ORACLE®

Oracle on Linux Best Practices

Saar Maoz

Consulting Performance Engineer

Doron Ofek

Linux and Open Source Technology leader

Oracle Week, Israel

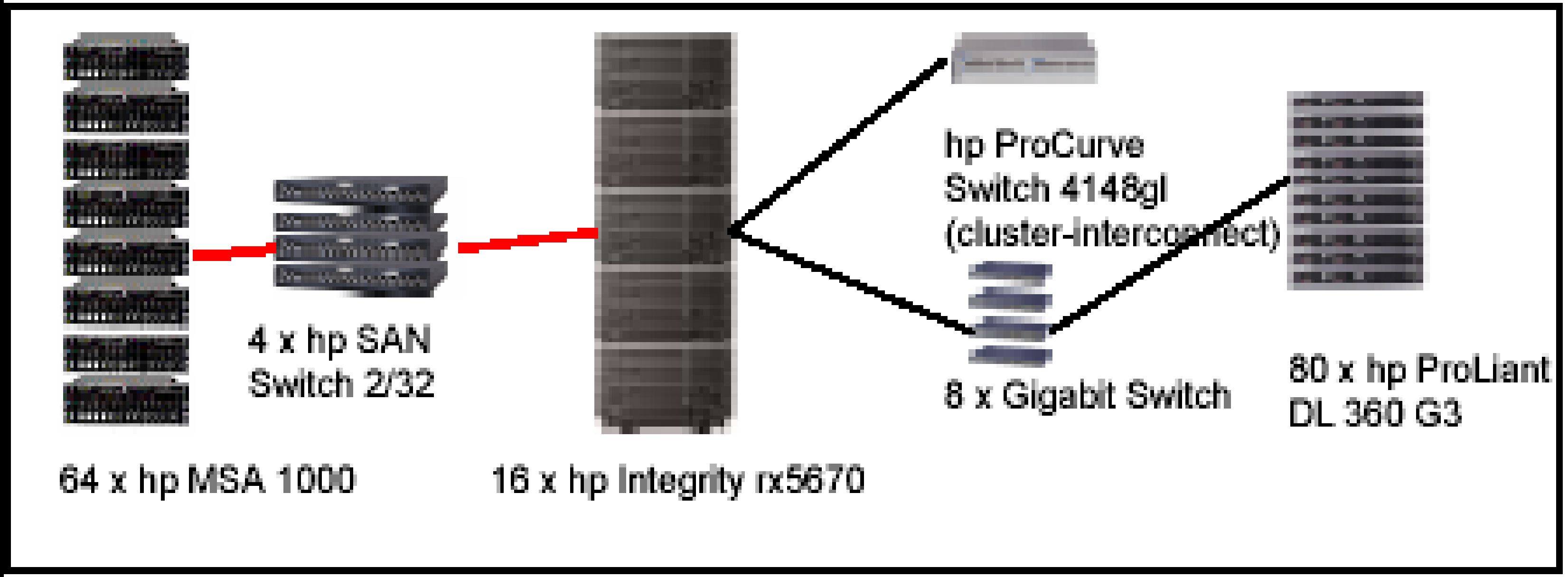
Nov, 2004

Oracle Corporation

16 Nodes 10g RAC TPC-C Bench.

1,184,893.38 tpmC @ $5.52/tpmC

TOP Overall TPC-C Result!



More details at: *http://www.tpc.org/*

Source - Transaction Processing Council, as of January 14, 2004: Oracle10g Database with Real Application Clusters on Red Hat Enterprise Linux AS 3, 16 HP Integrity rx5670, with 64 Intel Itanium 2 1.5 GHz processors, 1,184,893.38 tpmC, $5.52/tpmC, (original publication date, 12-8-03) available 04/30/04

WARNING

- We are not responsible for typographical errors, before using new commands please check:
  - Man pages:
    - `$ man <command>`
    - `$ man -k <keyword>` search for command
  - Info pages (more detailed, interactive)
    - `$ info <command>`
  - Command help, typically:
  - `$ rpm -h` or `--help`

Tuning Philosophy

- Philosophies differ
  - Tuning for new or existing database
  - Tend to start with things we know
  - Perception of a problem may sway your philosophy
- Here’s mine...
  - Go for the best bang for your buck
    - Translation: Go after the big things first

Choosing a Kernel

- Choose the appropriate kernel for your machine:
  - More than 4 GB of physical memory
    - RH: Use the enterprise kernel
    - UL: Use the 64GB kernel
  - More than one CPU
    - RH/UL: Use the SMP enabled kernel
  - Otherwise use the uniprocessor kernel
- Use the latest Update or ServPack from RH/UL
  - RHEL3 Update 3 has an I/O issue bugzilla 131391

Memory Configurations (IA-32)

- “smp”: 2-level page tables
  - HW can only address 4GB of RAM
  - 4kB pages and 4MB “large” pages
- “enterprise”: 3-level page tables
  - HW can address up to 64GB of RAM
  - 4kB pages and 2MB “large” pages
  - RHEL2.1: max 16GB RAM supported
  - SLES8, RHEL3: 64GB officially supported

Firmware / Drivers

- Verify and upgrade all needed drivers & Firmware prior to major testing:
  - BIOS for computer
  - Firmware for on board IO/Network card
  - Firmware for SAN/NAS storage
  - Linux IO/Network Controller driver

Red Hat EL 2.1 Install Notes

- Install development packages
  - Required for linking during Oracle install
- Use gcc-2.96 for Oracle 9i related relinking

UnitedLinux/SuSE Install Notes

- Install from UL CD #1 to get generic UL install
- Install from vendor’s CD to get vendor “flavor”
- Install the orarun package to simplify setup
  - Will check for package dependencies for Oracle
  - Create Oracle user and groups
  - Facilitate automatic startup/shutdown of all services
- Use gcc-2.95 for Oracle 9i related relinking

Choosing a Shell – Your Choice

- Your working shell is really a matter of choice:
  - Oracle scripts use Bourne (sh) & C (csh) shells
  - Most scripts specify the shell on first line as a comment
    - `#!/bin/sh`
- To show the currently used shell
  - `# echo $0`
- To change the default shell: `chsh` or `usermod`

RPM Package Manager

- Manage software packages
  - Install, upgrade, remove, verify, query, build
- Package files referred to as RPMs
  - Distributed by the vendor
  - Include files to be installed plus some install scripts
- Source RPMs contain the source code
  - e.g., kernel-2.4.9-e.24.src.rpm
- Binary RPMs contain the pre-built binaries
  - e.g., kernel-2.4.9-e.24.i686.rpm
  - Choose the highest architecture the machine can use
    - e.g., i686, i586, i486, i386 (**`uname -m`**)

RPM Queries

- To show the names of all packages installed:
  - `rpm -qa`
- To list the files in an RPM file:
  - `rpm -qlp <pkg_name>.rpm`
- To list the files in an installed package:
  - `rpm -ql <pkg_name>`
- To determine which package installed a file:
  - `rpm -q --whatprovides <filename>`

RPM Actions

- Only root can install/upgrade/remove RPMs
- Use `--test` option to see if action will work
  - Checks if all dependencies will be satisfied
  - If test succeeds, remove `--test` to really do it
- Specify multiple packages on one command line to satisfy circular dependencies
  - e.g., A.rpm requires B.rpm, B.rpm requires A.rpm
  - `rpm -i A.rpm B.rpm`

RPM Actions (cont'd)

- To install a package:
  - `rpm -i <pkg_name>.rpm`
- To upgrade a package:
  - `rpm -U <pkg_name>.rpm`
- To remove a package:
  - `rpm -e <pkg_name>`

Linux Updates

- Apply the recommended updates by the distribution vendor:
  - Most vendors provide automatic updates
    - Red Hat Network supplies updates automatically
      - `up2date`
    - SuSE uses
      - `YaST` (Yet Another Setup Tool)
      - `YOU` (YaST Online Update)

Kernel “sysctl” Parameters

- Control dynamic kernel configuration/tuning
  - Most parameters can be changed on the fly!
- Can be set multiple ways:
  - In /etc/sysctl.conf: (Recommended)
    ```
    fs.aio-max-size=1048576
    ```
  - In /etc/rc.local (RH) or /etc/boot.local (UL):
    ```
    echo 1048576 > /proc/sys/fs/aio-max-size
    ```
  - Using sysctl:
    ```
    sysctl -w fs.aio-max-size=1048576
    ```
- Will be lost on reboot if not in /etc/sysctl.conf!

Kernel "sysctl" Parameters

- aio-max-size = 1048576 (1MB)
- aio-max-nr = 65536 (Leave default)
- shmmax = 3294967296 (3GB)
- shmall = 4194303 (4GB -1)
- If using VLM: shm-use-bigpages = 2

Boot Loaders

- Grub (recommended)
  - New and improved boot loader
  - Configured via `/boot/grub/menu.lst`
  - Not necessary to rerun after config changes
- LILO
  - Original Linux boot loader
  - Configured via /etc/lilo.conf
  - Must run /sbin/lilo after any change to lilo.conf or binary images (/boot/*)

Grub Configuration

```
default=0
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Ent Linux AS (2.4.21-9.EL)
 root (hd0,0)
 kernel /boot/vmlinuz-2.4.21-9.EL root=LABEL=/
 initrd /boot/initrd-2.4.21-9.EL.img
```

Loadable Kernel Modules

- To install a module, two methods:
  - `insmod <module name>.o`
  - `modprobe <module name>`
- To list the installed modules:
  - `lsmod`
  - `cat /proc/modules`
- To remove a module:
  - `rmmod <module name>`
- Do NOT recompile a module into kernel

Checking for a Tainted Kernel

- Use the **`/sbin/lsmod`** command to see whether the kernel is tainted:

```
# /sbin/lsmod
Module                Size            Used by     Not
tainted
nfs                    87936      0  (autoclean)
lockd                  60224      0  (autoclean)  [nfs]
sunrpc                 79952      0  (autoclean)  [nfs lockd]

iptable_filter          2912      0  (autoclean)  (unused)
ip_tables              14080      1  [iptable_filter]
ad1848                 23968      0  [cs4232]
ext3                   70240      5
```

Creating Partitions

- Linux fdisk can create partitions of any type
  - Partition type 83 - “Linux”
    - Linux –specific filesystems
    - Oracle Cluster Filesystem (OCFS)
    - Raw devices
  - Partition type 82 - “Linux swap”
- At most 4 primary partitions
- Additional partitions must be logical partitions inside an extended partition

Listing Partitions

```
# cat /proc/partitions
major minor #blocks name rio rmerge …
   8      0    8281507 sda  …
   8      1    8281476 sda1 …
   8     16    4192965 sdb  …
   8     17    2048256 sdb1 …
   8     18    2048287 sdb2 …
   8     32    9430155 sdc  …
   8     33    9430123 sdc1 …
   8     48    9430155 sdd  …
   8     49    4715046 sdd1 …
   8     50    4715077 sdd2 …
  22      0     252290 hdc  …
```

Creating Filesystems

- Create partitions of type “Linux” (type id=83)
- Use `/sbin/mkfs` to format the partition
  - Must give **`-t`** to specify filesystem type
  - OR, use `/sbin/mkfs.<fstype>` instead
- Example:
  - **Ext2:** `# mkfs -t ext2 /dev/sdb1`
  - **Ext3:** `# mkfs -j /dev/sdb1`

Mounting Filesystems

- For one-time or ad hoc mounts, mount manually
  - Use `/bin/mount` to mount, `umount` to un-mount
For frequently used filesystems, use `/etc/fstab`
  - Entries can be mounted automatically or not
- Unmounting a filesystem flushes and releases all buffers for it from the pagecache

One-time Mounts

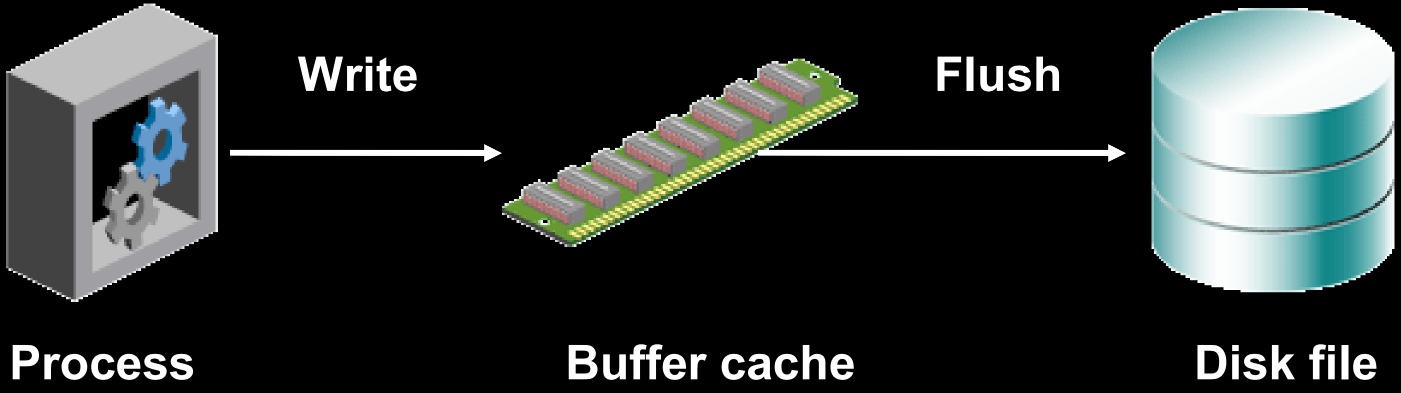
- Most filesystem types recognized automatically
  - Exception: OCFS
  - Use `-t` option to force correct filesystem type
- Example:
  - `/bin/mount -t ocfs /dev/sdb1 /oracle`
- where:
  - /dev/sdb1 is the device name of the partition
  - /oracle is the mountpoint where it will be mounted

Preconfigured Mounts

- Put commonly used entries in /etc/fstab
- Example:
  - `/dev/sdb1 /oracle ext3 defaults 1 2`
- To mount:
  - `mount /oracle`
  - `mount /dev/sdb1`
- Use `noauto` mount option to prevent an entry from being mounted automatically at boot time
  - e.g., /mnt/cdrom, /mnt/floppy

I/O Modes

- Disk I/O can be performed in several different modes
  - Asynchronous vs. synchronous
  - Direct vs. buffered



ext2 / ext3 / reiserfs

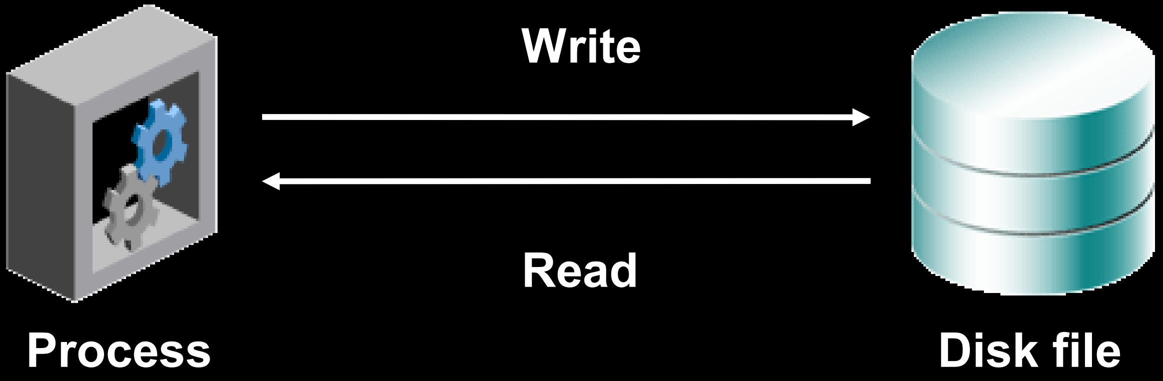
- ext2 used to be the most common Linux filesystem
- ext3 is based on ext2
  - Same on-disk structure
  - ext2 can be converted to ext3
  - ext3 can be mounted as an ext2 file system
- reiserfs/ext3 are both journaling filesystems
  - Preserves data integrity better than ext2
  - Faster and safer fsck after system crash
- Mount with `noatime` option.

Oracle Clustered Filesystem

- OCFS delivers raw-like performance with tremendous advantages in management and usability
  - All database related files reside on a clustered filesystem
  - Visible to all nodes
- Open source project sponsored by Oracle
- Supports asynchronous I/O as of v1.0.9
- Free to use
- Get fix for bug #2883583
  - Allows cp, dd & tar of open database files needed for hot backup

Raw Devices

- I/O directly to partitions instead of a filesystem
- Eliminates copying to/from filesystem cache
- Each raw device is a character device
  - Character device major number 162
- Requires more experienced administration



Raw Devices (cont'd.)

- Devices will be either in /dev or /dev/raw
- Device names are raw1 – 255 by convention
  - May need to manually create raw129 – raw255:
    - e.g., `mknod /dev/raw/raw129 c 162 129`
- Minor #0 is special and can't be used for I/O
  - Implies a limit of 255 raw devices
- DB files should be symlinks to /dev/raw/raw*

Creating Raw Devices

- Create partitions of type “Linux” (type id=83)
- Use /sbin/raw to bind raw device with a partition
- Examples:

```
– # /sbin/raw /dev/raw/raw1 /dev/sdc1
– # /sbin/raw /dev/raw/raw1 8 33
```

- Give oracle user ownership of raw device:

```
– # chown oracle:dba /dev/raw/raw1
```

- Bindings are not persistent across reboot
  - RedHat: /etc/sysconfig/rawdevices
  - SuSE: /etc/raw

Asynchronous I/O

- Oracle10g & 9i are shipped with asynchronous I/O support disabled (not using libaio)
- It’s easy to enable (re-link Oracle) and the benefits are immediate on intensive write databases
- 9i see bug 3016968, 10g see bug 3438751
- See Note: 225751.1 on enabling

Asynchronous I/O - Enabling

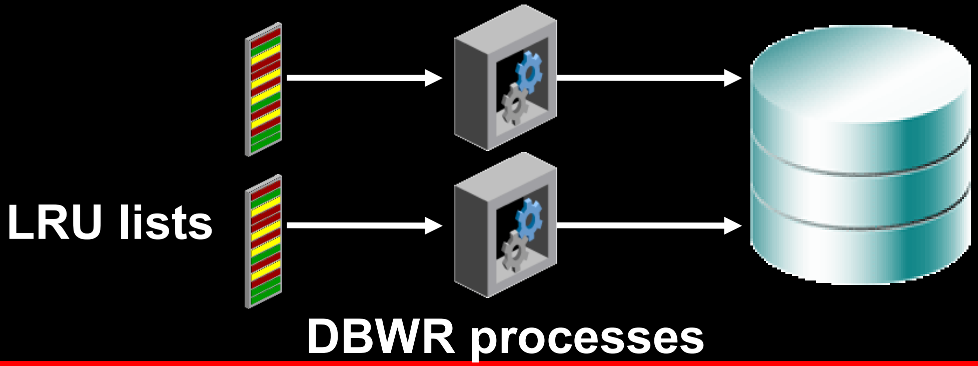
- Must re-link to use libaio:
  - `$ make -f ins_rdbms.mk async_on ioracle`
- init.ora:
  - `disk_asynch_io=true` (the default)
  - `filesystemio_options=asynch` set this as well if datafiles are on a filesystem (e.g. ext3 or OCFS)
- Using multiple DBWRs with async I/O is usually better than using I/O slaves

Asynchronous I/O - Using

- 2 DBWRs appears to be a good default for a large buffer cache
  - `db_writer_processes=2`
  - No need for I/O slaves (Emulate A-I/O)
    - `#dbwr_io_slaves=20`
- If large read sizes occur, set:
  - `fs.aio-max-size` to the largest read size (default is 128KB)
  - Oracle maximum I/O size is 1MB

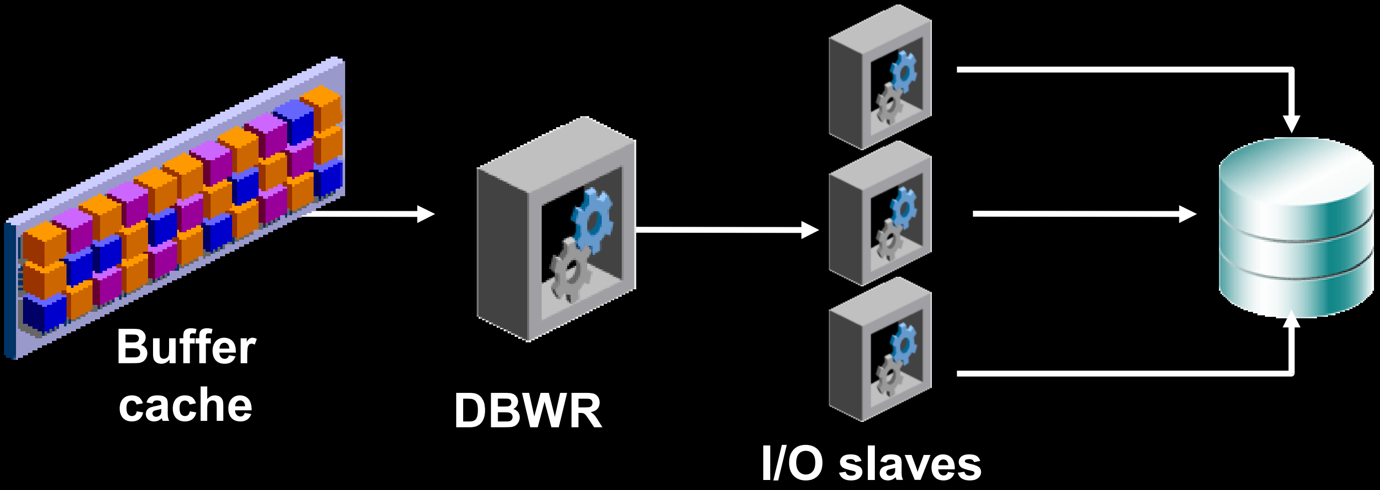
Multiple DBWR Processes

- DB_WRITER_PROCESSES (new)
  - Set to no more than CPU_COUNT, up to 20
  - Multiple DBWR processes write from LRU to disk
  - These processes can use asynchronous I/O
  - These processes are best used in OLTP environments



DB Writer Slaves

- DBWR_IO_SLAVES (old)
  - Used to simulate asynchronous I/O
  - One DBWR, multiple writers to disk



/proc Filesystem

- /proc is a virtual file system
- Provides an instantaneous view of the operation of the system
  - /proc/meminfo, /proc/mounts, /proc/partitions
  - Can be viewed with cat, more, less
- Can be used to configure kernel parameters
  - Settable parameters are below /proc/sys
  - Can be set with **`echo`** or via **`sysctl`**

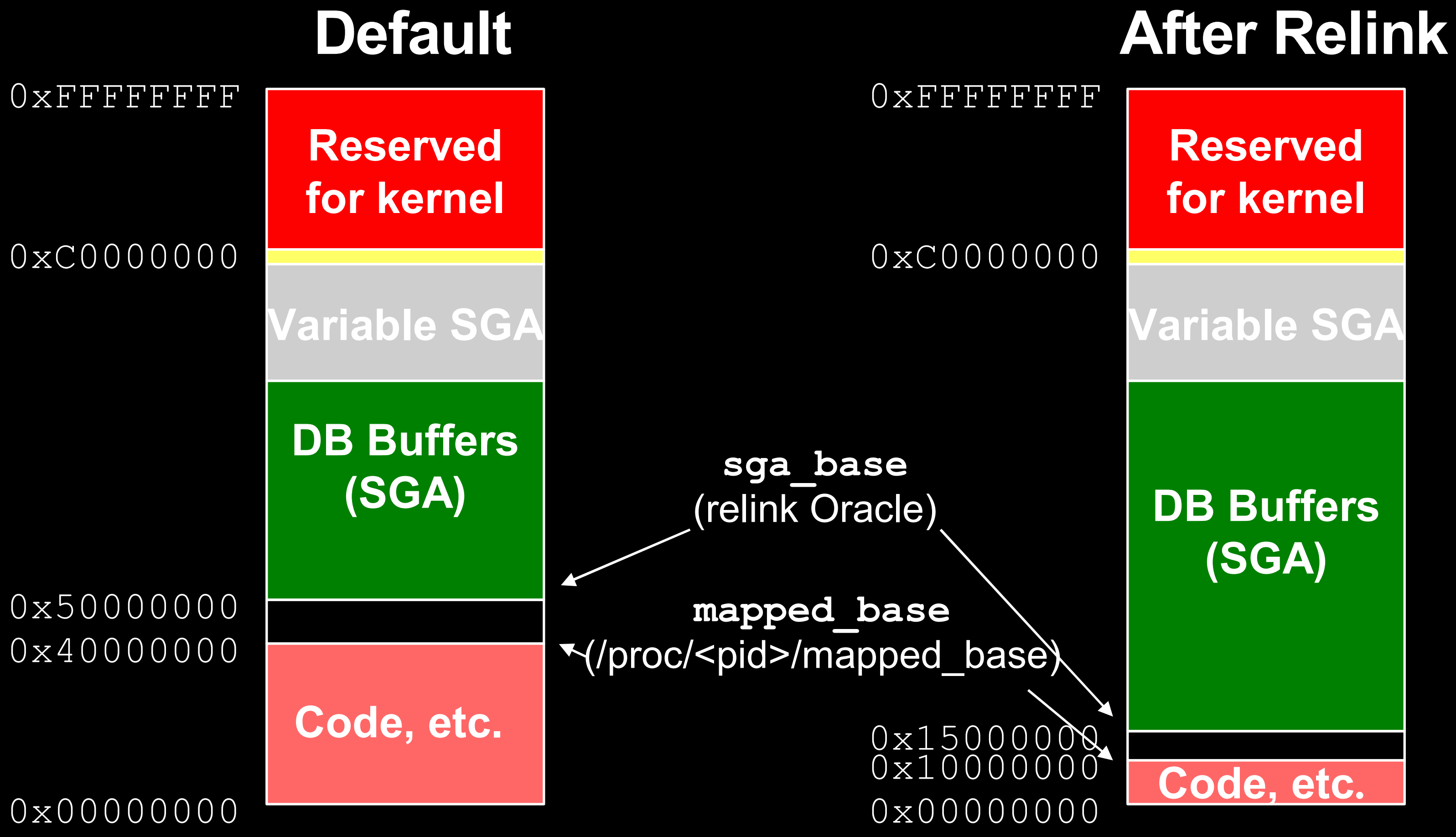
System V Shared Memory

- Used by Oracle for the SGA
  - May have multiple segments if shmmax is low
  - Normally all segments deleted at shutdown
  - If instance crashes, segments may hang around
- To view existing segments:
  - `/usr/bin/ipcs`
- To manually remove a segment:
  - `/usr/bin/ipcrm shm <shm_id>`

SGA Larger than 1.7 GB (IA-32)

- On Red Hat 2.1 & SuSE SLES8 & SLES9
  - Due to default layout of a process’ address space Oracle can attain an SGA of only about 1.7GB
- It’s possible to rearrange user-mode address space to accommodate a larger SGA of about 2.7GB
  - Metalink Note 200266.1
- RHEL3 allows an SGA size of about 3.6GB
  - No special modifications needed

SGA Larger than 1.7 GB (cont'd)



SGA Larger Than 1.7 GB (cont'd)

- These steps are required for implementing an SGA between 1.7 GB and 2.7 GB
  1. Modify shmmax
  2. Modify a shell for starting the database instance by lowering mapped_base
  3. Relocate the SGA:
     a. Modify ksms.s
     b. Relink the database executables

Modifying shmmax

- Set the shmmax parameter to hold the entire SGA in a single shared memory segment
  - This can be done for real memory up to 4 GB
  - For this boot:

```
# echo 3000000000 > /proc/sys/kernel/shmmax
```

  - For future boot ups change sysctl.conf:

```
kernel.shmmax = 3000000000
```

Lowering the mapped_base

- The mapped_base parameter is set at the process level, and can only be set by root
  - Start a shell as the oracle user
    - Find the oracle user shell process ID:

```
$ echo $$
```

  - Start a shell as the root user
    - Set the mapped_base for the oracle user shell:

```
# echo 268435456 > /proc/<userpid>/mapped_base
```

  - In the oracle user’s shell:
    - Start the database
    - Start the listener

Relocating the SGA

- Execute the following commands as the Oracle software owner:

```
$ cd $ORACLE_HOME/lib
$ cp -a libserver9.a   libserver9.a.BCK.orig

$ cd $ORACLE_HOME/rdbms/lib
$ cp ksms.s ksms.s_orig   /* if ksms.s exists,
back it up first.*/
$ genksms -s 0x15000000 > ksms.s

$ make -f ins_rdbms.mk ksms.o
$ make -f ins_rdbms.mk ioracle
```

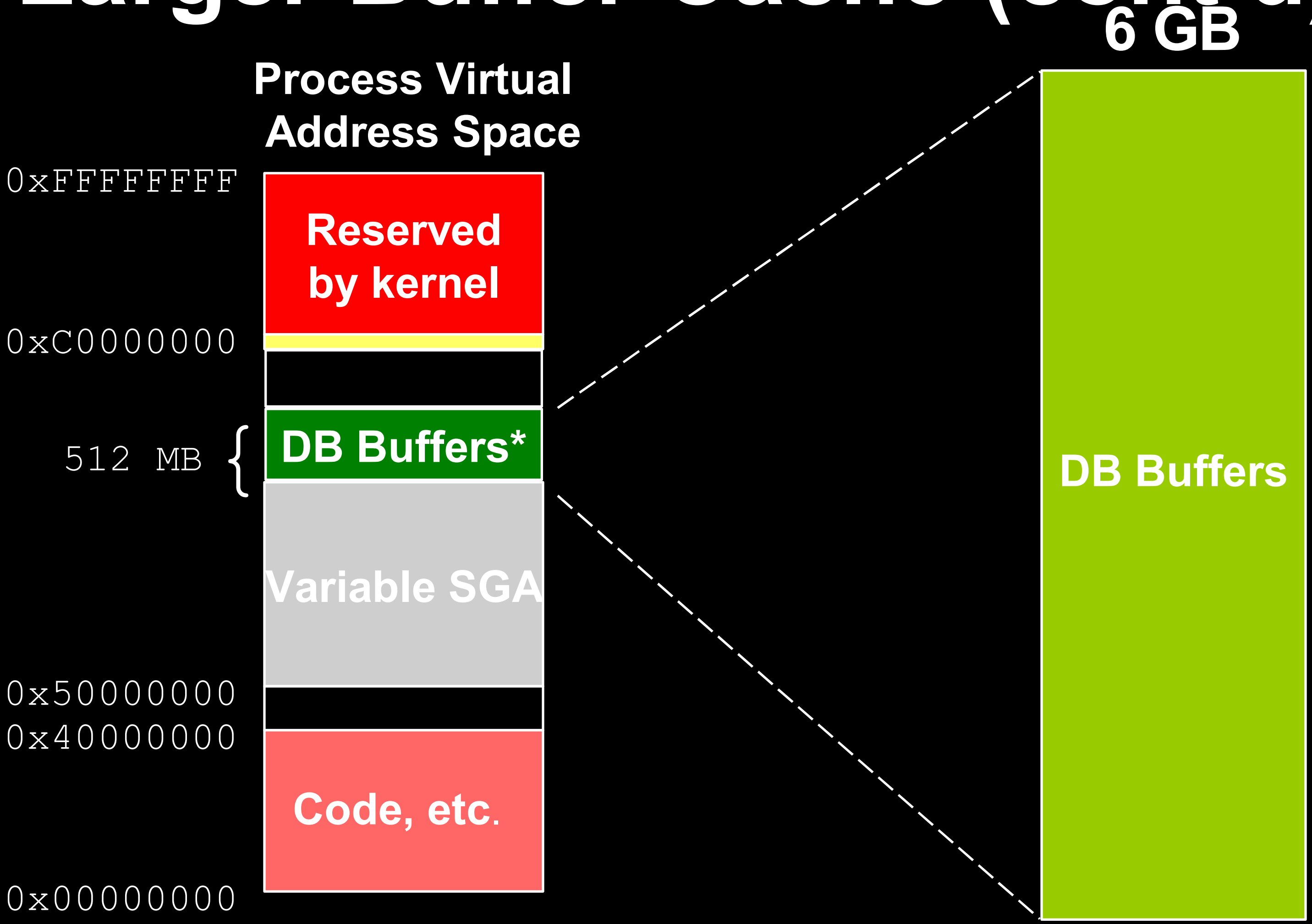
SGA Larger than 1.7 GB (cont'd)

- Once the Oracle executable has been relinked with a lowered SGA base, all connected users must have a lowered mapped_base
  - Listener takes care of remote connections
- See Metalink Note 200266.1 for details on what can go wrong and for a sample program that can automatically lower mapped_base
  - Lowered base address propagates to child processes
  - Start the listener from a lowered base address shell

Larger Buffer Cache (IA-32)

- Oracle has the capability to use an extended buffer cache greater than 4GB
- Using Indirect Data Buffers has some overhead, in most cases the benefit is worth it
  - Reduced I/O rate since more data is cached in memory
- Additional things to note:
  - Requires enterprise kernel
  - Dynamic SGA parameters cannot be used in this case
  - Enable bigpages for further performance boost

Larger Buffer Cache (cont'd)



Larger Buffer Cache (cont'd)

- Steps to enable Indirect Data Buffers (from Oracle9i Administrators Reference, Rel 2 for Linux):
  1. Set shmmax to hold the entire SGA
  2. Enable shared memory filesystem:
     - `# mount -t shm -o size=8g shmfs /dev/shm`
     - In fstab: `none /dev/shm tmpfs size=8g 0 0`
  3. Set the initialization parameters:
     - `use_indirect_data_buffers=true`
     - use only `db_block_buffers` and `db_block_size` (cannot use `db_cache_size`)

Bigpages (IA-32)

- Separate memory pool using large hw pages
- Non-swappable
- Must be set aside at boot time
  - Boot with kernel parameter “bigpages=8192MB”

- **Use Workaround in bug 3080838**
  - To prevent Kernel panic in sshd_config set:
    - UsePrivilegeSeparation no   -OR-
    - Compression no   (preferred)

Bigpages (cont’d)

- To enable/reserve large contiguous pages on next reboot, add to Linux boot parameter:
  - Grub: `bigpages=<size>MB`
  - Lilo.conf: `append=“bigpages=<size>MB”`
- To configure Oracle to allocate the SGA from the bigpages pool set in /etc/sysctl.conf:
  - **`Kernel.shm-use-bigpages=2`**
- Look in `/proc/meminfo` at `BigPagesFree` to see the size of the bigpage pool

Bigpages (cont'd)

- This should be used all the time once you know Oracle's memory requirements
- The bigpage pool is only available for shared memory, so the system may swap if there is not enough memory left to satisfy per-user memory needs
- The bigpage pool needs to be slightly larger than your SGA or startup will fail

Parameter: shm-use-bigpages

- Allowed values:
  - 0: don’t use bigpages pool for shared memory
  - 1: use bigpages pool for SysV shared memory
  - 2: use bigpages pool for SysV and shmfs
- Default value: 0

- **Use Workaround in bug 3080838**
  - To prevent Kernel panic in sshd_config set:
    - UsePrivilegeSeparation no -OR-
    - Compresson no (preferred)

Hugetlbfs (RHEL3&4, SLES9)

- Similar to bigpages but, configured at runtime, no need to reboot:
  - Shutdown instance (free memory)
  - Ask for 6000MB:
    - By MB: `echo 6000 > /proc/sys/vm/hugetlb_pool`
    - # Pages: `echo 1500 > /proc/sys/vm/nr_hugepages`
    - Page size: `cat /proc/meminfo|grep Hugepagesize`
  - `cat /proc/meminfo`
    - Verify you got 6000MB, if not might need to reboot
  - Startup instance
    - Verify usage in `/proc/meminfo`

Network configurations

- Increase SDU/TDU in SQL*Net Note: 44694.1
  - Both in tnsnames.ora & listener.ora:
    - `(SDU = 8192) (TDU = 32767)`
  - Will reduce (V$SYSTEM_EVENT)
    - SQL*Net more data to client
    - SQL*Net more data from client
- tcp.nodelay in protocol.ora, Note: 1005123.6
  - `tcp.nodelay=yes`
  - Consume more bandwidth but more responsive

Process Specific

- Specific process is a suspect:
  - System call trace:
    - `strace -p <pid>`
  - Library call trace:
    - `ltrace -p <pid>`
  - Detailed process statistics:
    - `ps -o <options>`
    - **Try:** `ps -e -o pid,ppid,pcpu,rss,vsz,pri,wchan,cmd`
- Who has my file open?
  - `lsof [-p <pid]`
- For Process Tree, use `pstree -p`
- Not seeing a process, it's probably a thread, try: `ps -efm`

Installing and Configuring Statspack

- Install Statspack
  - $ORACLE_HOME/rdbms/admin/spcreate.sql
- Collect statistics:
  - Execute statspack.snap;
- Produce a report
  - @spreport.sql
- Collect timing information, set STATISTICS_LEVEL = TYPICAL.

Correlating Database and OS Measurements

- Correlate I/O timing reported by Oracle to I/O timing reported by OS utilities & Hardware
  - For example: Database says I/O takes 60ms but hardware says 10ms
- OS and database statistics should be collected at the same time periods to have a meaningful comparison
  - Run `sar` and Statspack continuously

DB Tuning in Oracle 10g

- Automatic Workload Repository (AWR)
  - All statistics are collected every 60 minutes
    - If `STATISTICS_LEVEL` set to ALL or TYPICAL
  - Built into the Oracle Kernel (no SQL used)
  - HTML based statspack like reports possible
    - Obsoletes statspack
- Automatic Database Diagnostic Monitor (ADDM)
  - Produce actual implementation suggestions based on AWR snapshots

Summary: Linux Monitoring Tools

- Overall tools
  - `sar , vmstat`
- CPU
  - `/proc/cpuinfo , mpstat , top`
- Memory
  - `/proc/meminfo , /proc/slabinfo`
- Disk I/O
  - `iostat, sar`
- Network
  - `iptraf, netstat, mii-tool`
- Individual process debugging
  - `strace , ltrace, lsof`

Linux Tuning Summary

- Take advantage of easy Linux improvements:
  - IA-32:
    - Increase user address space to fit a larger SGA
    - Use Large Buffer Cache when needed and possible
  - Use bigpages/hugetlbfs to map SGA more efficiently
  - Use Asynchronous I/O
- Maintain up to date Linux environment

Unbreakable
ORACLE

QUESTIONS ANSWERS

ORACLE®