel Processes: 32

real 0m10.12s

user 0m0.16s

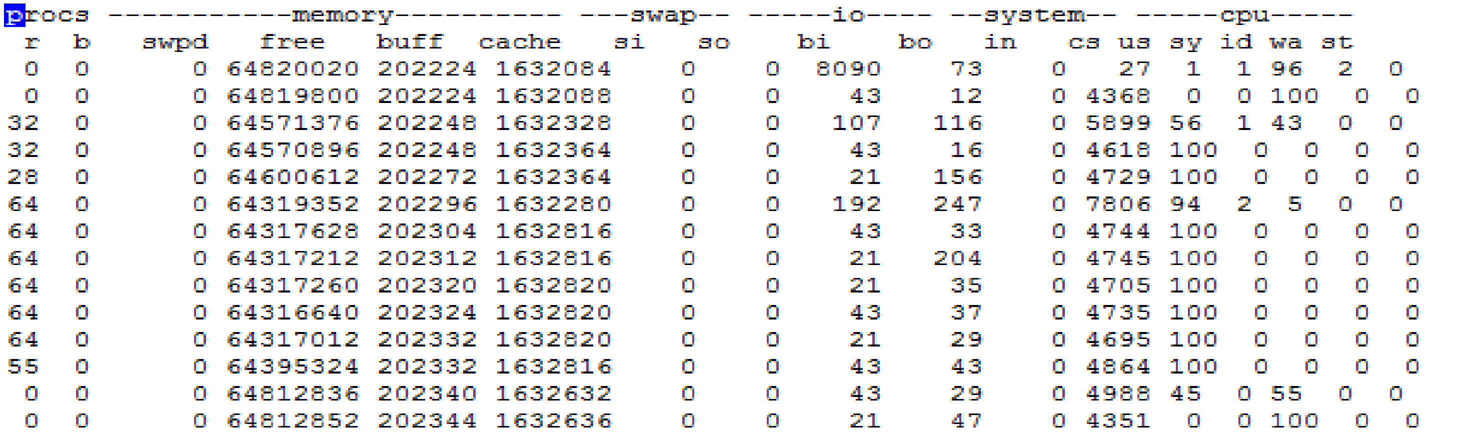
sys 0m0.14s

Running Parallel Processes: 64

real 0m20.05s

user 0m0.34s

sys 0m0.27s



```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 64820020 202224 1632084    0    0  8090    73    0   27  1  1 96  2  0
 0  0      0 64819800 202224 1632088    0    0    43    12    0 4368  0  0 100  0  0
32  0      0 64571376 202248 1632328    0    0   107   116    0 5899 56  1 43  0  0
32  0      0 64570896 202248 1632364    0    0    43    16    0 4618 100  0  0  0  0
28  0      0 64600612 202272 1632364    0    0    21   156    0 4729 100  0  0  0  0
64  0      0 64319352 202296 1632280    0    0   192   247    0 7806 94  2  5  0  0
64  0      0 64317628 202304 1632816    0    0    43    33    0 4744 100  0  0  0  0
64  0      0 64317212 202312 1632816    0    0    21   204    0 4745 100  0  0  0  0
64  0      0 64317260 202320 1632820    0    0    21    35    0 4705 100  0  0  0  0
64  0      0 64316640 202324 1632820    0    0    43    37    0 4735 100  0  0  0  0
64  0      0 64317012 202332 1632820    0    0    21    29    0 4695 100  0  0  0  0
55  0      0 64395324 202332 1632816    0    0    43    43    0 4864 100  0  0  0  0
 0  0      0 64812836 202340 1632632    0    0    43    29    0 4988 45  0 55  0  0
 0  0      0 64812852 202344 1632636    0    0    21    47    0 4351  0  0 100  0  0
```

11.2.0.4 -> 12.1.0.1 - I/O Test

Oracle I/O Calibrate (high I/O) Test:

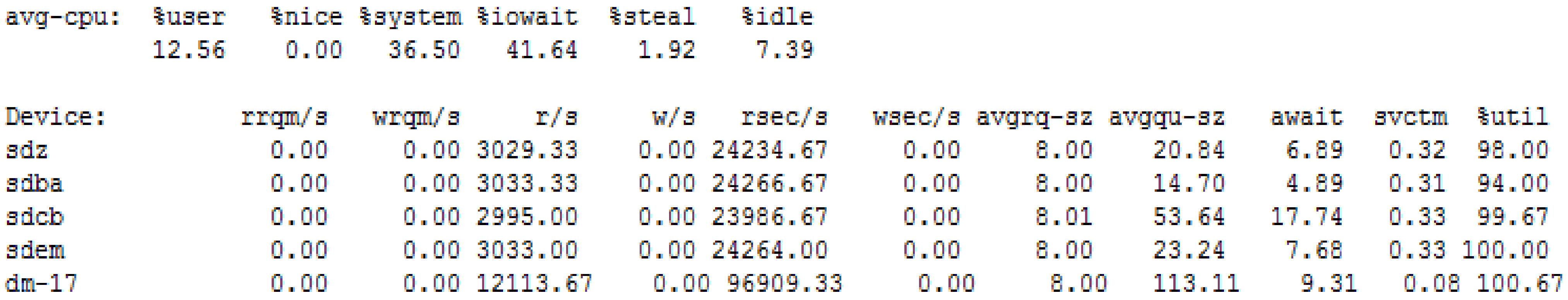
- **Not much change between releases (for this particular I/O test)**

Oracle 11.2.0.4

max_iops = **332989**
latency = 0
max_mbps = 3109

Oracle 12.1.0.1

max_iops = **333576**
latency = 0
max_mbps = 3116



```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          12.56    0.00   36.50   41.64    1.92    7.39

Device:         rrqm/s   wrqm/s     r/s     w/s   rsec/s   wsec/s avgrq-sz avgqu-sz   await  svctm  %util
sdz               0.00     0.00 3029.33    0.00 24234.67     0.00     8.00    20.84    6.89   0.32  98.00
sdba              0.00     0.00 3033.33    0.00 24266.67     0.00     8.00    14.70    4.89   0.31  94.00
sdcb              0.00     0.00 2995.00    0.00 23986.67     0.00     8.01    53.64   17.74   0.33  99.67
sdem              0.00     0.00 3033.00    0.00 24264.00     0.00     8.00    23.24    7.68   0.33 100.00
dm-17             0.00     0.00 12113.67     0.00 96909.33     0.00     8.00   113.11    9.31   0.08 100.67
```

Oracle 12c Trace File Analyzer Disable

Trace File Analyzer Collector (TFA): collects log and trace files from all nodes and products into a single location.

- Written in Java with its own JVM
- Large memory footprint for the heap etc.
- Can be disabled with a single command
- **Note:** next time you run rootcrs.pl (patching for example) it may reinstall itself.

Stop TFA

```
# /etc/init.d/init.tfa stop
```

Start TFA

```
# /etc/init.d/init.tfa  start
```

Stop and removes related inittab entries

```
# /etc/init.d/init.tfa shutdown
```

Oracle 12c Cluster Verification Utility (CVU) - Disable

Cluster Verification Utility (CVU):

- The CVU tool automatically runs, pointing out configuration issue.
- In Oracle 12.1.0.2, scheduled to run automatically every time the cluster is started and periodically after that.
- The CVU itself and checks use CPU and RAM resources, and are better run manually when such resources are limited.
- It’s a quick removal

```
# crs_stat -t
Name          Type          Target    State     Host
------------------------------------------------------------
ora....ER.lsnr ora....er.type ONLINE    ONLINE    clone01
ora....N1.lsnr ora....er.type ONLINE    ONLINE    clone01
ora....N2.lsnr ora....er.type ONLINE    ONLINE    clone01
ora....N3.lsnr ora....er.type ONLINE    ONLINE    clone01
ora.OCR2.dg    ora....up.type ONLINE    ONLINE    clone01
ora.asm        ora.asm.type   ONLINE    ONLINE    clone01
ora....SM1.asm application    ONLINE    ONLINE    clone01
ora....01.lsnr application    ONLINE    ONLINE    clone01
ora....e01.ons application    ONLINE    ONLINE    clone01
ora....e01.vip ora....t1.type ONLINE    ONLINE    clone01
ora.cvu        ora.cvu.type   ONLINE    ONLINE    clone01
ora....network ora....rk.type ONLINE    ONLINE    clone01
ora.oc4j       ora.oc4j.type  OFFLINE   OFFLINE
ora.ons        ora.ons.type   ONLINE    ONLINE    clone01
ora.scan1.vip  ora....ip.type ONLINE    ONLINE    clone01
ora.scan2.vip  ora....ip.type ONLINE    ONLINE    clone01
ora.scan3.vip  ora....ip.type ONLINE    ONLINE    clone01

# srvctl stop cvu -force
```

Source: Marc Fielding http://www.pythian.com/blog/slimming-down-oracle-rac-12cs-resource-footprint/

Oracle 12c OC4J – Ensure Disabled

OC4J:

- Every Oracle 12c grid install contains OC4J
- Linux on System z oc4j is disabled by default.
- Ensure oc4j is disabled.

```
# crs_stat -t
Name           Type           Target    State     Host
------------------------------------------------------------
ora....ER.lsnr ora....er.type ONLINE    ONLINE    clone01
ora....N1.lsnr ora....er.type ONLINE    ONLINE    clone01
ora....N2.lsnr ora....er.type ONLINE    ONLINE    clone01
ora....N3.lsnr ora....er.type ONLINE    ONLINE    clone01
ora.OCR2.dg    ora....up.type ONLINE    ONLINE    clone01
ora.asm        ora.asm.type   ONLINE    ONLINE    clone01
ora....SM1.asm application    ONLINE    ONLINE    clone01
ora....01.lsnr application    ONLINE    ONLINE    clone01
ora....e01.ons application    ONLINE    ONLINE    clone01
ora....e01.vip ora....t1.type ONLINE    ONLINE    clone01
ora.cvu        ora.cvu.type   OFFLINE   OFFLINE
ora....network ora....rk.type ONLINE    ONLINE    clone01
ora.oc4j       ora.oc4j.type  OFFLINE   OFFLINE
ora.ons        ora.ons.type   ONLINE    ONLINE    clone01
ora.scan1.vip  ora....ip.type ONLINE    ONLINE    clone01
ora.scan2.vip  ora....ip.type ONLINE    ONLINE    clone01
ora.scan3.vip  ora....ip.type ONLINE    ONLINE    clone01
```

Oracle 12c – JIT Compiler Improvements

- New in Oracle 12.1.0.1+ JIT Compiler for Java Stored Procedures versus interpreted.

Oracle 11.2.0.4

```
alter session set java_jit_enabled=true;
ERROR: ORA-02097: parameter cannot be
modified because specified value is invalid

var time_compiled NUMBER;
var time_interpreted NUMBER;
exec :time_compiled := factorial(20);

alter session set java_jit_enabled=false;
exec :time_interpreted := factorial(20);
```

```
INTERP_TIME_MS              JIT_TIME_MS
      2893                  2856
```

Oracle 12.1.0.1

```
alter session set java_jit_enabled=true;
-- Force compile
select dbms_java.compile_method
('JITDemo', 'factorial', '(J)J') from dual;

var time_compiled NUMBER;
var time_interpreted NUMBER;
exec :time_compiled := factorial(20);

alter session set java_jit_enabled=false;
exec :time_interpreted := factorial(20);
```

```
INTERP_TIME_MS              JIT_TIME_MS
      4148                  182
```

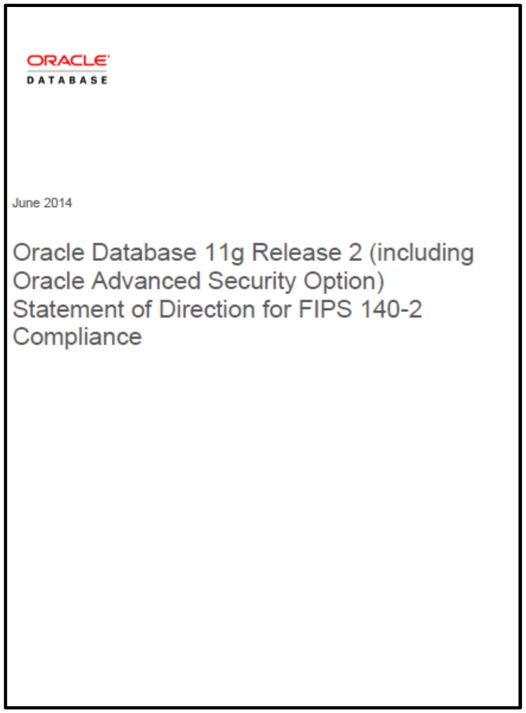
Oracle FIPS 140-2 Security Compliance

- NIST US Government Organization for Security Compliance certifications
- Statement of Direction from Oracle's Advanced Security Team
- Included in 12.1.0.2 release
- 11.2.0.4 -> **Patch 19207156: MES BUNDLE ON TOP OF RDBMS 11.2.0.4.2**

- FIPS140 Transparent Data Encryption (TDE) activated with parameter:
  - ALTER SYSTEM SET DBFIPS_140 = TRUE;

- FIPS140 Secure Sockets Layer (SSL) activated with SQL*Net parameter:
  - $ORACLE_HOME/ldap/admin/fips.ora

```
SSLFIPS_140=TRUE
```



ORACLE DATABASE

June 2014

Oracle Database 11g Release 2 (including Oracle Advanced Security Option) Statement of Direction for FIPS 140-2 Compliance

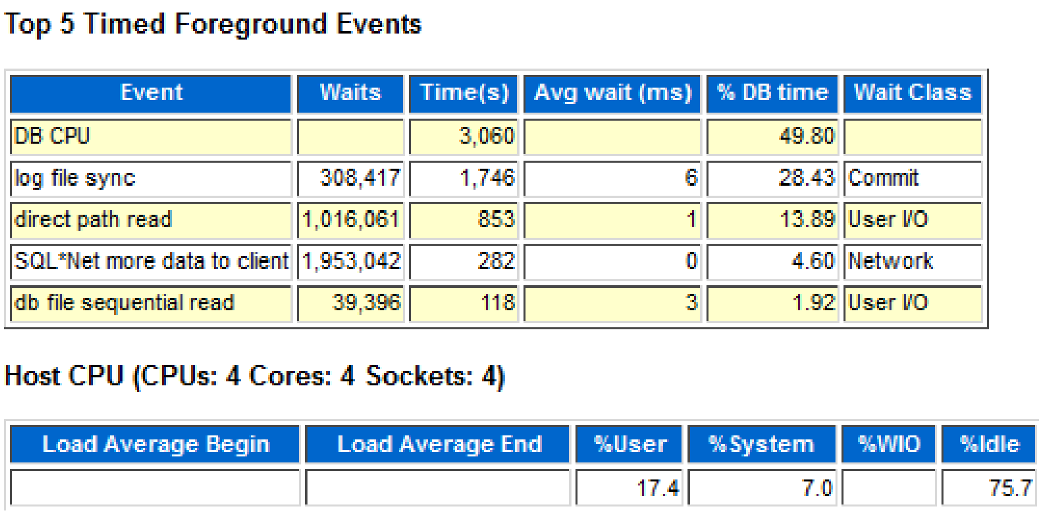
Customer Case:

- Running Oracle with Linux on System z since 2010.
- Moved a new workloads to Linux on System z.
- Encountered some unique performance challenges with one Application moving to System z.

Customer Experience Case (pg 2)

Testing new workload with Linux on System z, higher than expected CPU observed in testing.

Source non System z – 4 cpu cores no RAC



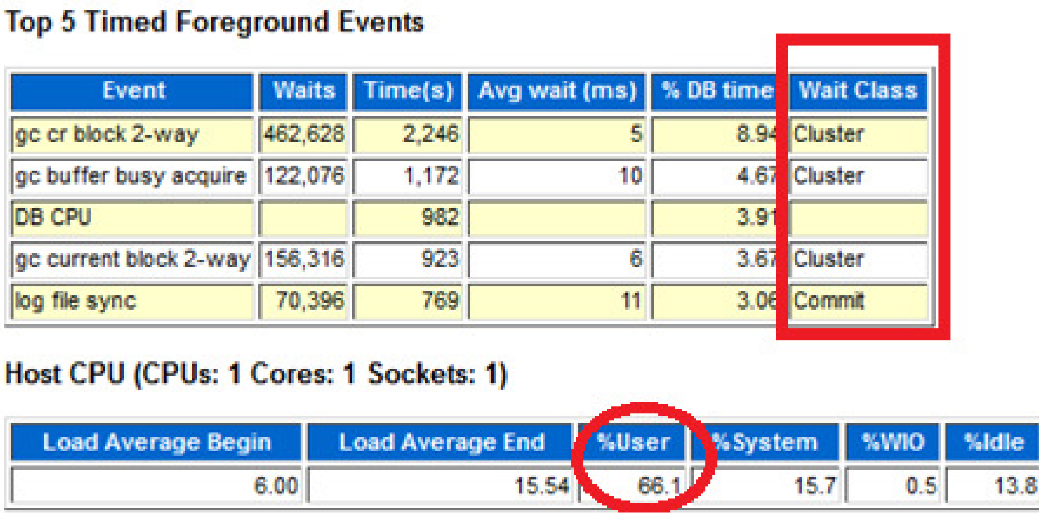
Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		3,060		49.80	
log file sync	308,417	1,746	6	28.43	Commit
direct path read	1,016,061	853	1	13.89	User I/O
SQL*Net more data to client	1,953,042	282	0	4.60	Network
db file sequential read	39,396	118	3	1.92	User I/O

Host CPU (CPUs: 4 Cores: 4 Sockets: 4)

Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
		17.4	7.0		75.7

Target System z – 1 cpu core w/ RAC



Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
gc cr block 2-way	462,628	2,246	5	8.94	Cluster
gc buffer busy acquire	122,076	1,172	10	4.67	Cluster
DB CPU		982		3.91	
gc current block 2-way	156,316	923	6	3.67	Cluster
log file sync	70,396	769	11	3.06	Commit

Host CPU (CPUs: 1 Cores: 1 Sockets: 1)

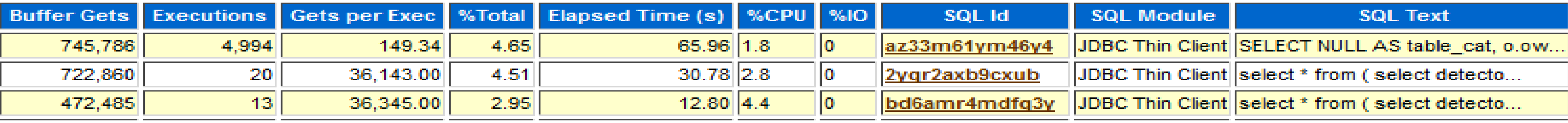
Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
6.00	15.54	66.1	15.7	0.5	13.8

Asked to do an Oracle Health Check…

Initial Generic Recommendations:

- Sometimes with highly transactional Oracle RAC databases 4K Oracle block size can help.
  - **Considered 4K block region for small indexes and sequences to reduce Interconnect times**
- For Oracle RAC databases – suggest creating a “service” for distinct workloads
- Oracle sequence caching, pre fetch 1000+ sequences (noorder) on each RAC Cluster node
- Tune SQL & Investigate index compression (pctfree 0) for queries with block contention.

SQL Ordered by Gets (AWR Report)



Buffer Gets	Executions	Gets per Exec	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
745,786	4,994	149.34	4.65	65.96	1.8	0	az33m61ym46y4	JDBC Thin Client	SELECT NULL AS table_cat, o.ow...
722,860	20	36,143.00	4.51	30.78	2.8	0	2yqr2axb9cxub	JDBC Thin Client	select * from (select detecto...
472,485	13	36,345.00	2.95	12.80	4.4	0	bd6amr4mdfq3y	JDBC Thin Client	select * from (select detecto...

After Some Data…. Initial Recommendations…

(1) Turn REORDER (VM) off for Large Virtual Memory Oracle Linux guests.

- Reorder can delay the guest for approximately 1seconds per 8g.
- Recommend turn REORDER off for any Oracle guest 8G or greater.
- Upgrade to z/VM 6.3 (which has re-order turned off automatically)

(2) Oracle Support Recommended -> turn off ASLR (System z Linux kernel parameter – Address Space

- ASLR Linux feature is enabled and it is recommended to disable it by setting kernel.randomize_va_space = 0.
- For it add/modify this parameter in /etc/sysctl.conf   kernel.randomize_va_space=0

(3) Increase virtual memory from 8GB to 10GB on each guest

- Observed some Linux swap when running on just one Linux guest

(4) Implement HugePages for better memory management.

- kernel parameter change, and re-start Oracle instances
- Reduces the # of pages the kernel must manage and makes the system more efficient

(5) Increase Linux guests from 1 virtual CP to 2 virtual CPs

- Increase share so it can use 2 IFL's if it needs the resources.
- Objective is to get the workload to run; then scale back after we get the a successful run.

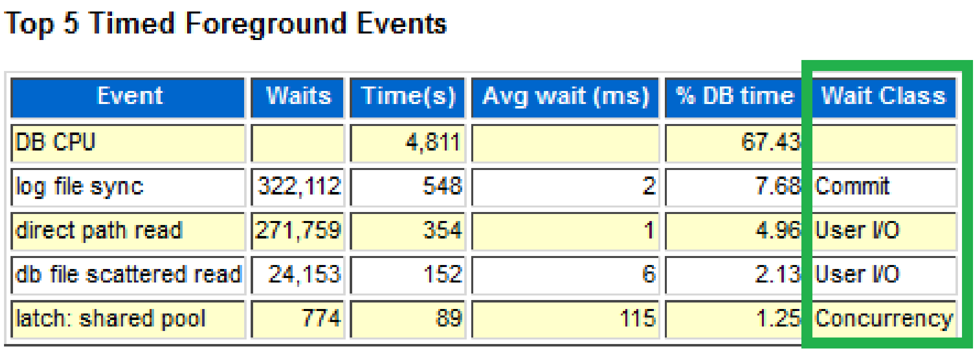
(6) Run the workload in a non RAC environment to tune, then implement with Oracle RAC

- Once ran with good performance single-node, then implement the desired 2-node environment to provide the high availability.

Resolution Steps

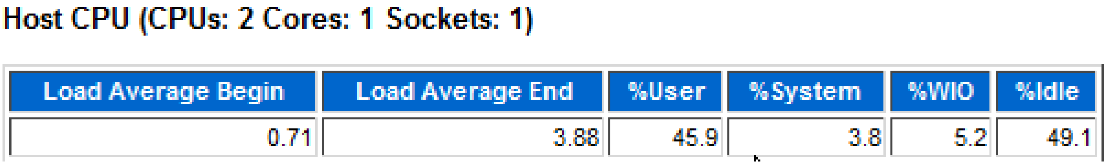
Adjusting # of Virtual IFLs from 1 -> 2 reduced Oracle concurrency and overall cpu load

Linux on System z – 2 virtual cpu non RAC



Top 5 Timed Foreground Events

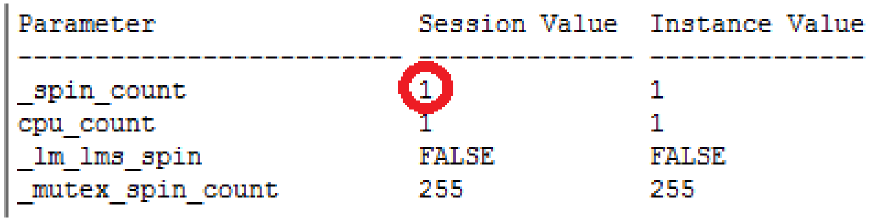
Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		4,811		67.43	
log file sync	322,112	548	2	7.68	Commit
direct path read	271,759	354	1	4.96	User I/O
db file scattered read	24,153	152	6	2.13	User I/O
latch: shared pool	774	89	115	1.25	Concurrency



Host CPU (CPUs: 2 Cores: 1 Sockets: 1)

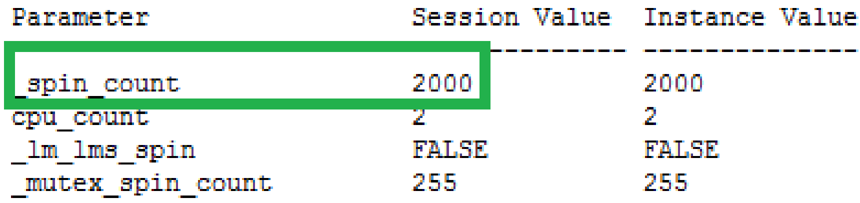
Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
0.71	3.88	45.9	3.8	5.2	49.1

Oracle Parm with 1 Virtual IFL



```
Parameter                 Session Value  Instance Value
------------------------- -------------- --------------
_spin_count               1              1
cpu_count                 1              1
_lm_lms_spin              FALSE          FALSE
_mutex_spin_count         255            255
```

Oracle Parm with 2 Virtual IFL's



```
Parameter                 Session Value  Instance Value
------------------------- -------------- --------------
_spin_count               2000           2000
cpu_count                 2              2
_lm_lms_spin              FALSE          FALSE
_mutex_spin_count         255            255
```

- An Oracle latch helps prevent two processes from simultaneously updating the same area of the SGA. Oracle **spin_count parameter is based on # of cpus**. _spin_count will wait x# of cpu cycles per Oracle process that has to wait for a latch that is busy.

z/VM 6.3 with SMT Enabled

```
# vmcp q mt
Multithreading is enabled.
             Requested    Activated
             Threads      Threads
MAX_THREADS    MAX          2
CP core        MAX          1
IFL core       MAX          2
ICF core       MAX          1
zIIP core      MAX          1
```

```
cat /proc/cpuinfo
vendor_id       : IBM/S390
# processors    : 24
bogomips per cpu: 20325.00
features        : esan3 zarch stfle msa ldisp eimm dfp etf3eh highgprs
processor 0: version = FF,  identification = 05DA97,  machine = 2964
processor 1: version = FF,  identification = 05DA97,  machine = 2964
processor 2: version = FF,  identification = 05DA97,  machine = 2964
processor 3: version = FF,  identification = 05DA97,  machine = 2964
…
processor 22: version = FF,  identification = 05DA97,  machine = 2964
processor 23: version = FF,  identification = 05DA97,  machine = 2964
```

- Oracle is licensed by the # of physical CPU Cores (IFLs) in a Hard Partitioned LPAR.

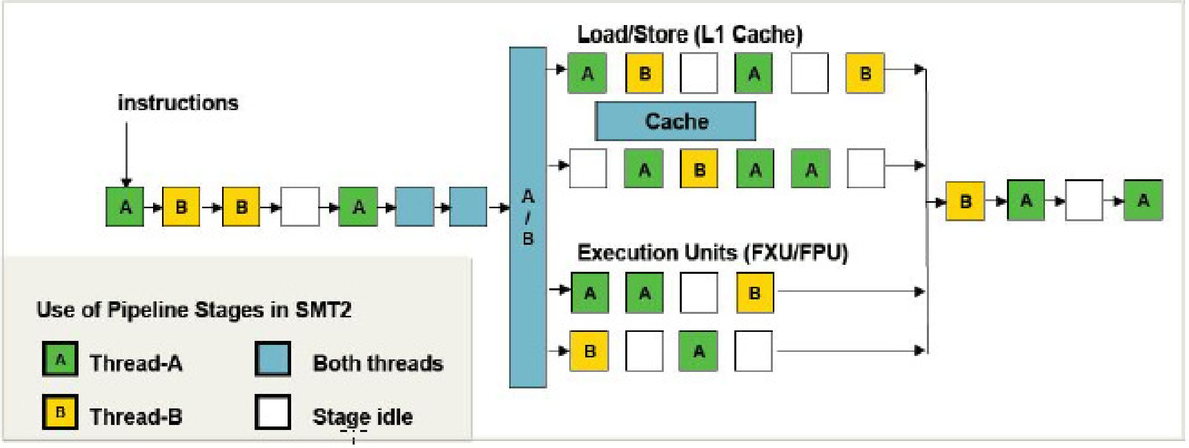
- With z/VM SMT enabled the number of processors will show as the number of virtual processor threads that have been allocated and is not what is licensed on.

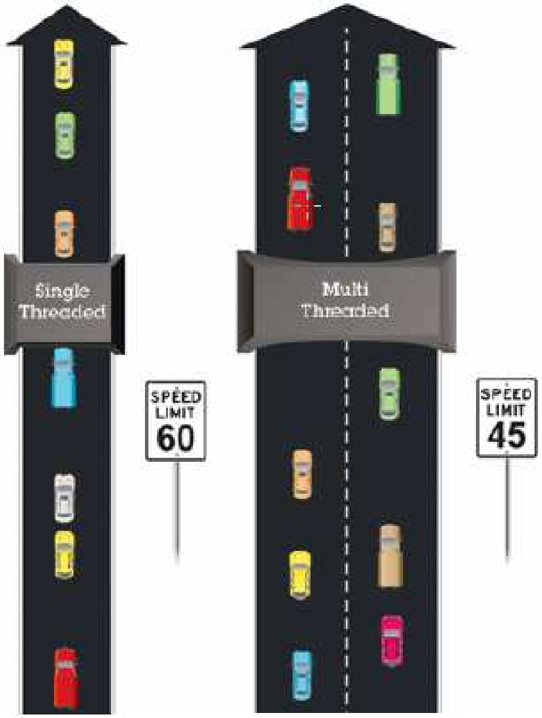
z13 (z/VM with SMT Enabled)

- Double the number of hardware threads per core
  - Independent threads can be more effective utilizing pipeline

- Threads share resources – may impact single thread perf
  - Pipeline (eg. physical registers, fxu, fpu, lsu etc)
  - Cache

- Throughput improvement is workload dependent



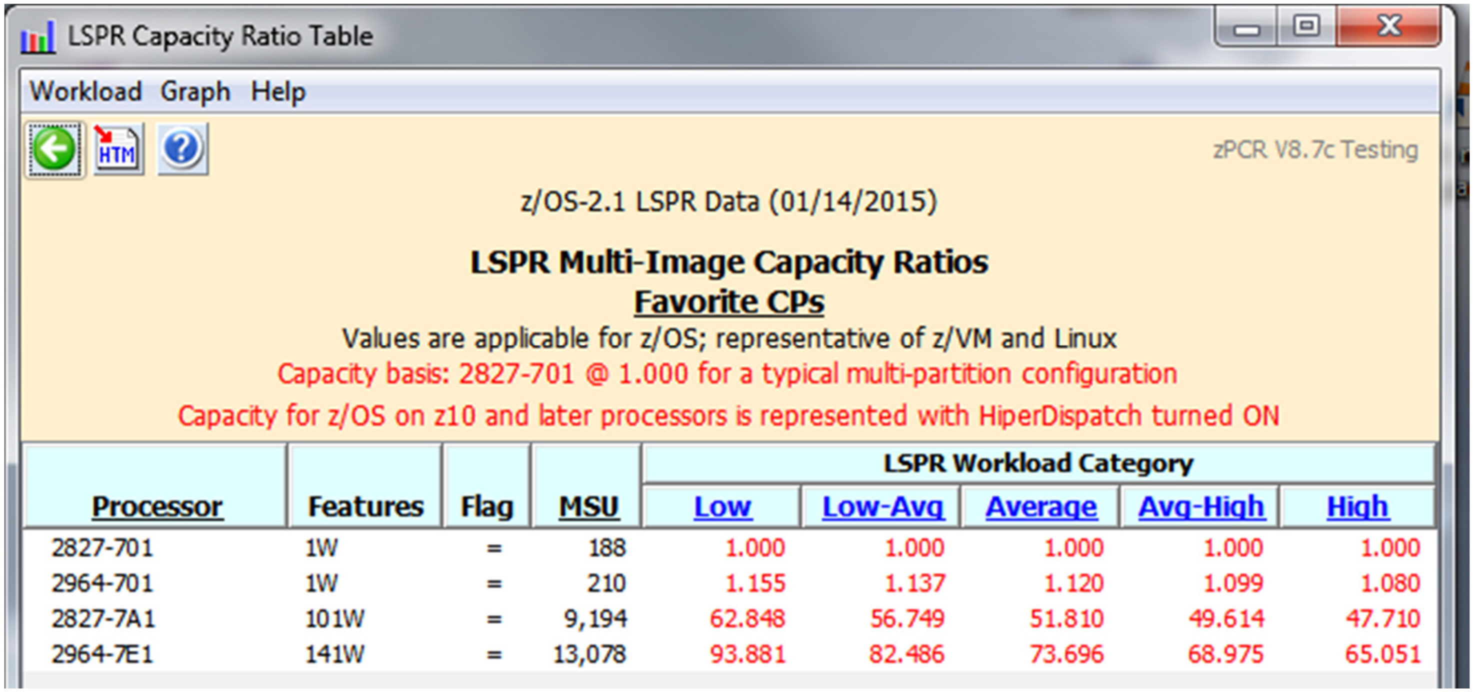


Which approach is designed for the highest volume of traffic? Which road is faster?

** Illustrative numbers only*

New! - IBM z13 CPU Performance

- Published performance improvement with out SMT (threading) is **12%** and **32%** for workloads that can benefit from SMT.
- **SMT** - Pre-install guidance based on internal testing and eventual field experience (20% for IFLs, 25% for zIIPs)
- **.For Oracle workloads were seeing performance gains consistent with these z13 SMT performance guidance.**



zPCR V8.7c Testing

z/OS-2.1 LSPR Data (01/14/2015)

LSPR Multi-Image Capacity Ratios

Favorite CPs

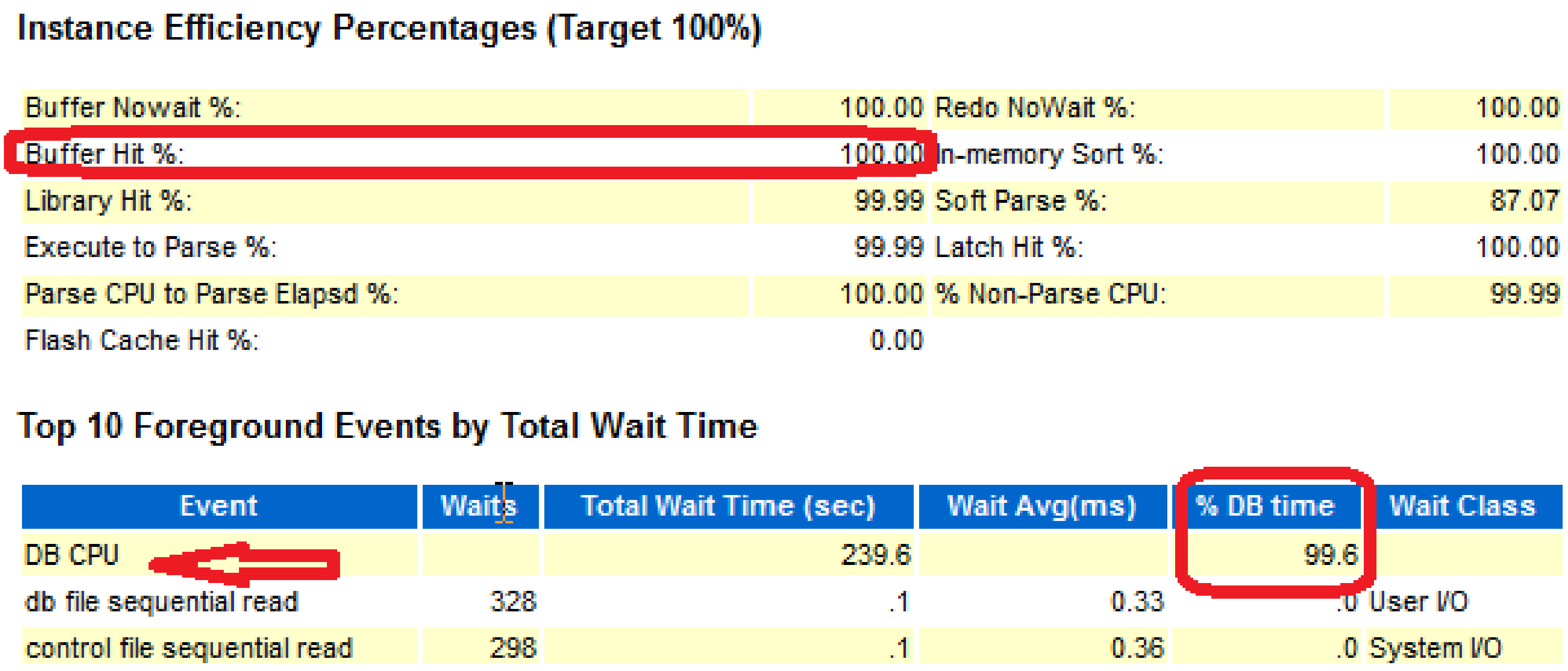
Values are applicable for z/OS; representative of z/VM and Linux

Capacity basis: 2827-701 @ 1.000 for a typical multi-partition configuration

Capacity for z/OS on z10 and later processors is represented with HiperDispatch turned ON

				LSPR Workload Category				
Processor	Features	Flag	MSU	Low	Low-Avg	Average	Avg-High	High
2827-701	1W	=	188	1.000	1.000	1.000	1.000	1.000
2964-701	1W	=	210	1.155	1.137	1.120	1.099	1.080
2827-7A1	101W	=	9,194	62.848	56.749	51.810	49.614	47.710
2964-7E1	141W	=	13,078	93.881	82.486	73.696	68.975	65.051

Testing on New z13 with 2 Dedicated IFLs



Instance Efficiency Percentages (Target 100%)

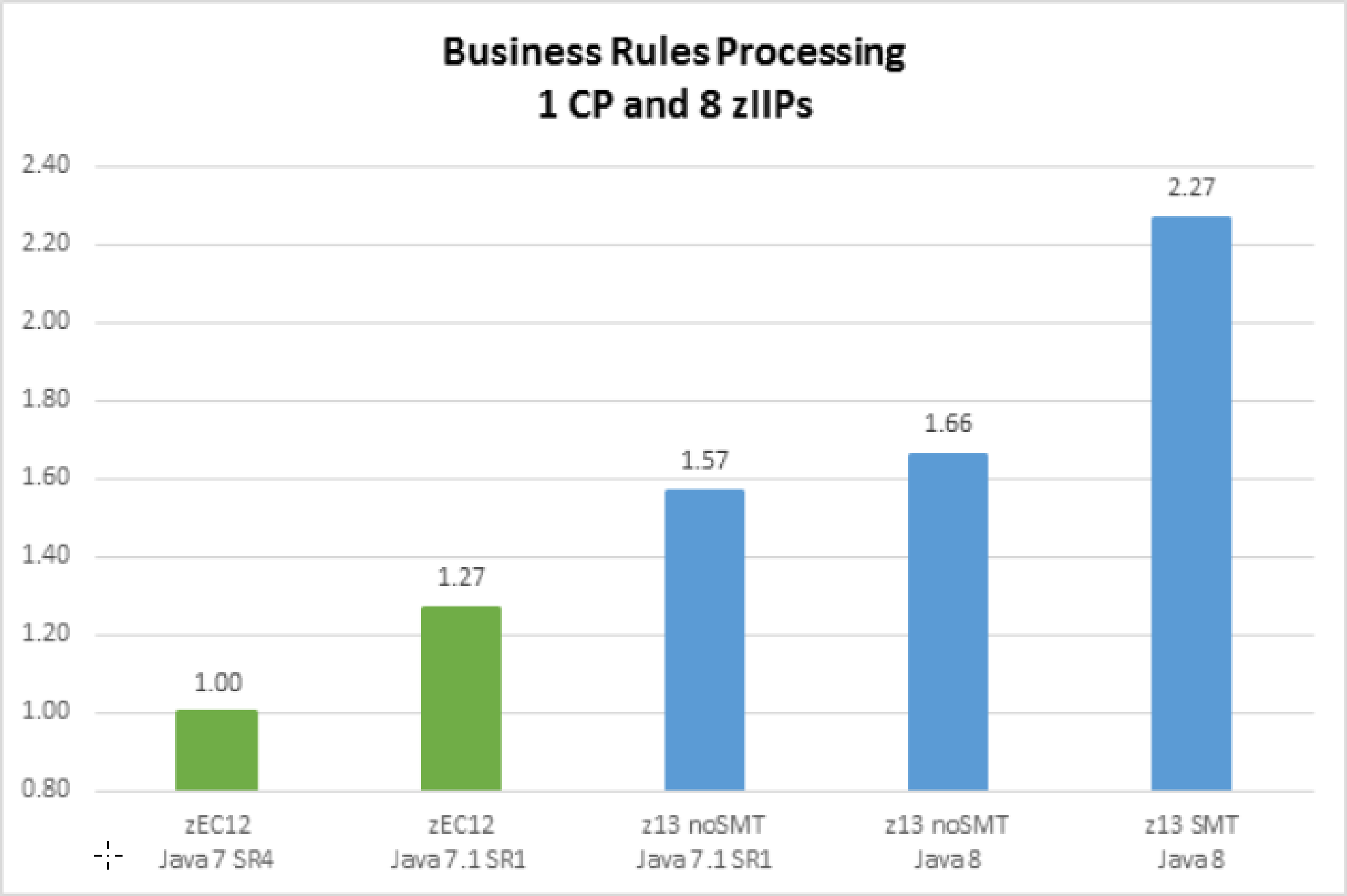
Buffer Nowait %:	100.00	Redo NoWait %:	100.00
Buffer Hit %:	100.00	In-memory Sort %:	100.00
Library Hit %:	99.99	Soft Parse %:	87.07
Execute to Parse %:	99.99	Latch Hit %:	100.00
Parse CPU to Parse Elapsd %:	100.00	% Non-Parse CPU:	99.99
Flash Cache Hit %:	0.00		

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		239.6		99.6	
db file sequential read	328	.1	0.33	.0	User I/O
control file sequential read	298	.1	0.36	.0	System I/O

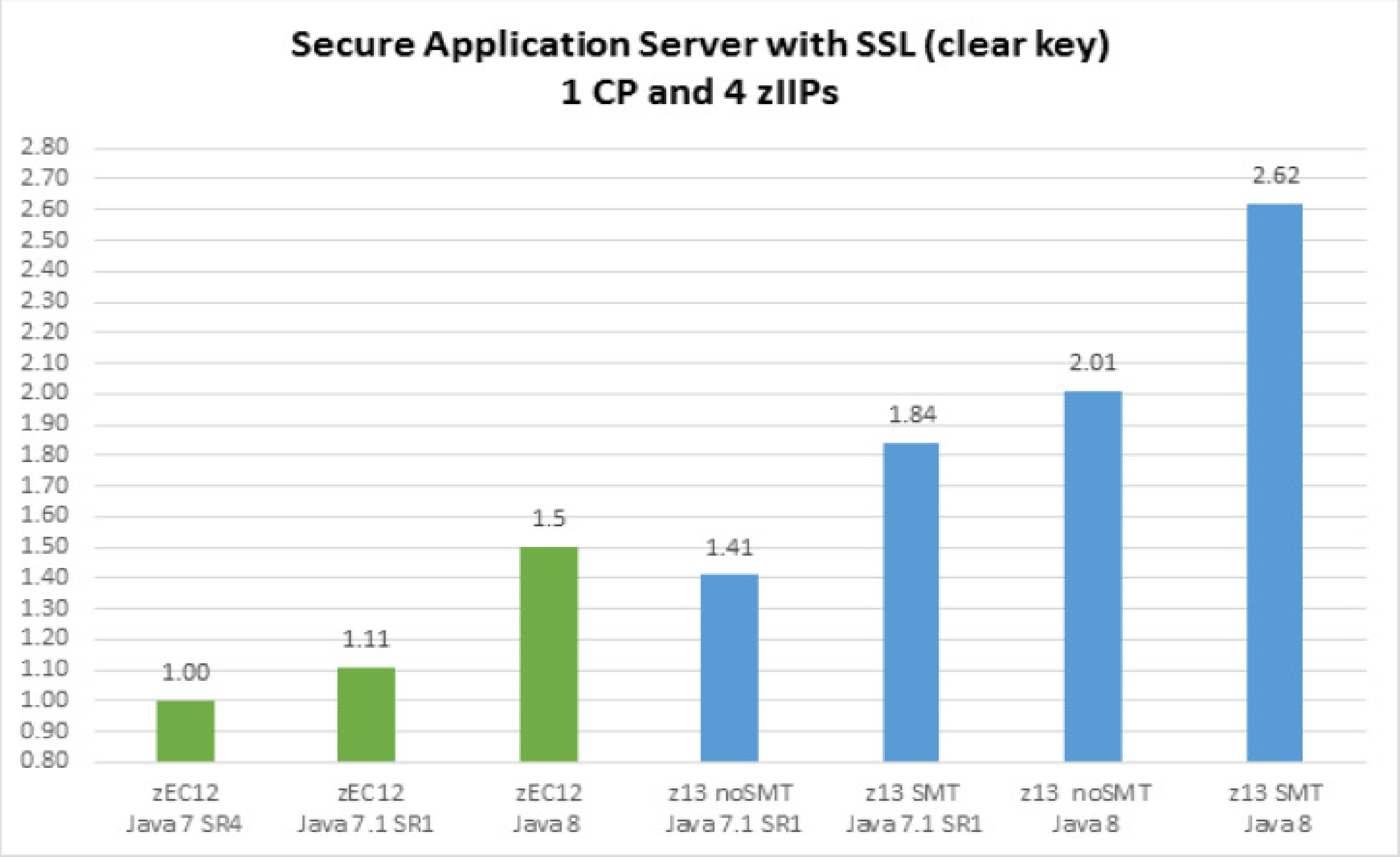
- Silly Little Oracle Benchmark (SLOB) – (Kevin Closson – author)
- Logical I/O (Random memory access to Oracle SGA)
- Want to have 99% + DB CPU and 100% Buffer Hit Ratio for a clean test from Oracle Automatic Workload Repository (AWR) Report.

z13 Java Performance Improvements – z/OS



Aggregate 2.27x improvement from IBM Java 8 and IBM z13

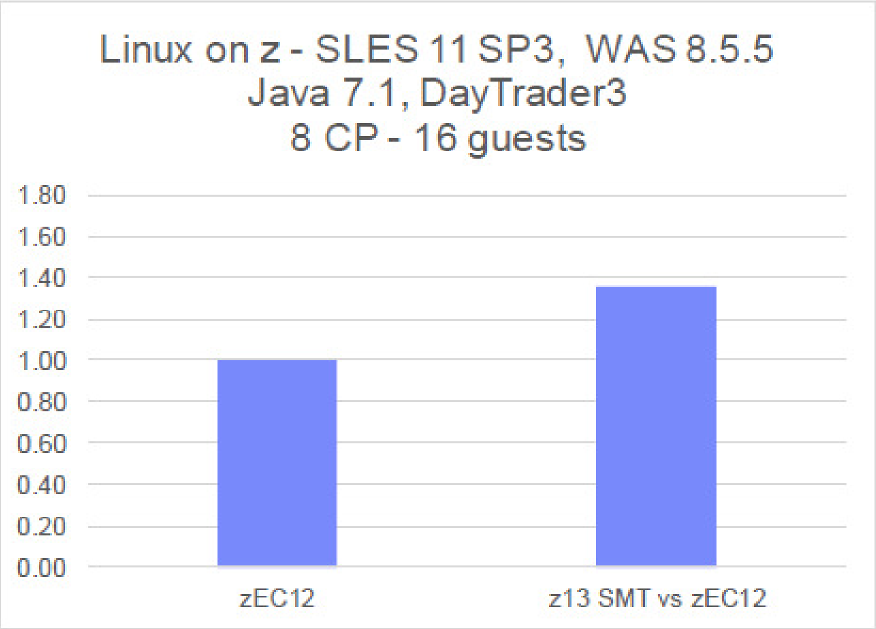
z13 - SSL Performance Improvements

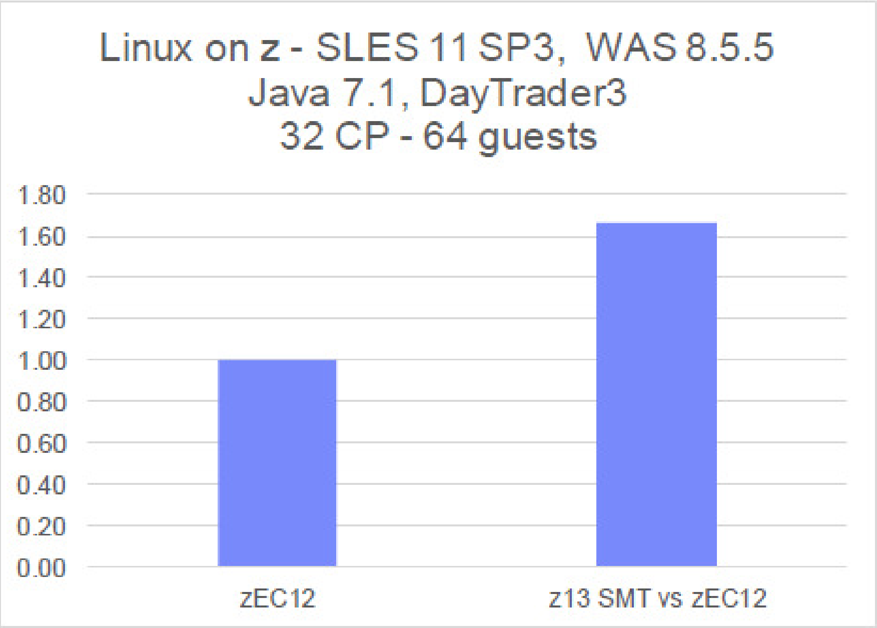


2.6x improvement in throughput with IBM Java 8 and IBM z13

z13 – WebSphere Performance Improvements

WebSphere – Linux on z Virtualized Cluster



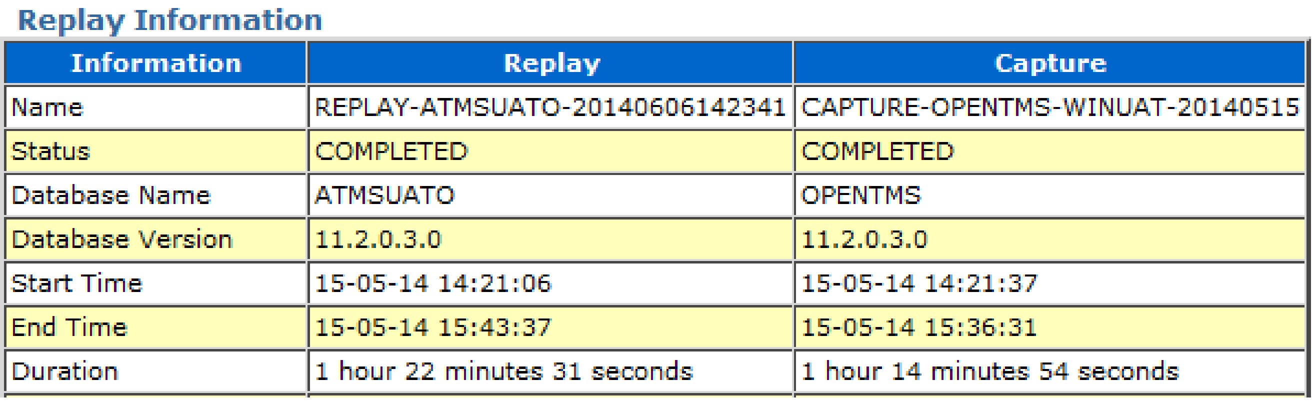


Between 1.36x to 1.66x improved throughput for a virtualized WAS cluster running DayTrader 3.0 on IBM z13 when compared to zEC12

Oracle Real Application Testing

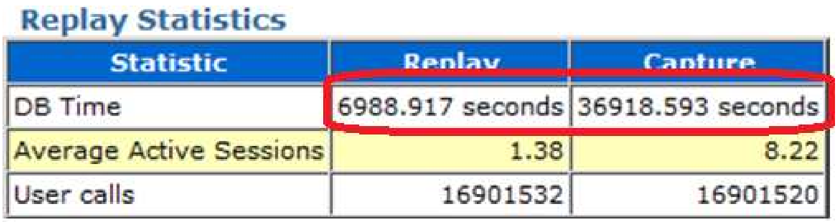
- **Helps resolve the Issue of having the Application Team setup (multiple hours) and test each test run.**

- **Allows the capability to “capture” a production or test workload and replay in test with various configurations.**



Replay Information

Information	Replay	Capture
Name	REPLAY-ATMSUATO-20140606142341	CAPTURE-OPENTMS-WINUAT-20140515
Status	COMPLETED	COMPLETED
Database Name	ATMSUATO	OPENTMS
Database Version	11.2.0.3.0	11.2.0.3.0
Start Time	15-05-14 14:21:06	15-05-14 14:21:37
End Time	15-05-14 15:43:37	15-05-14 15:36:31
Duration	1 hour 22 minutes 31 seconds	1 hour 14 minutes 54 seconds



Replay Statistics

Statistic	Replay	Capture
DB Time	6988.917 seconds	36918.593 seconds
Average Active Sessions	1.38	8.22
User calls	16901532	16901520

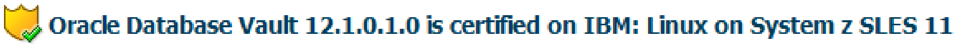
IBM z13 DEMO

Questions?

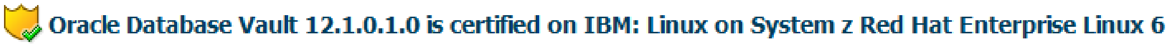
Oracle Database Vault & Audit Vault

Oracle Database Vault

Increase the security of existing applications and address regulatory mandates that call for separation of duties, least privilege, and other preventive controls to ensure data integrity and data privacy. Oracle Database Vault proactively protects application data stored in the Oracle database from being accessed by privileged database users.



✓ Oracle Database Vault 12.1.0.1.0 is certified on IBM: Linux on System z SLES 11



✓ Oracle Database Vault 12.1.0.1.0 is certified on IBM: Linux on System z Red Hat Enterprise Linux 6

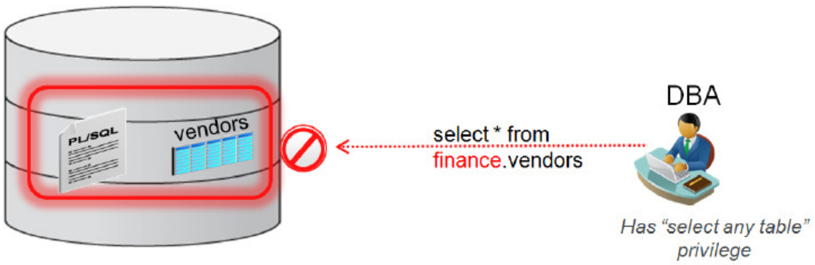


Figure 1. Oracle Database Vault Realms block access from privileged accounts

Oracle Audit Vault and Database Firewall

Monitor Oracle and non-Oracle database traffic to detect and block threats, as well as improve compliance reporting by consolidating audit data from databases, operating systems, directories, and other sources.

Master Note For Oracle Audit Vault (Doc ID 1199033.1)