

Administración Avanzada de Oracle 9i.

Este documento electrónico puede ser descargado libre y gratuitamente desde Internet para su ejecución e impresión, sólo para fines educativos y/o personales, respetando su integridad y manteniendo los créditos de los autores en el pie de página.
Queda por tanto prohibida su venta.

Francisco Fernández Martínez (pacof@um.es)
Juan Luis Serradilla Amarilla (juanlu@um.es)
Universidad de Murcia

TEMARIO

- Instalación de Oracle9i.
- Diseño y creación de una base de datos.
- Estructuras de almacenamiento (bloque, fila, segmento).
- Ajuste y monitorización de la instancia.
- Soporte de globalización (NLS).
- Net9.
- MTS.
- Configuración del archivado.
- Backup.
- Recovery.

OBJETIVOS

- Saber instalar Oracle9i (servidor, cliente y BD) sobre Linux.
- Diseñar (memoria, CPU y disco) y crear una base de datos.
- Conocer el formato del bloque oracle, estructura de la fila, tipos de segmentos y tipos de índices (B-tree vs IOT).
- Monitorizar la instancia con las vistas dinámicas de rendimiento. Ajustar sharedpool, buffer cache, redo log, rollback, temporal (sort), checkpoints, sql, cpu. Utilidad Statspack.
- Adaptar una BD Oracle9i a un lenguaje y territorio (fecha, etc).
- Configurar Net9 en el servidor y en el cliente.
- Saber activar y configurar MTS.
- Configurar el archivado de una BD Oracle9i.
- Conocer las cuestiones avanzadas del backup de una BD.
- Conocer las cuestiones avanzadas del recovery de una BD.

TEMA 1

INSTALACIÓN DE ORACLE9i

TEMA 1.

INSTALACIÓN DE ORACLE9i

- Productos instalables.
- Tipos de Bases de Datos.
- Asistentes de Configuración.
- Requerimientos HW y SW.
- Parámetros del Kernel LINUX.
- Precauciones con Fedora Core 2 Linux.
- Preinstalación.
- Instalación.
- Postinstalación.

PRODUCTOS INSTALABLES

- Oracle9i Database
 - Enterprise Edition: BD preconfigurada, sqlnet, opciones con licencia, herramientas de configuración, OEM (OMS, IA, Consola y packs con licencia), Oracle Utilities y documentación.
 - Standard Edition: BD preconfigurada, sqlnet, OEM (OMS, Intelligent Agent, Consola y packs standard), Oracle Utilities (dbverify, exp, imp, etc)..
 - Custom: seleccionar componentes de la opción "Enterprise Edition".
- Oracle9i Client
 - Administrator: OEM (consola y packs con licencia), sqlnet, utilidades, software cliente , OLAP API (datawarehouse) y documentación.
 - Runtime: sqlnet y ficheros de soporte.
 - Custom: personalizada.
- Oracle9i Management and Integration
 - Oracle Management Server: OMS, OEM, software cliente y documentación.
 - Oracle Internet Directory: LDAP OID Server, herramientas cliente LDAP y esquema de BD para OID.
 - Custom: personalizada.

TIPO DE BASE DE DATOS

- Propósito General: transacciones cortas (OLTP) y también largas (procesos batch). La BD creada incluye:
 - Parámetros de inicialización usando spfile y gestión automática de UNDO.
 - Oracle Options e interMedia
 - Advanced Replication
 - Servidores Dedicados (no usa MTS)
 - NOARCHIVELOG
- OLTP (Online Transaction Processing): muchas transacciones cortas concurrentes, consistentes en consultas sencillas sobre pocos datos (también actualizaciones).
- Data Warehouse: consultas complejas sobre muchos datos (OnLine Analytical Processing).
- Personalizada.
- Sólo Software: no crea BD ni configura sqlnet.

ASISTENTES DE CONFIGURACIÓN

- Database Configuration Assistant (dbca).
 - Permite copiar una BD preconfigurada o crear una personalizada.
 - Arranca automáticamente una vez que se ha instalado el sw Oracle.
- Oracle Net Configuration Assistant (netca): configura la red C/S de Oracle en \$ORACLE_HOME/network/admin.
 - Opciones Enterprise y Standard: configura listener.ora, sqlnet.ora y tnsnames.ora.
 - Personalizada: permite configurar un servicio de directorio (LDAP), crear listeners para conectar a la BD y seleccionar el método de resolución de nombres. Configura, además de los anteriores, el fichero ldap.ora.
 - Cliente: permite configurar tnsnames.ora, sqlnet.ora y ldap.ora.
- Database Upgrade Assistant (dbua): para actualizar una BD \geq 8.0.6/8.1.5, después de instalar el sw de Oracle9i.
- Oracle Enterprise Manager Configuration Assistant: configura OMS y crea (o actualiza) su repositorio (necesita BD).
- Oracle Internet Directory Configuration Assistant: configura OID, arrancando OID Server (necesita BD).

REQUERIMIENTOS HW Y SW (Linux)

- HW:
 - RAM 512M (256M para el cliente): "grep Mem /proc/meminfo"
 - HD 3Gb para el sw, 1Gb para la bd y al menos 400Mb de /tmp: "df"
 - Swap igual a la RAM ó 1Gb (el mayor): "/sbin/swapon -s"
- SW:
 - Linux kernel \geq 2.4.7: "uname -a"
 - Gcc \geq 2.2.2 (máximo recomendable 2.96): "gcc -v"
 - Entorno gráfico (Xwin): "xclock"
 - Acceso a los ejecutables: make, ar, ld, nm.
 - Parámetros del kernel (directorio /proc/sys/kernel)

PARÁMETROS DEL KERNEL

Valores mínimos para una BD (Linux)

- SEMMNI = 100 (# máximo de conjuntos de semáforos).
- SEMMNS = 256 (# máximo de semáforos, sólo para la instalación inicial).
 - SEMMNS = $\text{sum}(\text{processes}) + \text{max}(\text{processes}) + 10 * \text{count}(\text{BBDD})$
processes es el parámetro del init de cada BD.
- SEMOPM = 100 (# máximo de operaciones por "semop call").
- SEMMSL = 100 (valor mínimo recomendado, sólo para la instalación inicial).
- SHMMAX = 2147483648 (tamaño máximo del segmento de memoria compartida).
 - 2 GB para kernel SMP. Se recomienda la mitad de la RAM.
- SHMMIN = 1 (tamaño mínimo de un segmento de memoria compartida).
- SHMMNI = 100 (# máximo de segmentos de memoria compartida).
- SHMSEG = 4096 (# máximo de segmentos de memoria por proceso).
- SHMVMX = 32767 (valor máximo de un semáforo).

PARÁMETROS DEL KERNEL

Ejemplo de asignación (Linux)

```
cd /proc/sys/kernel
echo SEMMSL_value SEMMNS_value SEMOPM_value SEMMNI_value > sem
echo "250 32000 100 128" > sem
echo 1073741824 > shmmax
echo 4096 > shmmni
echo 65536 > /proc/sys/fs/file-max (File Handles)
ulimit -n 65536 (File Handles)
echo 1024 65000 > /proc/sys/net/ipv4/ip_local_port_change (sockets)
ulimit -u 16384 (# máximo de procesos por usuario)
```

FEDORA CORE 2 Linux

- LD_ASSUME_KERNEL=2.4.1 (usuario oracle, antes de instalar)
- Para poder instalar hay que seguir las instrucciones para "Red Hat Enterprise Linux 3" y así evitar los dos errores siguientes:
 - " ... error while loading shared libraries: libstdc++.i386-1.so.2 ..."
 - # yum install compat-libstdc++.i386
 - # yum install compat-gcc.i386 compat-gcc-c++.i386
 - # mv /usr/bin/gcc /usr/bin/gcc333
 - # mv /usr/bin/g++ /usr/bin/g++333
 - # ln -s /usr/bin/gcc296 /usr/bin/gcc
 - # ln -s /usr/bin/g++296 /usr/bin/g++
 - "... __libc_wait, version GLIBC_2.0 not defined in file libc.so.6 ..."
 - Aplicar parche 3006854 (descargarlo de Metalink y ejecutar como root).

PREINSTALACIÓN (Linux)

- Crear puntos de montaje: uno para el sw y hasta seis para las bases de datos:
 - /u01 (sw)
 - /u02 (datos), /u03 (índices), /u04 (redo), /u05 (system), /u06 (temp) y /u07 (rollback)
- Crear grupo dba: "groupadd dba".
- Crear grupo oinstall: "groupadd oinstall".
- Crear usuario oracle con grupo primario dba y secundario oinstall: "useradd -g dba -G oinstall oracle".
- Crear directorio /u01/app/oracle y /u02/oradata, /u03/oradata, etc.
- Variables de entorno del usuario oracle justo antes de instalar:
 - DISPLAY=workstation_name:0.0 (servidor Xwin)
 - ORACLE_BASE=/u01/app/oracle (el resto es opcional)
 - PATH debe incluir \$ORACLE_HOME/bin, /usr/bin, /bin, /usr/bin/X11/ y /usr/local/bin
 - ORACLE_HOME=\$ORACLE_BASE/product/9.2.0.1.0
 - ORACLE_SID

INSTALACIÓN

- Montar CD desde el root:
 - # mount /mnt/cdrom
- Como usuario oracle, lanzar script de instalación desde fuera del punto de montaje del CD (por ejemplo, desde el HOME del usuario oracle).
 - Comprobar DISPLAY, LD_ASSUME_KERNEL y ORACLE_BASE.
 - Verificar que /usr/bin/gcc apunta a gcc296.
 - \$ /mnt/cdrom/runInstaller
 - Nota. Se podría ejecutar en modo "no interactivo" con:
\$ /mnt/cdrom/runInstaller -responsefile mirespfile -silent
 - Cambiar CD: "eject" o "umount /mnt/cdrom" y luego "mount /mnt/cdrom".
 - Si da "Relink error for ins_oemagent.mk", ignorar y completar la instalación. Después, aplicar parche para Bug 3119415.
 - Si da "ctx relinking error in ins_ctx.mk", ignorar. Se solucionará aplicando el patchset 9.2.0.4.

POSTINSTALACIÓN

- Incluir variables en `.bash_profile` del usuario oracle:
 - `export ORACLE_BASE=/u01/app/oracle`
 - `export ORACLE_HOME=$ORACLE_BASE/product/9.2.0.1.0`
 - `export LD_LIBRARY_PATH=$ORACLE_HOME/lib`
 - `export PATH=$ORACLE_HOME/bin:$PATH`
 - `export ORACLE_SID=mibd`
- Activar E/S asíncrona. Ejecutar el siguiente comando de `$ORACLE_HOME/rdbms/lib`:
 - `make -f ins_rdbms.mk asynch_on`
- Comprobar seguridad de la BD:
http://otn.oracle.com/deploy/security/oracle9i/pdf/9iR2_checklist.pdf
- Comprobar parches: al escribir ésto estaba disponible el de la 9.2.0.6.

REVISION DE SEGURIDAD

- Instalar sólo aquellas opciones que sean necesarias (Java, Intermedia, etc).
- Bloquear cuentas de usuarios creados por defecto que no vayamos a utilizar. DBCA bloquea todas menos SYS, SYSTEM, SCOTT y DBSNMP, por ejemplo: outln, mdsys, wksys, ctxsys, ordsys, etc.
- Cambiar claves de usuarios creados por defecto: SYS, SYSTEM, etc.
- Proteger el DD con `O7_DICTIONARY_ACCESSIBILITY = FALSE`, impidiendo acceso al DD a través de privilegios ANY. Así, por ejemplo, usuarios con DROP ANY no podrán borrar el DD
- Practicar el principio de “los privilegios justos”. Por ejemplo, para conectar a la BD dar CREATE SESSION y no CONNECT.
- Restringir los usuarios unix con acceso al servidor Oracle.
- Restringir el acceso al servidor Oracle a través de y desde la red.
- Aplicar todos los parches de seguridad que vayan saliendo.
 - <http://otn.oracle.com/deploy/security/alerts.html>

COMPROBACIÓN FINAL

- En el directorio \$ORACLE_BASE/oraInventory/logs podemos revisar lo ocurrido durante la instalación.
- Comprobar que el SW y la BD (si hemos instalado una) funcionan.
 - Conectar a la BD usando Sql*Plus

```
$ sqlplus /nolog
SQL*Plus: Release 9.2.0.6.0 - Production on Tue Apr 5 12:34:21 2005
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
SQL> connect / as sysdba
Connected.
```
 - Comprobar el Listener Sql*Net

```
$ lsnrctl status
LSNRCTL for Linux: Version 9.2.0.6.0 - Production on 05-APR-2005 12:36:30
Copyright (c) 1991, 2002, Oracle Corporation. All rights reserved.
...
Instance "MNCS", status READY, has 1 handler(s) for this service...
...
The command completed successfully
```
 - OEM (consola):

```
$ oemapp console
```

PRACTICAS TEMA 1

- 1.1. Comprobar los requerimientos HW y SW para la instalación de Oracle9i sobre Linux.
- 1.2. Comprobar las tareas de preinstalación.
- 1.3. Comprobar cuánto ocupa el sw de Oracle9i.
- 1.4. Buscar el fichero de log de la instalación y echarle un vistazo.
- 1.5. Comprobar tareas de post instalación relacionadas con la seguridad de la BD.
- 1.6. Consulta el “Simulador de Instalación de Oracle9i”.
- 1.7. Revisar alertas de seguridad en OTN.
- 1.8. Revisar el manual de instalación de Oracle9i para Linux.

TEMA 2

DISEÑO Y CREACIÓN DE UNA BASE DE DATOS

TEMA 2.

DISEÑO Y CREACIÓN DE UNA BD

- Contiguas de E/S
 - Diccionario de Datos (tablespace SYSTEM).
 - Procesos (DBWR, LGWR, ARCH).
 - Tipos de segmentos (tablas, índices, temp, rollback).
 - Datos estáticos y dinámicos.
- Dimensionar memoria: SGA (SharedPool, BufferCache, BufferRedoLog).
- Dimensionar CPU.
- Dimensionar ficheros de la BD (RedoLog, System, Temp, Rollback).
- Creación de la BD con CREATE DATABASE y OMF.
- DataBase Configuration Assistant (DBCA).

DISEÑO DE LA BASE DE DATOS

Contiendas en la E/S

- Es muy importante tener en cuenta las posibles contiendas en la E/S:
 - Contienda entre el DD y los datos en sí. En el tablespace SYSTEM sólo debe residir el DD y el segmento de rollback SYSTEM.
 - Contienda entre procesos: DBWR, LGWR y ARCH. Discos diferentes para redolog online y archivado; y el resto de la BD.
 - Contienda entre tipos de segmentos: datos, índices, rollback, temp. Separar segmentos de datos, índices, rollback y temp.
 - Contienda entre datos estáticos y dinámicos. Identificar los datos estáticos (tablas pequeñas: países, provincias, tipos de vía, letras del nif, sexos, etc). Se accederán con mucha frecuencia y sólo en lectura. Si no se detectan a priori, se pueden identificar usando la auditoría (p.e. las que no tengan insert/update/delete durante un día representativo, o varios días).
- Un tablespace para cada área funcional o cada aplicación (gestión económica, gestión de personal, gestión académica, gestión de investigación, etc).

DISEÑO DE LA BASE DE DATOS

Organización de los tablespaces

- Utilizar tablespaces locales.
- Separar los índices de los datos: se puede mover un índice con "alter index rebuild tablespace tsp_indices;".
- Separar los segmentos de rollback. Además crear un tablespace especial de rollback para las transacciones grandes. Se puede crear un tablespace de "undo" si se quiere que el SGBD gestione automáticamente el rollback.
- Separar los segmentos temporales en un tablespace temporal. Además, crear tablespaces temporales adicionales para cada usuario que genere muchos segmentos temporales. Si las aplicaciones usan tablas temporales globales, también deben residir en un tablespace temporal independiente.
- Separar las vistas materializadas en un tablespace independiente.
- Separar las tablas temporales de trabajo, como pueden ser las usadas para cargar información desde ficheros (no hacerlo directamente en las tablas a las que va dirigida finalmente la información).
- En las tablas particionadas, separar las particiones en diferentes tablespaces, al menos los datos vigentes de los históricos.

DIMENSIONAR LA MEMORIA

- La RAM debe albergar no sólo la SGA de nuestra BD, sino todos los procesos de la misma, más aquellos procesos necesarios en nuestro sistema (que no son de la BD). Podemos empezar asignando el 25% de la RAM a la SGA de una BD en producción con una carga razonable (máximo 55-75% de la RAM).
- La memoria necesaria para el ejecutable de oracle, la podemos ver con el comando unix "size".

```
$ size $ORACLE_HOME/bin/oracle
  text  data  bss      dec      hex filename
41989941 309348 21656 42320945 285c431 /u01/app/oracle/product/9.2.0.1.0/bin/oracle
```

- La memoria asociada a cada servidor dedicado (uno por sesión):

- "ps v pid_proceso" o "ps -u usuario v" ("free -m" nos dice la memoria que queda libre)
 - RSS -> Tamaño de la parte residente del proceso en memoria real (en Kb).
 - %MEM -> Porcentaje de la memoria real (RAM) que ocupa la parte residente.

```
$ ps -u oracle v|grep -i local
  PID TTY STAT TIME MAJFL  TRS   DRS  RSS %MEM COMMAND
14778 ?    S     0:00      0 41006 48541 7808 0.7 oracleCURSO01 (DESCRIPTION=(LOCAL=Y
14909 ?    S     0:00      4 41006 48025 7804 0.7 oracleCURSO01 (DESCRIPTION=(LOCAL=Y
```

- Una BD con 100 sesiones podría necesitar unos 800M sólo para procesos de la BD. Para optimizar el uso de la PGA, sin MTS, asignar PGA_AGGREGATE_TARGET, de modo que se usa esa zona de memoria para todas las PGAs (mínimo 10M). En v\$pga_target_advice, buscar 1ª fila con ESTD_OVERALLOC_COUNT=0.

DIMENSIONAR LA SGA

Shared-Pool y Buffer-Caché

- El tamaño aproximado de la SGA es el siguiente:
$$\text{DB_CACHE_SIZE} + [\text{DB_KEEP_CACHE_SIZE} + \text{DB_RECYCLE_CACHE_SIZE} + \text{DB_nk_CACHE_SIZE}] + \text{SHARED_POOL_SIZE} + [\text{LARGE_POOL_SIZE} + \text{JAVA_POOL_SIZE}] + \text{LOG_BUFFERS} + 1\text{MB}$$

Nota: puede haber hasta cuatro DB_nk_CACHE_SIZE, con n = 2, 4, 8, 16, 32k (uno de los tamaños de bloque es el que usa DB_CACHE_SIZE.)
- No existe ninguna fórmula mágica para dimensionar la SGA, sólo aproximaciones y recomendaciones:
 - Memoria recomendada para todas las SGAs de un servidor:
 - Para servidores con RAM \leq 1Gb: RAM * 0.55
 - Para servidores con RAM $>$ 1Gb: RAM * (0.60 a 0.75)
 - Memoria recomendada por instancia: (RAM * 0.55) / nºinstancias.
 - Shared pool = mem por instancia * 0.45
 - Buffer cache = mem por instancia * 0.45
- En Oracle9i se recomienda usar el nuevo parámetro PGA_AGGREGATE_TARGET (sort_area_size se ignora, si no tenemos MTS), pudiendo asignar 80% RAM para oracle y de ese 80%, un 20% para pga_aggregate_target (=RAM*0.80*0.20). El valor mínimo de pga_aggregate_target=10M.

DIMENSIONAR LA SGA

Log Buffer

- Una vez que hemos fijado el tamaño inicial de la SGA, en la vista V\$SHARED_POOL_ADVICE podemos ver si nos interesa redimensionar la SharedPool (el parámetro STATISTICS_LEVEL debe valer ALL o TYPICAL, no BASIC).

```
SQL> SELECT shared_pool_size_for_estimate "Size of Shared Pool in MB",
2      shared_pool_size_factor "Size Factor",
3      estd_lc_time_saved "Time Saved in sec"
4      FROM v$shared_pool_advice;
```

Size of Shared Pool in MB	Size Factor	Time Saved in sec
24	.5	525
48	1	525
72	1.5	525
96	2	526

- El parámetro del init LOCK_SGA=true evita que se haga swapping de la SGA.
- V\$SGA_DYNAMIC_COMPONENTS: Tamaño de las zonas dinámicas.
- LOG_BUFFER no necesita más de pocos cientos de Kb, incluso en un sistema con mucha carga, 1MB posiblemente sea suficiente. Asignar más de 3Mb es inútil ya que el LGWR escribe cuando el buffer se llena 1/3 o 1Mb.

DIMENSIONAR LA CPU

- Si partimos de una sola CPU, podemos detectar que llega a su límite de uso si:
 - Porcentaje máximo de ocupación de CPU = 90%.
 - Porcentaje máximo de procesamiento OS/usuario = 40/60%.

Nota: podemos usar los comandos Linux "top" y "sar".

```
$ sar
```

18:20:00	CPU	% <u>user</u>	%nice	% <u>system</u>	%iowait	% <u>idle</u>
18:30:00	all	0,20	0,00	0,06	0,20	99,54
18:40:00	all	0,16	0,00	0,17	0,12	99,54
Media:	all	0,08	0,00	0,05	0,33	99,54

- Si hay varias CPUs, la carga debe estar balanceada. Para una BD mediana con una carga razonable, lo ideal sería disponer al menos de 2 CPUs.
- Para una instancia con "servidores dedicados" (sin MTS), tendremos un proceso servidor por cada sesión, además de los procesos background. Por ejemplo, para 100 usuarios simultáneos tendremos un mínimo de 100 procesos nuevos en nuestro sistema.

DIMENSIONAR FICHEROS

Redo Log

- Redo = 100M (cada fichero).
- Al menos tres grupos y, si disponemos de discos, los multiplexaremos en dos discos (dos miembros por grupo).
- Empezaremos con ficheros de 100M, intentando que los "log switch" sucedan cada 20-30 minutos.
- Si queremos que los checkpoints coincidan con los "log switch" (y como máximo cada 30 minutos), asignaremos los parámetros:
 - `log_checkpoint_interval=0`
 - `log_checkpoint_timeout=1800` (30 minutos, pondremos 0 si queremos que coincidan con el "log switch", independientemente del tiempo transcurrido).
 - `FAST_START_MTTR_TARGET=0` (desactiva "fast recovery instance").Nota: `log_checkpoint_to_alert=true` muestra checkpoints en fichero `alert.log`.
- `FAST_START_MTTR_TARGET` es nuevo en 9i, y permite indicar el nº de segundos que esperamos que (como mucho) tarde la "recuperación de la instancia" (después de una "caída"). El valor máximo son 3600 segundos (1h).

DIMENSIONAR FICHEROS

System, Temp y Rollback

- SYSTEM \geq 260M. Activaremos el "autoextend" con next=10M.
- TEMP \geq 100M.
 - Los segmentos temporales ocupan espacio igual que una tabla: para ordenar una tabla de 100M, por todas sus columnas, necesitaré 100M.
 - Operaciones que necesitan segmentos temporales: SELECT ORDER BY, GROUP BY, DISTINCT, UNION, INTERSECT or MINUS, non-indexed joins, CREATE INDEX, CREATE/ENABLE PRIMARY/UNIQUE KEY CONSTRAINT, temporary tables, Temporary LOBs, Statistics collection.
 - Oracle9i introduce PGA_AGGREGATE_TARGET para limitar el tamaño total de las PGAs (valor mínimo 10M), quedando obsoleto sort_area_size.
- RBS \geq 100M. Usar modo "auto" (no permite "set transaction"). Calcula nº de segmentos de rollback (RS) a partir de SESSIONS al crear el tablespace. Si usamos modo "manual", crear un RS por cada 4 sesiones, y 20 extensiones iniciales cada uno. "Optimal" debe ser suficiente para 90% de transacciones.

FICHERO DE PARÁMETROS

- Para arrancar la instancia, el servidor Oracle tiene que leer el fichero de parámetros de inicialización (spfile o init), cuya ubicación predeterminada es \$ORACLE_HOME/dbs.
- El fichero de parámetros de inicialización puede ser de dos tipos:
 - Init: se trata de un fichero de texto, editable, cuyo nombre sigue el patrón init\$ORACLE_SID.ora.
 - Spfile: es un fichero binario, no editable pero visualizable, cuyo nombre sigue el patrón spfile\$ORACLE_SID.ora.
- Se crea, a partir de un init, con:

```
CREATE SPFILE [= 'nombre'] FROM PFILE [= 'nombre'];
```

Nota1. Si se omiten los nombres, toma los valores por defecto.
Nota2. La BD no podrá abrir el nuevo spfile hasta el siguiente arranque.
Nota2. Se puede crear un init a partir de un spfile, invirtiendo la sintaxis.
- Los parámetros del spfile se modifican con:

```
ALTER SYSTEM SET parámetro = valor [SCOPE = MEMORY |  
SPFILE | BOTH]
```

Nota. Si queremos modificar el parámetro sólo en el spfile, indicaremos SPFILE.

EJEMPLO DE INIT.ORA

Suponemos RAM=1Gb, un máximo de 50 sesiones simultáneas, checkpoints cada 30min, gestión automática de PGAs (RAM*0.80*0.20, resto SGA):

```
db_name=AVAN01      # Nombre de la BD.
compatible = 9.2.0
background_dump_dest = /u01/app/oracle/admin/AVAN01/bdump
core_dump_dest = /u01/app/oracle/admin/AVAN01/cdump
user_dump_dest = /u01/app/oracle/admin/AVAN01/udump
max_dump_file_size = 10240      # tamaño máximo fichero traza de 5M
control_files = (/u02/oradata/AVAN01/control1.ctl, /
    u03/oradata/AVAN01/control2.ctl)
undo_management = auto          # Gestión automática de Rollback
undo_tablespace = undo_rbs      # Tablespace de "undo"
pga_aggregate_target=160M # Espacio máx todas las PGAs (fuera de SGA)
sga_max_size = 640M            # Tamaño máximo de la SGA (RAM*0.8*0.8)
db_cache_size = 256M          # Tamaño de la caché de datos (SGA*0.45)
shared_pool_size = 256M       # Tamaño de la shared pool (SGA*0.45)
large_pool_size = 0
java_pool_size = 0
log_buffer = 3145728          # Tamaño, en bytes, de la caché de redo (3Mb)
log_checkpoint_interval = 0   # desactiva "checkpoint interval"
log_checkpoint_timeout = 1800 # checkpoint como mucho cada 30minutos.
processes = 50                # N° máx. de procesos (background + sesiones)
```

CREAR LA BASE DE DATOS

- Para crear una BD necesitamos:
 - Conectarnos al servidor Oracle como "SYS AS SYSDBA", autenticándonos contra el S.O. o usando un fichero de claves.
 - Suficiente memoria para arrancar la instancia y espacio en disco para crear la BD.
- Para ubicar los ficheros que componen la BD:
 - Guardaremos, al menos, dos copias del fichero de control, en discos separados.
 - Multiplexaremos los redolog en discos diferentes.
 - Separaremos los ficheros de datos que provoquen contención en disco; por ejemplo: datos, índices, temp y rollback.
- La BD la podemos crear con el asistente gráfico (en Linux "dbca") o con el comando CREATE DATABASE:
 - Crearemos un fichero init.ora, y si queremos, un spfile.ora.
 - Arrancaremos la instancia con STARTUP NOMOUNT.
 - Crearemos la BD con el comando CREATE DATABASE.
 - Ejecutaremos los scripts catalog.sql, catproc.sql y catexp.sql que están en \$ORACLE_HOME/rdbms/admin.

EJEMPLO CON "CREATE DATABASE"

```
connect / as sysdba
startup nomount
CREATE DATABASE "AVAN01"
    maxdatafiles 254
    maxinstances 1
    maxlogfiles 32
    character set WE8ISO8859P15
DATAFILE '/u02/oradata/AVAN01/system01.dbf' SIZE 260M
    AUTOEXTEND ON NEXT 10M EXTENT MANAGEMENT LOCAL
UNDO TABLESPACE UNDO_RBS
    DATAFILE '/u03/oradata/AVAN01/undo_rbs01.dbf' SIZE 100M
DEFAULT TEMPORARY TABLESPACE TEMP
    TEMPFILE '/u03/oradata/AVAN01/temp01.dbf' SIZE 100M REUSE
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K
logfile '/u04/oradata/AVAN01/redo01.log' SIZE 100M,
    '/u04/oradata/AVAN01/redo02.log' SIZE 100M,
    '/u04/oradata/AVAN01/redo03.log' SIZE 100M;
rem *** CREACION DE LAS VISTAS DEL DD ***
@$ORACLE_HOME/rdbms/admin/catalog.sql
@$ORACLE_HOME/rdbms/admin/catproc.sql
@$ORACLE_HOME/rdbms/admin/catexp.sql
connect system/manager
@$ORACLE_HOME/sqlplus/admin/pupbld.sql
```


ORACLE MANAGED FILES (OMF)

- OMF permite que Oracle se encargue de la creación de los ficheros que componen la BD, simplificando la administración de la misma.
- OMF se activa mediante dos parámetros de inicialización:
 - DB_CREATE_FILE_DEST. Define el directorio donde se ubicarán los ficheros.
 - DB_CREATE_ONLINE_LOG_DEST_N. Establece los directorios donde se guardarán (multiplexados) los ficheros redolog; donde N puede valer de 1 a 5.

Nota. Se pueden activar ambos parámetros o sólo uno ellos.

- Ejemplo para crear una BD, usando OMF, separando los ficheros redolog y de control del resto:
 - Parámetros de inicialización (indicando dos copias de redolog y control):
 - DB_CREATE_FILE_DEST='/u02/oradata/CURSO30'
 - DB_CREATE_ONLINE_LOG_DEST_1='/u03/oradata/CURSO30'
 - DB_CREATE_ONLINE_LOG_DEST_2='/u04/oradata/CURSO30'
 - Creación de la BD:

```
CREATE DATABASE DEFAULT TEMPORARY TABLESPACE TEMP
UNDO TABLESPACE UNDO_RBS;
```

DBCA (DataBase Configuration Assistant)

- En Unix/Linux se ejecuta con "dbca". Se trata de un asistente gráfico que permite:
 - Crear una BD. A partir de plantillas predefinidas, que contienen:
 - Sólo la estructura. Se pueden cambiar todos los parámetros del init.
 - También con ficheros: no se puede cambiar nada.
 - Añadir opciones a una BD existente (Java, Intermedia, Spatial, OLAP, etc).
 - Borrar una BD.
 - Gestionar plantillas de BD: crear una plantilla a partir de otra, o a partir de sólo la estructura de una BD, o también incluyendo los datos de la misma.

VISTAS DEL DD

- V\$DATABASE (Base de datos).
- V\$INSTANCE (Instancia).
- V\$SGA (SGA).
- V\$SGA_DYNAMIC_COMPONENTS (Zonas dinámicas de la SGA).
- V\$SGASTAT (SGA detallada).
- V\$SHARED_POOL_ADVICE (Estimaciones de tamaño de la SharedPool)
- V\$BUFFER_POOL (Buffers en la caché de datos)
- V\$DB_CACHE_ADVICE (Estimaciones db_cache_size)
- V\$PGA_TARGET_ADVICE (Estimaciones pga_aggregate_target)
- V\$SQLAREA (Sentencias SQL).
- V\$PROCESS (Procesos).
- V\$BGPROCESS (Procesos background).
- V\$DATAFILE (Ficheros de datos de la BD).
- V\$CONTROLFILE (Ficheros de control de la BD).
- V\$LOGFILE (Ficheros redo log de la BD).

PRACTICAS TEMA 2

- 2.1. Diseñar una BD teniendo en cuenta las contiguas de E/S y la organización de los tablespaces básicos. Suponer 4 discos: /u01 (SW) y /u02, /u03, /u04 (BD).
- 2.2. Crear un init.ora, diseñando requerimientos de memoria, suponiendo 1Gb de RAM y 21 instancias con 10 usuarios cada una. Se puede usar un máximo del 75% de la RAM para las SGAs de todas las bases de datos. Los checkpoints sucederán cuando se llene el fichero redolog y, como máximo, cada 30 minutos. La gestión de rollback será automática. Tendremos dos copias del fichero de control en /u02 y /u03.
- 2.3. Crear la BD que hemos diseñado anteriormente, y para la que también hemos creado un init.ora. Hacerlo manualmente con CREATE DATABASE. El nombre de la BD será ALUxy. Antes de lanzar el comando de creación, ponte de acuerdo con un compañero para que no lo lanceis a la vez.
- 2.4. Crear la misma BD usando OMF, y cambiando el nombre por OMFxy.
- 2.5. Consultar el "Simulador de DBCA".

TEMA 3

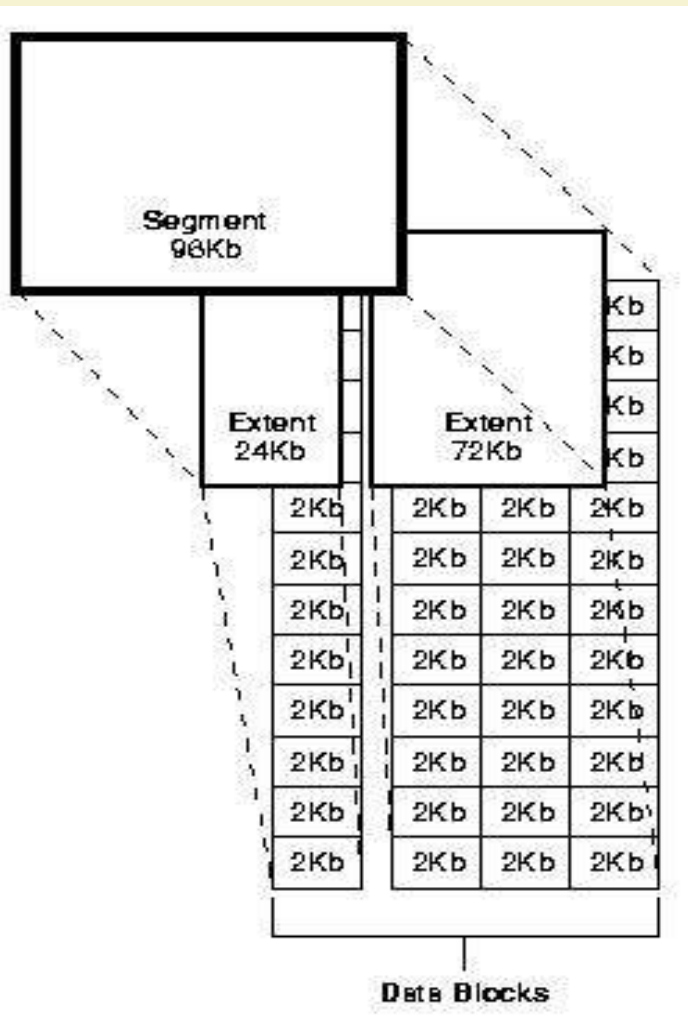
ESTRUCTURAS DE ALMACENAMIENTO

TEMA 3.

ESTRUCTURAS DE ALMACENAMIENTO

- Bloques, extensiones y segmentos.
- Varios tamaños de bloque en la misma BD.
- Formato del bloque.
- Generación del espacio libre en un bloque.
- Estructura de la fila.
- Extensiones y su asignación.
- Segmentos.
- Gestión automática del espacio de los segmentos.
- Comprimir los datos de un segmento.
- Métodos de acceso a datos: índices B-tree y Bitmap.

BLOQUES, EXTENSIONES Y SEGMENTOS



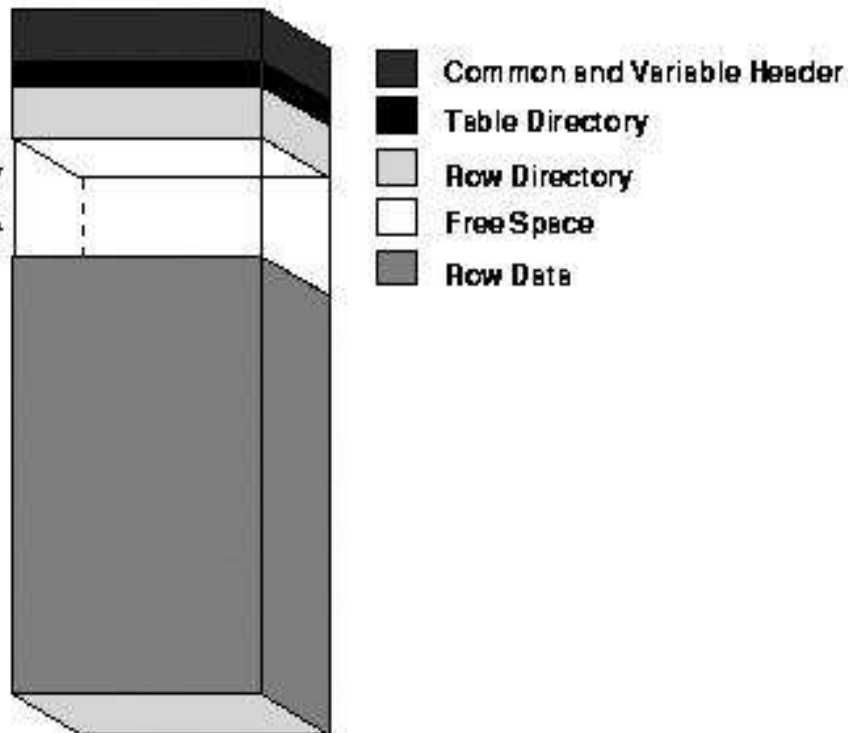
- La unidad mínima de asignación de espacio es el bloque de datos (múltiplo del bloque del S.O.: 2, 4, 8, 16 y 32Kb); de modo que un conjunto contiguo de bloques forman una extensión. Y las extensiones forman segmentos de una determinada estructura de datos (tablas, índices, rollback, temporal, etc).
- Oracle asigna espacio a un segmento en unidades de una extensión. Todas las extensiones de un mismo segmento residen en el mismo tablespace.
- Oracle pide espacio en múltiplos del bloque de datos, cuyo tamaño se indica con el parámetro `db_block_size`.

DIFERENTES TAMAÑOS DE BLOQUE EN LA MISMA BD

- Aunque el parámetro `db_block_size` define el tamaño del bloque "estandar" de la BD, se pueden usar otros tamaños indicándolo a nivel de tablespace, y definiendo la correspondiente caché de datos con el nuevo tamaño de bloque.
- La Database Buffer Cache consta de tres cachés independientes:
 - `DB_CACHE_SIZE`. Dimensiona la caché por defecto, que siempre existe y cuyo tamaño no puede valer cero.
 - `DB_KEEP_CACHE_SIZE`. Dimensiona la caché donde se guardarán los bloques de tablas que se usan con mucha frecuencia.
 - `DB_RECYCLE_CACHE_SIZE`. Dimensiona la caché que almacena los bloques de las tablas que se usan muy poco.
 - `DB_nK_CACHE_SIZE`. Dimensiona hasta 4 cachés con tamaño de bloque "no estandar", con $n=2,4,8,16,32$.
 - El uso de una u otra caché lo indicaremos con el parámetro `BUFFER_POOL`, de la clausula `STORAGE` de la tabla: `keep`, `recycle` o `default`.
- Al crear un tablespace, con la clausula `BLOCKSIZE`, se puede especificar una tamaño de bloque "no estandar". Para ello debe estar definido el parámetro `DB_CACHE_SIZE` y, al menos, un `DB_nK_CACHE_SIZE`; de forma que n coincida con el valor que acompaña a `BLOCKSIZE`. No se pueden indicar tamaño de bloque no estandar para tablespaces temporales.

FORMATO DEL BLOQUE

Database Block



- Cabecera: información general del bloque (dirección del bloque, tipo de segmento, etc). 4 bytes.
- Directorio de tabla: información sobre la tabla. 4 bytes.
- Directorio de filas: información sobre las filas que actualmente contiene el bloque (direcciones, etc). 2 bytes por fila.
- Overhead (es el conjunto de los tres anteriores). Como media, su tamaño oscila entre 84 y 107 bytes.
- Datos

- Espacio libre: se usa para inserts y updates, y también para registrar las transacciones (“transaction entries”, unos 23 bytes por cada una).

GENERACION DE ESPACIO LIBRE EN UN BLOQUE

- Sentencias que generan espacio libre en un bloque: delete y update. Dicho espacio quedará disponible para los inserts posteriores si:
 - El insert sucede en la misma transacción que libera espacio.
 - Sucede en otra transacción después del commit de la primera.
- El espacio libre no tiene por qué estar contiguo. Sólo se compactará si al hacer un insert/update hay suficiente espacio libre, pero no está contiguo.
- Row Chaining. Si una nueva fila no cabe en un bloque, será almacenada en una "cadena de bloques" (dos o más). Si se trata de una fila que aumenta de tamaño por un update, y no cabe en el bloque, su contenido será "migrado" a otro bloque, manteniendo un puntero en el bloque original.
- Se reservará, al menos, el espacio libre indicado por PCTFREE (%) para actualizaciones. Cuando el espacio libre baje a pctfree, no se permitirán más inserciones (hasta que el espacio ocupado no baje de PCUSED).

ESTRUCTURA DE LA FILA

- Estructura de la fila:
 - Cabecera: 3 bytes (de ellos un byte para nºcolumnas).
 - Rowid: 6 bytes (1 para el fichero, 4 para el bloque y 1 para fila dentro del bloque).
 - Cabecera de columna: 1byte (o 3bytes si columna>250bytes).
 - Columna: datos (si null, no ocupa espacio).
 - Cabecera de columna.
 - Columna ...
- Longitud de las columnas:
 - char(n): n bytes.
 - varchar2(n): hasta n bytes.
 - Date: 8 bytes en 9i (antes 7 bytes)
 - number(n,m): $n/2$ (entero mayor), más 1 byte. Por ejemplo: number(9,2), $9/2=4.5$, $5+1=6$.

EXTENSIONES Y SU ASIGNACIÓN

- Una extensión es una unidad lógica de almacenamiento compuesta por un conjunto de bloques contiguos. Una o más extensiones componen un segmento.
- Oracle recomienda usar tablespaces “manejados localmente”, con extensiones de tamaño uniforme (uniform size) o variable (autoallocate), donde ya no sirven los parámetros next, pctincrease, minextents, maxextents y default storage:
 - Autoallocate. Puedes definir la extensión inicial y Oracle determina el tamaño óptimo para las siguientes (tamaño mínimo 64Kb). Es el defecto para tablespaces permanentes.
 - Uniform Size. Puedes indicar el tamaño de la extensión o usar el valor por defecto (1M). Los tablespaces temporales “locales” sólo pueden usar este tipo de asignación de extensiones.
- Para asignar una extensión a un segmento, en un tablespace local, Oracle busca en el bitmap del fichero el número requerido de bloques libres contiguos.

SEGMENTOS

- Un segmento es un conjunto de extensiones que contienen todos los datos de una estructura lógica de almacenamiento específica, dentro de un tablespace (segmentos de datos, segmentos de índice, segmentos temporales, etc).
- Segmento de datos: es una tabla normal, o una partición de una particionada, o un cluster de tablas.
- Segmento de índices: índice no particionado, o partición de uno particionado. No tienen que estar en el mismo tablespace que las tablas correspondientes.
- Segmentos temporales: Oracle los usa para hacer ordenaciones, que no caben en memoria: `create index`, `select ... order by`, `select distinct`, `select ... group by`, `select ... union`, `select ... intersect`, `select ... minus`. También se usan para crear tablas temporales y sus índices (son tablas que contienen datos sólo durante una transacción o una sesión). Se puede mejorar su rendimiento ajustando el parámetro `sort_area_size`. Sin MTS, usar `pga_aggregate_target` para todas las áreas de sort de todas las sesiones (todas las PGAs).

GESTION AUTOMATICA DEL ESPACIO DE LOS SEGMENTOS

- Desde la versión 9i se incluye la posibilidad de gestionar el espacio de los segmentos automáticamente, siendo más fácil, con mejor utilización del espacio, y también mejor rendimiento en las operaciones concurrentes de inserción.
- Cada segmento contiene un bitmap que describe el estado de ocupación de cada bloque, y que se guarda en un conjunto independiente de bloques.
- Se puede activar sólo para los tablespaces permanentes manejados localmente, excepto el SYSTEM:

```
CREATE TABLESPACE TSP_CURSO
  DATAFILE '/u02/oradata/CURSO/tsp_curso01.dbf' SIZE 10M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K
  SEGMENT SPACE MANAGEMENT AUTO;
```

Se ignoran las cláusulas PCTUSED, FREELISTS y FREELISTS GROUPS.

- Una vez creado un tablespace local con gestión automática del espacio de los segmentos, no tiene vuelta atrás.
- Oracle recomienda la gestión automática porque obtiene mayor rendimiento y es mucho más fácil de mantener.

COMPRESION DE LOS SEGMENTOS DE DATOS

- Podemos indicar la clausula COMPRESS al crear una tabla (por defecto será NOCOMPRESS), para que Oracle intente comprimir los datos (excepto para tipos LOB y tablas IOT). Esto es muy útil en entornos con muy pocas actualizaciones (insert/update), como datawarehouse.
- Se reduce el uso de disco y memoria (caché de datos) y se recomienda para las operaciones de sólo lectura.
- Los datos se comprimen, en cada bloque, eliminando los valores repetidos de una misma columna, que se guardan en una tabla. El factor de compresión será mayor en bloques con valores repetidos. Se pueden reorganizar las filas agrupando valores repetidos de una columna en el mismo bloque: "create table as select order by", ordenando por columnas con baja cardinalidad.
- Al crear un tablespace, podemos indicar COMPRESS en la clausula DEFAULT; para que por defecto, se compriman los datos en las tablas que contenga.
- Debido a un bug, la compresión sólo funciona con $\geq 9.2.0.3$.

METODOS DE ACCESO A DATOS

INDICES B-TREE Y BITMAP

- Usaremos **índices B-tree**, en general, cuando vayamos a hacer consultas que acceden frecuentemente a no más del 10-15% de las filas de la tabla.
- La altura de un B-tree (BLEVEL) no debe ser mayor de tres (mejor ≤ 2). Rellenaremos *_INDEXES.BLEVEL haciendo analyze del índice:

```
analyze index i_nombre compute statistics;
```
- Podemos reconstruir un índice degradado haciendo un "rebuild":

```
alter index i_nombre rebuild;
```
- **Índices Bitmap**; muy útiles para accesos en sólo lectura sobre columnas de baja cardinalidad, en tablas muy grandes (aplicaciones DSS). Muy eficientes para predicados OR. No usarlos si se hacen updates con frecuencia (aplicaciones OLTP). Bloqueos a nivel de segmento de bitmap (no de fila).

```
create bitmap index i_b_nombre on t_tabla (cols) tablespace tsp_nombre;
```
- Las **IOT** son tablas almacenadas en un índice B-tree. Son útiles para tablas estaticas que se acceden con frecuencia por la clave primaria (casi todas sus columnas forma parte de la PK), y que no van a tener ningún otro índice.

VISTAS DEL DD

- DBA_TABLESPACES (Tablespaces de la BD).
- DBA_SEGMENTS (Segmentos que hay en los tablespaces).
- DBA_EXTENTS (Extensiones que componen los segmentos).
- DBA_FREE_SPACE (Extensiones libres en cada tablespace).
- DBA_INDEXES (Indices creados sobre tablas de la BD).
- `SELECT DBMS_METADATA.GET_DDL('TABLE','T1') FROM DUAL;`
(muestra la sentencia DDL para crear la tabla T1).

PRACTICAS TEMA 3

- 3.1. Comprobar el tamaño del bloque de la BD. ¿Cuántas cachés de datos hay definidas?. Definir la caché keep con 1M. Definir también una caché para bloques de 4Kb, con 1M.
- 3.2. Crear un tablespace SEGAUTO con gestión automática de segmentos.
- 3.3. Crear una tabla TABLA01 en el tablespace anterior, con una columna VARCHAR2(1000). Insertar tres filas de forma que ocupe varias extensiones. Borrar todas las filas con "delete" y comprobar las extensiones de la tabla. Hacer "truncate table" y comprobar de nuevo el espacio ocupado por la tabla.
- 3.4. Crear una tabla TABLA02 igual que la anterior, pero con compresión de datos.

PRACTICAS TEMA 3

- 3.5. Crear tablespace TSP4K con tamaño de bloque de 4Kb. Crear una tabla TABLA03, igual que TABLA01, y comprobar cuantos bloques ocupan las dos y comparar los datos.
- 3.6. Comprobar los parámetros `pga_aggregate_target`, `workarea_size_policy` y `sort_area_size`. ¿Se puede poner `pga_aggregate_target=0`? ¿cómo desactivar `pga_aggregate_target`?
- 3.7. Crear una TABLA04 en el tablespace SEGAUTO, con cuatro columnas: `c1 char(2)`, `c2 varchar2(10)`, `c3 date` y `c4 number(10,2)`. Estimar el tamaño medio de la fila, y el número medio de filas por bloque. insertar 100 filas en la tabla , calcular las estadísticas y comprobar los datos calculados (número de filas, número de bloques y longitud media de la fila).

TEMA 4

AJUSTE Y MONITORIZACIÓN DE LA INSTANCIA

TEMA 4.

AJUSTE Y MONITORIZACIÓN DE LA INSTANCIA

- Eventos de espera.
- Vistas dinámicas: información del sistema y de las sesiones.
- Ajuste de la Shared-Pool.
- Ajuste de la buffer cache.
- Ajuste de los buffers redo log.
- Ajuste del rollback.
- Ajuste del tablespace temporal.
- Ajuste de los checkpoints.
- Ajuste de las sentencias SQL.
- Ajuste de la CPU.
- Utilidad STATSPACK.

EVENTOS DE ESPERA

- V\$EVENT_NAME. Lista de posibles esperas.
 - V\$SYSTEM_EVENT. Esperas totales por evento desde arranque de BD.
 - V\$SESSION_EVENT. La misma información anterior, por sesión.
 - V\$SESSION_WAIT. Sesiones que están esperando actualmente.
 - Para recibir valores en WAIT_TIME poner TIMED_STATISTICS=TRUE.
 - Eventos más comunes:
 - free buffer waits: DBWR escribe poco a menudo (subir nº de checkpoints).
 - latch free: contención en latches (verificar V\$LATCH).
 - buffer busy waits: hay contención E/S (ajustar E/S o distribuir datos).
 - db file sequential read: hay contención E/S (ajustar E/S o distribuir datos)
 - db file scattered read: igual anterior, pero multibloque (ajustar E/S o distribuir datos).
 - db file parallel write: checkpoints muy espaciados (subir su número).
 - undo segment tx slot: faltan segmentos de rollback (añadir más).
 - undo segment extension: demasiadas extensiones dinámicas (ampliaciones y reducciones) de los segmentos de rollback (ajustar tamaño).
- Nota: la lista completa de eventos está en el manual "Oracle9i Database Reference".

<http://cursos.atika.um.es/oradoc92/server.920/a96536/apa5.htm>

VISTAS DINAMICAS

INFORMACION DEL SISTEMA

- Base de datos
 - **V\$SYSTEM_EVENT**: esperas totales por evento.
 - **V\$SYSSTAT**: estadísticas básicas de la instancia.
 - **V\$SGASTAT**: estado de uso de la SGA.
 - V\$WAITSTAT: estadísticas de contención.
 - V\$PROCESS: procesos oracle.
 - V\$SESSION: sesiones en curso.
 - V\$SORT_SEGMENT: estado de uso de los segmentos temporales.
 - V\$PGASTAT: estadísticas de uso de la PGA.
- Memoria
 - **V\$BUFFER_POOL_STATISTICS**: estadísticas de la caché de datos.
 - **V\$LIBRARYCACHE**: Rendimiento de la Library Caché.
 - V\$DB_OBJECT_CACHE: Objetos que hay en la Library Caché.
 - V\$SQLAREA: Sentencias SQL y estadísticas asociadas.
 - V\$ROWCACHE: Rendimiento de la Dictionary Caché.

VISTAS DINAMICAS

INFORMACION DEL SISTEMA

- E/S
 - **V\$FILESTAT**: Estadísticas de E/S de los ficheros de datos.
 - **V\$TEMPSTAT**: Estadísticas de E/S de los ficheros temporales.
- Contención
 - **V\$LATCH**: Estadísticas de latches.
 - **V\$WAITSTAT**: Estadísticas de contención.
 - **V\$ROLLSTAT**: Estadísticas de los segmentos de rollback.
 - **V\$UNDOSTAT**: Estadísticas de undo.

VISTAS DINAMICAS

INFORMACION DE SESIONES

- Sesión (datos de las sesiones actualmente en curso)
 - **V\$SESSION**: sesiones.
 - **V\$SESSTAT**: estadísticas de cada sesión.
 - **V\$SESSION_EVENT**: eventos de cada sesión.
 - **V\$SESSION_WAIT**: esperas de cada sesión.
 - V\$LOCK: bloqueos actuales y peticiones de bloqueo.
 - V\$ACCESS: objetos bloqueados y las sesiones que los están usando.
 - V\$TRANSACTION: transacciones en curso.
 - V\$OPEN_CURSOR: cursores abiertos y compilados.
 - V\$SORT_USAGE: segmentos temporales en uso
 - V\$SESS_IO: estadísticas de E/S de cada sesión.

Nota: la lista completa de vistas dinámicas está en el manual "Oracle9i Database Reference".

AJUSTE DE LA SHARED-POOL

INTRODUCCION

- Oracle guarda las sentencias SQL y los paquetes en la Shared-pool, una caché que presenta los siguientes problemas:
 - La unidad de reserva de espacio no es constante, variando desde unos cuantos bytes hasta muchos Kb.
 - No toda la memoria puede ser liberada cuando el usuario termina con ella; ya que se trata de una caché que pretende maximizar la compartición.
 - No dispone de una zona en disco para paginar.
 - Posibles síntomas de problemas de ajuste de la Shared Pool:
 - Contención en los latches “%library cache%” (V\$LATCH).
 - Contención en el latch “%shared pool%” (V\$LATCH).
 - Altos tiempos de CPU para compilar, “parse time cpu” (V\$SYSSTAT).
 - Muchas recargas (reloads) en V\$LIBRARYCACHE.
 - Muchas llamadas de compilación, “parse count%” (V\$SYSSTAT).
 - Frecuentes errores ORA-04031, debidos a la fragmentación.
- Nota: los latches son microbloqueos necesarios para proteger las operaciones en la SGA (SharedPool, LibraryCache, RedoBuffer, etc). Son puntos potenciales de contención.

AJUSTE DE LA SHARED-POOL

SENTENCIAS SQL

- La ejecución de una sentencia SQL tiene varias fases: análisis sintáctico y semántico (compilación), y cálculo del plan de ejecución:
 - En la “library cache” se guardan las sentencias SQL ya compiladas.
 - En la “sql area” se guardan los planes de ejecución de cada una de ellas.
- Al compilar una sentencia SQL, hay que distinguir entre “hard” y “soft” parse:
 - Hard Parse: la sentencia SQL no existe en la SharedPool (Library Cache). Es costoso en términos de CPU y latches.
 - Soft Parse: la sentencia SQL ya existe en la SharedPool y puede usar una versión de la misma.
- Dos sentencias SQL son iguales si tienen el mismo texto (incluyendo espacios en blanco y mayúsculas/minúsculas); y además:
 - Los nombres de objetos deben apuntar a los mismo objetos reales.
 - El modo del optimizador (optimizer goal) debe ser el mismo.
 - Los tipos y longitudes de las variables bind deben ser los mismos.
 - El entorno NLS (idioma y país) debe ser el mismo.

AJUSTE DE LA SHARED-POOL GENERALIDADES

- Que siempre haya espacio libre en la SharedPool, puede significar que sobra; y que haya poco o nada no supone un problema si el rendimiento es bueno:

```
SELECT * FROM V$SGASTAT WHERE NAME = 'free memory' AND POOL = 'shared pool';
POOL          NAME                                BYTES
-----
shared pool free memory                          693036
```

- En la library cache, el pinhitratio $\geq 95\%$ (V\$LIBRARYCACHE) y lo más cercano a 1 (100%). Reloads debe ser casi 0.
- En la row cache (V\$ROWCACHE) el ratio getmisses/gets $\leq 15\%$.
- Las aplicaciones OLTP deben usar "bind variables" (no para DSS).
Nota: en DSS es mejor darle mucha información al CBO q compartir el código.
- Usar DBMS_SHARED_POOL.KEEP (dbmspool.sql) para fijar paquetes muy usados en la SharedPool (como SYS.STANDARD).
- Haciendo "flush" se puede eliminar la fragmentación (ora-4031):
`alter system flush shared_pool;`

Nota: puede bajar rendimiento hasta que objetos vuelven a la caché. No usar cuando BD tiene mucha carga. No descarga paquetes "fijados", ni sentencias ya compiladas de sesiones en curso.

AJUSTE DE LA SHARED-POOL LIBRARY CACHE

- V\$LIBRARYCACHE. Estadísticas de la Library Caché.
 - GETS (peticiones) – Cada petición de ejecutar una sentencia SQL (¿está el SQL en memoria?).
 - PINS (ejecuciones) – Cada petición de metadatos de un SQL (¿está el SQL ya compilado?, si lo está se ejecuta). PINHITRATIO debe ser casi 1.
 - RELOADS (recompilaciones) – Cada petición de metadatos (PIN) que no los encuentra en memoria pq los ha sacado el algoritmo LRU (ejecuciones que requieren recompilar sentencia). Debe ser casi 0.
- Los NAMESPACE que se corresponden con sentencias SQL y PL/SQL son: SQL AREA, TABLE/PROCEDURE, BODY y TRIGGER.
- Para hacer que RELOADS sea casi 0, podemos subir shared_pool_size, pero además, para aprovechar dicho incremento, subiremos también open_cursors.
- Si RELOADS es casi 0 y siempre tenemos una cantidad significativa de la SharedPool libre, bajaremos shared_pool_size.

AJUSTE DE LA SHARED-POOL ROW CACHE

- V\$ROWCACHE. Estadísticas de la Row Caché (Diccionario de Datos).
 - PARAMETER: tipo de petición.
 - GETS: peticiones a la caché, del tipo en cuestión.
 - GETMISSES: peticiones fallidas que generan E/S.
 - MODIFICATIONS: actualizaciones de la caché.

```
SELECT parameter, sum(gets), sum(getmisses)
      , 100*sum(gets - getmisses) / sum(gets) pct_succ_gets
      , sum(modifications) updates
FROM V$ROWCACHE WHERE gets > 0 GROUP BY parameter;
```

PARAMETER	SUM(GETS)	SUM(GETMISSES)	PCT_SUCC_GETS	UPDATES
dc_object_ids	16942	537	96.8303624	173
dc_objects	7534	966	87.1781258	414
...				

- Debemos procurar que ratios $\geq 85\%$.
- Podemos ver el ratio general con:

```
SELECT (SUM(GETS - GETMISSES - FIXED)) / SUM(GETS) "ROWCACHE" FROM V$ROWCACHE;
ROWCACHE
-----
.982133497
```

AJUSTE DE LA SHARED-POOL RESTO DE VISTAS DINAMICAS

- En V\$SHARED_POOL_ADVICE podemos ver si nos interesa redimensionar la SharedPool (el parámetro STATISTICS_LEVEL debe valer ALL o TYPICAL, no BASIC).

```
select SHARED_POOL_SIZE_FOR_ESTIMATE SIZE_ESTIMATE,  
SHARED_POOL_SIZE_FACTOR SIZE_FACTOR,  
ESTD_LC_TIME_SAVED_FACTOR PARSE_SAVED_FACTOR  
from v$shared_pool_advice;
```

```
SIZE_ESTIMATE SIZE_FACTOR PARSE_SAVED_FACTOR  
-----  
4 .5 .9806  
8 1 1  
16 2 1.0194
```

- V\$SGASTAT. Detalle de cada una de las partes de la SharedPool.
- V\$SQLAREA. Estadísticas sobre todos los cursores compartidos, incluyendo el texto inicial (1000 caracteres) de cada sentencia. El uso de esta vista consume muchos latches (V\$SQL no).
- V\$SQLTEXT. Texto completo de las sentencias, en varias líneas (filas).
- V\$DB_OBJECT_CACHE. Objetos en caché, incluyendo paquetes, funciones, procedimientos, tablas, índices, sinónimos, secuencias, vistas, triggers, etc.

AJUSTE DE LA SHARED-POOL

COMPARTIR CODIGO SQL

- Nos fijaremos el objetivo “compilar una vez y ejecutar muchas”.
- Detectar sentencias similares que usan literales (V\$SQLAREA):

```
SELECT substr(sql_text,1,40) "SQL", count(*), sum(executions) "TotExecs"  
FROM v$sqlarea  
WHERE executions < 5  
GROUP BY substr(sql_text,1,40) HAVING count(*) > 30 ORDER BY 2;
```

Nota: los valores 5, 40 y 30 son ejemplo para detectar sentencias que se ejecutan poco (<5), cuyos 40 primeros caracteres son iguales en muchos casos (>30). Luego veríamos si es posible convertirlas en una o en unas pocas.

- En 9i, con CURSOR_SHARING=SIMILAR (antes sólo FORCE), Oracle determina qué literales puede sustituir por variables bind sin afectar el plan de ejecución. Si afectase al plan de ejecución no se hace el cambio (si se usa FORCE sí).
- Caché privada sesión (SharedPool): al compilar una sentencia primero se busca en ella. Empezar con SESSION_CACHED_CURSORS=50 (valor alto puede provocar ora-4031, comprobar % de uso). Util con Oracle Forms cuando los forms se abren y cierran con frecuencia.

AJUSTE DE LA SHARED POOL LATCHES DE LA LIBRARY CACHE

Si alguno de los latches está provocando la mayoría de “sleeps”, entonces hay un problema. Hay que tener en cuenta que estos datos se acumulan desde el arranque de la BD, y por tanto no muestran problemas “intermitentes”.

```
select name,gets,misses,sleeps from v$latch
where name like 'library%';
```

NAME	GETS	MISSES	SLEEPS
library cache	97155739	42902	12272
library cache load lock	11705	0	0

“misses” son fallos al intentar coger un “latch”, y “sleeps” son aquellos fallos que provocan que la sesión correspondiente se ponga a “dormir” (lo cual implica una espera).

Para detectar el proceso en cuestión (sólo saldrá algo si lo pillamos en el momento):

```
select a.name,pid from v$latch a , v$latchholder b
where a.addr=b.laddr and a.name = 'library cache%';
```

AJUSTE DE LA SHARED POOL ESPERAS POR LATCHES

Consultando V\$SESSION_WAIT durante un periodo de "lentitud", se puede determinar si hay un problema con "latches" y, en tal caso, con qué "latch" concreto. Si hay más de 3 ó 4 procesos esperando , puede haber un problema.

```
select count(*) number_of_waiters
  from v$session_wait w, v$latch l
 where w.wait_time = 0 and w.event = 'latch free' and
       w.p2 = l.latch# and l.name like 'library%';
```

También es interesante mirar sólo en v\$session_wait para ver si hay alguna otra causa de "lentitud".

```
select * from v$session_wait
 where event != 'rdbms ipc message' and
       event not like '%Net%' and sid > 5;
```

AJUSTE DE LA BUFFER CACHE

GENERALIDADES

- Oracle guarda copias de los bloques de datos en la "buffer caché" (caché de datos). Puede haber copias de diferentes puntos del tiempo y también bloques "dirty" (modificados pero que no se han llevado a disco).
- Cuando la caché se llena, Oracle moverá parte de sus bloques a disco (usa algoritmo LRU sobre la lista, o listas, de bloques no "dirty"), de modo que si posteriormente son accedidos tendrán que volver a recuperarse del disco.
- El latch "cache buffers lru chain" serializa operaciones sobre la(s) lista(s) LRU.
- El proceso DBWR es el responsable de llevar los bloques "dirty" a disco. Cualquier sesión puede leer los bloques de la caché.
- Podemos calcular el ratio de eficiencia (hit ratio) de la caché de datos consultando V\$SYSSTAT. Se recomienda que sea >80% para aplicaciones OLTP; pero esto no asegura un buen rendimiento (tb es muy importante el uso de índices selectivos).
- En aplicaciones DSS (datawarehouse) se pueden tener ratios mucho menores.

AJUSTE DE LA BUFFER CACHE HIT RATIO

- Consultando V\$SYSSTAT:

```
select 100*(1 - (f1 - f2 - f3)/(r1 + r2 - f2 -f3)) HitRatio
from (select value f1 from v$sysstat where name='physical reads'),
     (select value f2 from v$sysstat where name='physical reads direct'),
     (select value f3 from v$sysstat where name='physical reads direct (lob)'),
     (select value r1 from v$sysstat where name='consistent gets'),
     (select value r2 from v$sysstat where name='db block gets');
```

- Si usamos varias cachés (keep, recycle, etc), a partir de V\$BUFFER_POOL_STATISTICS:

```
SELECT name,
       100*(1-(physical_reads / (consistent_gets + db_block_gets))) HIT_RATIO
FROM V$BUFFER_POOL_STATISTICS
WHERE ( consistent_gets + db_block_gets ) !=0;
```

- Si activamos DB_CACHE_ADVICE, podremos consultar en V\$DB_CACHE_ADVICE posibles estimaciones para la "buffer caché".

```
select name,size_for_estimate,size_factor,ESTD_PHYSICAL_READ_FACTOR
from v$db_cache_advice;
```

NAME	SIZE_FOR_ESTIMATE	SIZE_FACTOR	ESTD_PHYSICAL_READ_FACTOR
DEFAULT	4	1	1
DEFAULT	8	2	.3228

AJUSTE DE LA BUFFER CACHE

OBJETIVOS

- Hay que evitar lo siguiente:
 - Básicamente las lecturas innecesarias, pues llevan bloques a la caché y, por tanto, provocan la salida de otros (que luego tendrán q volver a leerse de disco). Cuidado con los índices poco o nada selectivos y los "full scan" de tablas.
 - Contención en el latch 'cache buffers lru chain' (V\$LATCH).
 - Mucho tiempo empleado en la espera "write complete waits" (V\$BUFFER_POOL_STATISTICS).
 - Mucho tiempo empleado en la espera "free buffer waits" (V\$BUFFER_POOL_STATISTICS).
- Factores que puede mejorar el rendimiento del DBWR:
 - Atributos de los discos físicos (stripe size, velocidad, etc).
 - Raw devices vs File Systems.
 - Distribuir las escrituras sobre más discos/ficheros.
 - Usar E/S asíncrona cuando esté disponible.
 - Usar varios DBWRs cuando no haya E/S asíncrona (DBWR_IO_SLAVES).
 - Usar las diferentes cachés (keep, recycle, etc).

AJUSTE DEL REDO LOG BUFFER

GENERALIDADES

- La Caché de Redo (Redo Log Buffer) es un buffer circular en la SGA, que guarda información (redo entries) sobre todos los cambios hechos en la BD, que se utilizará sólo para recuperar la BD, si es necesario. Las "redo entries" van ocupando espacio contiguo, de forma secuencial, en la Caché de Redo.
- El LGWR es el encargado de escribir las "redo entries" de la cache de redo, en el fichero (o grupo) redo log activo. Una vez hecho ésto, el proceso de usuario puede reutilizar dichas entradas.
- Hay determinadas operaciones en las que resulta interesante "desactivar" el redo, ya que se pueden reconstruir fácilmente; como son crear un índice o crear una tabla "as select". La opción a usar en estos casos es NOLOGGIN.
- Dimensionar la Caché de Redo (log_buffer) por encima de 3M, no sirve de nada; ya que el LGWR vuelca la caché a disco cuando ésta se llena 1/3 ó máximo 1M.

AJUSTE DEL REDO LOG BUFFER

DETECTAR PROBLEMAS

- Contención en latches (V\$LATCH) de redo. Si el ratio de MISSES/GETS o el de IMMEDIATE_MISSES/(IMMEDIATE_GETS+IMMEDIATE_MISSES) > 1% es porque hay contención:

```
SELECT name, gets, misses, immediate_gets, immediate_misses
FROM v$latch
WHERE name in ('redo allocation', 'redo copy');
```

Si hay contención en "redo allocation", reduciremos uso de redo (NOLOGIN) o subiremos el tamaño de la Caché de Redo (log_buffer). Si CPUs>16 se puede subir LOG_PARALLELISM (de 2 a 8).

Si es en "redo copy", y CPUs > 1, se puede subir _log_simultaneous_copies (oculto).

- Contención en peticiones de espacio de redo en disco. Se refiere a la estadística "redo log space requests" de la vista V\$SYSSTAT, que refleja el nº de esperas al escribir el redo a disco, pq se ha llenado el fichero redolog. Ese valor debe ser (casi) cero. Si, por el contrario, dicho nº va incrementándose continuamente, habrá que comprobar checkpoints y log_switchs. Se debe hacer q los checkpoints coincidan con los log switchs (cada 15-30 minutos aprox.).

AJUSTE DEL ROLLBACK GENERALIDADES

- En los segmentos de rollback (RS) se guardan las imágenes de los datos, anteriores a las actualizaciones, por si es necesario deshacer las transacciones (TR) en cuestión; y para mantener la consistencia en lectura. Oracle va asignando los RS mediante "round robin".
- Cada TR sólo puede usar un RS para almacenar sus registros de undo.
- Varias TR pueden escribir en la misma extensión.
- Un RS se organiza en forma de anillo, de forma que la "cabeza" del RS nunca machacará una extensión ocupada por la "cola" del mismo.
- Las extensiones de un RS se usarán en orden. Si no puede usar la siguiente, reservará una nueva y la insertará en el anillo (reduce el rendimiento).
- Son importantes tanto el tamaño como la duración de una TR. Una TR q sólo modifica 1 byte durante largo tiempo, puede provocar q un RS se extienda si dicha extensión se necesita más adelante.
- Para cada RS, debemos asegurarnos de q la cabeza no alcance a la cola muy rápido, pues hará q el RS se extienda.

AJUSTE DEL ROLLBACK OPTIMIZACION

- El tamaño de los RS depende directamente de la actividad de las TR. Nos debemos fijar en la actividad "normal" de la BD para ajustar.
- Calcular un segmento de rollback por cada 4 transacciones concurrentes, con extensiones del mismo tamaño (un mínimo de 20 por RS).
- Fijar "optimal" de modo q el 90% de las transacciones quepan en un RS sin extenderse. Una vez fijado "optimal", dejar espacio libre en el tablespace para que, además, se pueda atender a la transacción más grande.
- Para prevenir "ORA-1555 Snapshot too old" (lo provocan las grandes consultas), cuantos más RS mejor (y no sólo más grandes) y del mismo tamaño, y que las transacciones sean lo más cortas posibles. Si no hay updates a la vez que "grandes selects", no hay ora-1555.
- Comprobar la contención en RS consultando V\$WAITSTAT.

```
SELECT CLASS, COUNT FROM V$WAITSTAT  
WHERE CLASS like '%undo%';
```

Si hay en "undo header", es q no hay bastantes RS (crear más).

AJUSTE DE SEGMENTOS DE SORT GENERALIDADES

- Cuando ocurre una ordenación, Oracle reserva tantas extensiones como necesite. Finalizada la ordenación, las extensiones se marcan como libres, pero no se liberan.
- Un mismo segmento de sort puede ser compartido por varias ordenaciones.
- La primera ordenación crea el segmento (si no existía) y las demás añadirán extensiones si lo necesitan.
- Se utiliza una zona de la SGA, fuera de la Shared Pool, llamada Sort Extent Pool; sincronizada usando el latch "sort extent pool" (V\$LATCH). Si tiene contención, subir el tamaño de la extensión por defecto del tablespace.

AJUSTE DE SEGMENTOS DE SORT OPTIMIZACION

- Si hay contención en el latch "sort extent pool" (V\$LATCH), subiremos el tamaño de la extensión por defecto del tablespace.
- Si además, también se producen esperas pq hay muchas ordenaciones concurrentes, se debe incrementar el parámetro SORT_AREA_SIZE, de forma que una mayor cantidad de ordenaciones se hagan en memoria.
- El tamaño de la extensión debe ser igual a SORT_AREA_SIZE (o múltiplo).
- En Oracle9i, usando el parámetro PGA_AGGREGATE_TARGET, me olvido de sort_area_size. PGA_AGGREGATE_TARGET indica el tamaño máximo que sumarán las PGAs (mínimo 10M).
- Podemos monitorizar los segmentos de sort con:
 - V\$SORT_SEGMENT muestra los segmentos temporales. Si las columnas ADDED_EXTENTS y FREED_EXTENTS muestran mucha actividad, añadiremos más espacio al tablespace. Si MAX_SORT_SIZE indica ordenaciones de gran tamaño, es posible q necesitemos un tablespace dedicado para ellas (sólo para las grandes).
 - V\$SORT_USAGE muestra el uso actual de los segmentos temporales.

AJUSTE DE LOS CHECKPOINTS

- Un checkpoint es el evento que sincroniza los bloques de datos en memoria con los ficheros de datos en disco (los escribe el DBWR).
- Indicaciones de ajuste:
 - El proceso CKPT puede mejorar significativamente el rendimiento.
 - Asignando el parámetro LOG_CHECKPOINTS_TO_ALERT a TRUE podremos ver en el fichero alert.log los tiempos de comienzo y fin de los checkpoints.
 - Si el valor de LOG_CHECKPOINT_INTERVAL es mayor que el tamaño del fichero redolog, los checkpoints ocurrirán cuando Oracle haga un “log switch”.
 - Lo ideal es que los checkpoints sólo sucedan cuando se hace “log switch”:
 - log_checkpoint_interval=0
 - log_checkpoint_timeout=0.
 - FAST_START_MTTR_TARGET=0 (desactiva “fast recovery instance”).

AJUSTE DE LAS CONSULTAS SQL

- Si observamos que nuestras consultas SQL van “lentas”, debemos comprobar:
 - El valor de OPTIMIZER_MODE. Ej: si CHOOSE, probar RULE.
`ALTER SYSTEM SET OPTIMIZER_MODE=RULE | FIRST_ROWS | ALL_ROWS | CHOOSE;`
 - ¿Están los índices de las tablas afectadas en estado “valid”?
`SELECT * FROM DBA_OBJECTS WHERE OBJECT_TYPE='INDEX' AND STATUS!='VALID';`
 - ¿Hay alguna otra consulta SQL “pesada” en ejecución?
- Si, además, estamos usando el optimizador por costes (CBO):
 - ¿Están calculadas las estadísticas para las tablas y los índices en cuestión?
`ANALYZE TABLE | INDEX nombre COMPUTE | ESTIMATE STATISTICS;`
Para tablas grandes “Estimate 30%” suele ser casi igual a “compute”.
 - Si hay estadísticas, ¿se generaron con “compute” o con “estimate”?
- Para el ajuste de sentencias SQL utilizaremos las siguientes herramientas:
 - AUTOTRACE de Sql*Plus:
`SQL> SET AUTOT[RACE] {OFF | ON | TRACE[ONLY]} [EXP[LAIN]] [STAT[ISTICS]]`
 - TKPROF sobre un fichero de traza:
`SQL> alter session set sql_trace=true;
SQL> Select ... ;
SQL> alter session set sql_trace=false;
$ tkprof tracefile outputfile [explain=usu/pw] [sys=no]`

AJUSTE DE LA CPU

- Porcentaje máximo de ocupación de CPU = 90%.
- Porcentaje máximo de procesamiento OS/usuario = 40/60%.
- Si hay varias CPUs, la carga debe estar balanceada.

UTILIDAD "STATSPACK"

- STATSPACK es una utilidad para recoger estadísticas que permiten detectar problemas de rendimiento. Además, mejora la utilidad que había anteriormente, UTLBSTAT/UTLESTAT.
- Se trata de un conjunto de scripts SQL y PL/SQL. El script de instalación de la herramienta crea un usuario, PERFSTAT. Los datos recopilados por statspack se guardarán en tablas.
- La documentación de la herramienta están en `$ORACLE_HOME/rdbms/admin/spdoc.txt`.
- STATSPACK trabaja con el concepto "snapshot" para identificar una colección de datos recopilados en un momento del tiempo, a los que asocia un identificador único, `snap_id`. Es necesario recopilar datos en forma de snapshots antes de poder obtener un informe de rendimiento.

INSTALAR "STATSPACK"

- Necesita un tablespace con al menos 100Mb, a ser posible "manejado localmente".
- La instalación crea un usuario PERFSTAT que será el propietario de todos los objetos y el código pl/sql que se cree. Debemos asegurarnos que el tablespace por defecto de este usuario es el que hemos creado, y tener cuidado con su tablespace temporal por defecto (no sería mala idea asignarle uno especial).
- Para la instalación debemos conectarnos a la BD como SYSDBA, y lanzar el script `$ORACLE_HOME/rdbms/admin/spcreate.sql`, que nos pedirá la clave para el nuevo usuario, así como el tablespace por defecto y el tablespace temporal. Este script, de forma automática, llama a otros tres: `spcusr.sql`, `spctab.sql`, `spcpkg.sql`. Debemos comprobar si se producen errores en los ficheros `spcusr.lis`, `spctab.lis`, `spcpkg.lis`.
- Si hay errores y tenemos que repetir la instalación, primero desinstalaremos con el script `spdrop.sql`.

RECOGIENDO DATOS CON "STATSPACK"

- La forma más sencilla de crear un snapshot es conectando como usuario PERFSTAT y ejecutando el procedimiento STATSPACK.SNAP. Previamente es conveniente activar el parámetro time_statistics (TRUE). Este primer snapshot nos servirá de base comparativa para los próximos snapshots que tomemos.
- Al crear un snapshot, con STATSPACK.SNAP, se pueden indicar parámetros para, por ejemplo, indicar el nivel de detalle de las estadísticas (i_snap_level), e incluso una sesión sobre la que recabar estadísticas adicionales (i_session_id). Para establecer un snapshot inicial (baseline) se recomienda i_snap_level=10.
- Podemos borrar un rango de snapshots con sppurge.sql. Si queremos borrar todos los snapshots, podemos hacerlo con sptrunc.sql (¡ojo!, que lo borrará todo, puede ser interesante hacer un export previo del usuario PERFSTAT).

OBTENIENDO INFORMES CON "STATSPACK"

- Podemos obtener un informe sobre el rendimiento general de la instancia con spreport.sql (desde el usuario PERFSTAT). Nos pedirá el intervalo de snapshots (inicial y final) y el nombre del fichero de salida.
- Si localizamos una sentencia SQL por su "hash_value" (V\$SQLAREA.HASH_VALUE), podemos generar un informe sobre ella con el script sprepsql.sql; que nos pedirá el intervalo de snapshots y el hash_value de la sentencia.

RESUMEN DE SCRIPTS "STATSPACK"

Instalación (como usuario SYSDBA):

spcreate.sql -> Instala STATSPACK ejecutando a su vez los scripts:

spcusr.sql -> Crea el usuario PERFSTAT

spctab.sql -> Crea las tablas

spcpkg.sql -> Crea el paquete statspack

spdrop.sql -> Desinstala STATSPACK ejecutando a su vez los scripts:

spdtab.sql -> Borra las tablas

spdusr.sql -> Borra el usuario PERFSTAT

Informes (como usuario PERFSTAT):

spreport.sql -> Genera un informe general del rendimiento de la instancia

sprepins.sql -> Genera un informe para la BD y la instancia indicados

sprepsql.sql -> Genera un informe para la sentencia SQL cuyo Hash Value se indique

spauto.sql -> Permite automatizar la recolección de estadísticas (usando dbms_job)

Mantenimiento (como usuario PERFSTAT):

sppurge.sql -> Permite borrar un rango de snapshots

sptrunc.sql -> Vacía (con truncate) todas las tablas, borrando todos los snapshots

spuexp.par -> Es un fichero de parámetros para exportar el usuario PERFSTAT

VISTAS DEL DD

- V\$SYSSTAT: estadísticas generales del sistema.
- V\$SGASTAT: estado de uso de las diferentes partes de la SGA.
- V\$SYSTEM_EVENT: estadísticas de eventos de espera a nivel de sistema.
- V\$SESSION_WAIT: sesiones que están en espera actualmente.
- V\$SESSION_EVENT: estadísticas de eventos de espera por sesión.
- V\$SESSTAT: estadísticas generales por sesión.
- V\$LIBRARYCACHE: ratios de la Library Caché (Sentencias SQL y PL/SQL).
- V\$ROWCACHE: ratios de la Row Caché (Diccionario de Datos).
- V\$PGASTAT: estado de la PGA.
- V\$BUFFER_POOL_STATISTICS: estadísticas de la caché de datos.
- V\$DB_OBJECT_CACHE: objetos que hay en la Library Caché.
- V\$LATCH: latches.
- V\$ROLLSTAT: estadísticas de uso de los segmentos de rollback.
- V\$UNDOSTAT: estadísticas de uso de los segmentos de rollback.
- V\$FILESTAT: estadísticas de E/S a nivel de fichero.
- V\$SESS_IO: estadísticas de E/S a nivel de sesión.
- V\$SORT_SEGMENT: estadísticas de uso de los segmentos temporales.
- V\$SORT_USAGE: áreas de sort actualmente en uso.
- Obtener IP: `Select SYS_CONTEXT('USERENV' , 'IP_ADDRESS') FROM DUAL;`

PRACTICAS TEMA 4

- 4.1. Revisar eventos del sistema y comprobar los más significativos. Comprobar evento "latch free" por sesiones. Comprobar sesiones esperando por el evento "db file sequential read". Comprueba el parámetro TIMED_STATISTICS.
- 4.2. Revisa las estadísticas del sistema más significativas. Comprueba el tamaño medio de la PGA de cada sesión. Revisa las lecturas lógicas y físicas y calcula el ratio de E/S. Comparar el uso de CPU para "SQL del sistema" (acceso al DD) sobre el total.
- 4.3. Ver el estado de ocupación de las partes más significativas de la Shared Pool.
- 4.4. Comprobar la contención en latches de la Shared Pool y Library Cache.
- 4.5. Comprobar el pinhitratio de la Library Caché, así como los reloads. Verificar el espacio libre de la Shared Pool, y el valor de open_cursors.
- 4.6. Ver el ratio de la Row Cache.
- 4.7. Comprobar si el sistema recomienda ampliar la SharedPool.

PRACTICAS TEMA 4

- 4.8. Detectar sentencias similares que usan literales. Verificar el parámetros `cursor_sharing` y `session_cached_cursors`. Asignar `cursor_sharing=similar`.
- 4.9. Instalar el paquete `DBMS_SHARED_POOL`. Comprobar paquetes que se pueden "fijar" en la SharedPool y hacerlo.
- 4.10. Comprobar sentencias que ocupan mucha memoria ($\geq 10\%$ de SharedPool).
- 4.11. Calcular el ratio de eficiencia de la Caché de Datos. Comprobar el parámetro `db_cache_advice`. Consultar si Oracle recomienda incrementar la Caché de Datos. Comprobar si hay contención en el latch "cache buffers lru chain". Ver si hay esperas del tipo "write complete waits" o "free buffer waits".

PRACTICAS TEMA 4

- 4.12. Comprobar el tamaño de la Cache de Redo. Ver si hay contención en los latches de redo. Verificar la estadística "redo log space requests".
- 4.13. Comprobar los segmentos de rollback ONLINE. Verificar las extensiones que tiene cada uno, así como el espacio total y libre del tablespace que los contiene. Ver si hay contención en segmentos de rollback.
- 4.14. Instalar la utilidad STATSPACK. Crear snapshots y generar informe.

TEMA 5

SOPORTE DE GLOBALIZACIÓN

TEMA 5.

SOPORTE DE GLOBALIZACIÓN

- Conjunto de caracteres para la base de datos.
- Lenguaje y territorio. Variable de entorno NLS_LANG.
- Parámetros NLS. NLS_SESSION_PARAMETERS.
 - NLS_TERRITORY
 - NLS_LANGUAGE
 - NLS_DATE_LANGUAGE

CONJUNTO DE CARACTERES

- El conjunto de caracteres para la BD a utilizar en España, es el WE8ISO8859P15 (que incluye el símbolo del euro).
- La variable de entorno NLS_LANG, permite indicar (desde la aplicación cliente que accede a Oracle) tanto el juego de caracteres a utilizar, como el país y el idioma:
 - export NLS_LANG=SPANISH_SPAIN.WE8ISO8859P15

NLS_SESSION_PARAMETERS

Para consultar las variables NLS asociadas a una sesión:

```
SQL> select * from nls_session_parameters;
```

PARAMETER	VALUE
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_CURRENCY	\$
NLS_ISO_CURRENCY	AMERICA
NLS_NUMERIC_CHARACTERS	P . ,
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD-MON-RR
NLS_DATE_LANGUAGE	AMERICAN
NLS_SORT	BINARY
NLS_TIME_FORMAT	HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZH:TZM
NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZH:TZM
NLS_DUAL_CURRENCY	\$
NLS_COMP	BINARY

Se pueden cambiar con "ALTER SESSION SET variable=valor;"

NLS_TERRITORY Y NLS_LANGUAGE

El país (NLS_TERRITORY) lleva asociado un formato de fecha, y el lenguaje (NLS_LANGUAGE) se refiere al idioma en que nos responde Oracle.

```
SQL> SELECT SYSDATE FROM DUAL;  
SYSDATE  
-----  
30-MAR-05
```

```
SQL> ALTER SESSION SET NLS_TERRITORY=SPAIN;  
Session altered.
```

```
SQL> SELECT SYSDATE FROM DUAL;  
SYSDATE  
-----  
30/03/05
```

```
SQL> ALTER SESSION SET NLS_LANGUAGE=SPANISH;  
Sesion modificada.
```

NLS_DATE_LANGUAGE

Podemos modificar el idioma de la fecha con NLS_DATE_LANGUAGE:

```
SQL> SELECT TO_CHAR(SYSDATE,  
    'DD/MON/YYYY' , 'NLS_DATE_LANGUAGE=SPANISH' ) FROM DUAL;
```

```
TO_CHAR(SYS  
-----  
20/DIC/2004
```

```
SQL> SELECT TO_CHAR  
    (SYSDATE, 'DD/MON/YYYY' , 'NLS_DATE_LANGUAGE=AMERICAN' ) FROM DUAL;
```

```
TO_CHAR(SYS  
-----  
20/DEC/2004
```

VISTAS DEL DD

- NLS_SESSION_PARAMETERS

PRACTICAS TEMA 5

- 5.1. Consultar NLS_SESSION_PARAMETERS.
- 5.2. Asignar las variables NLS necesarias para ajustar el país y el lenguaje a España y Español, respectivamente.
- 5.3. Convertir una fecha a "char", indicando primero el Español como lenguaje, y después el Americano.

APENDICE A.

Recursos Oracle en Internet.

- www.orafaq.org (Underground Oracle FAQs)
 - Sitio no oficial sobre Oracle (FAQs, foros, artículos, scripts, etc).
- otn.oracle.com (Oracle Technology Network)
 - Descargas de sw, documentación, foros, artículos, scripts, etc. Registro gratuito.
- otn.oracle.com/oramag (Oracle Magazine)
 - Revista Oracle Magazine.
- www.oracle.com (Web de Oracle)
 - Portal oficial de Oracle.
- metalink.oracle.com (Soporte Técnico Oracle)
 - Soporte técnico para usuarios con contrato de mantenimiento.
- www.dbazine.com (Revista electrónica)
 - Revista electrónica mensual especializada en Oracle
- asktom.oracle.com (Gurú de Oracle)
 - Artículos y preguntas a uno de los gurús de Oracle
- www.ixora.com.au (Otro gurú)
 - Para mejorar el rendimiento de Oracle sobre Unix

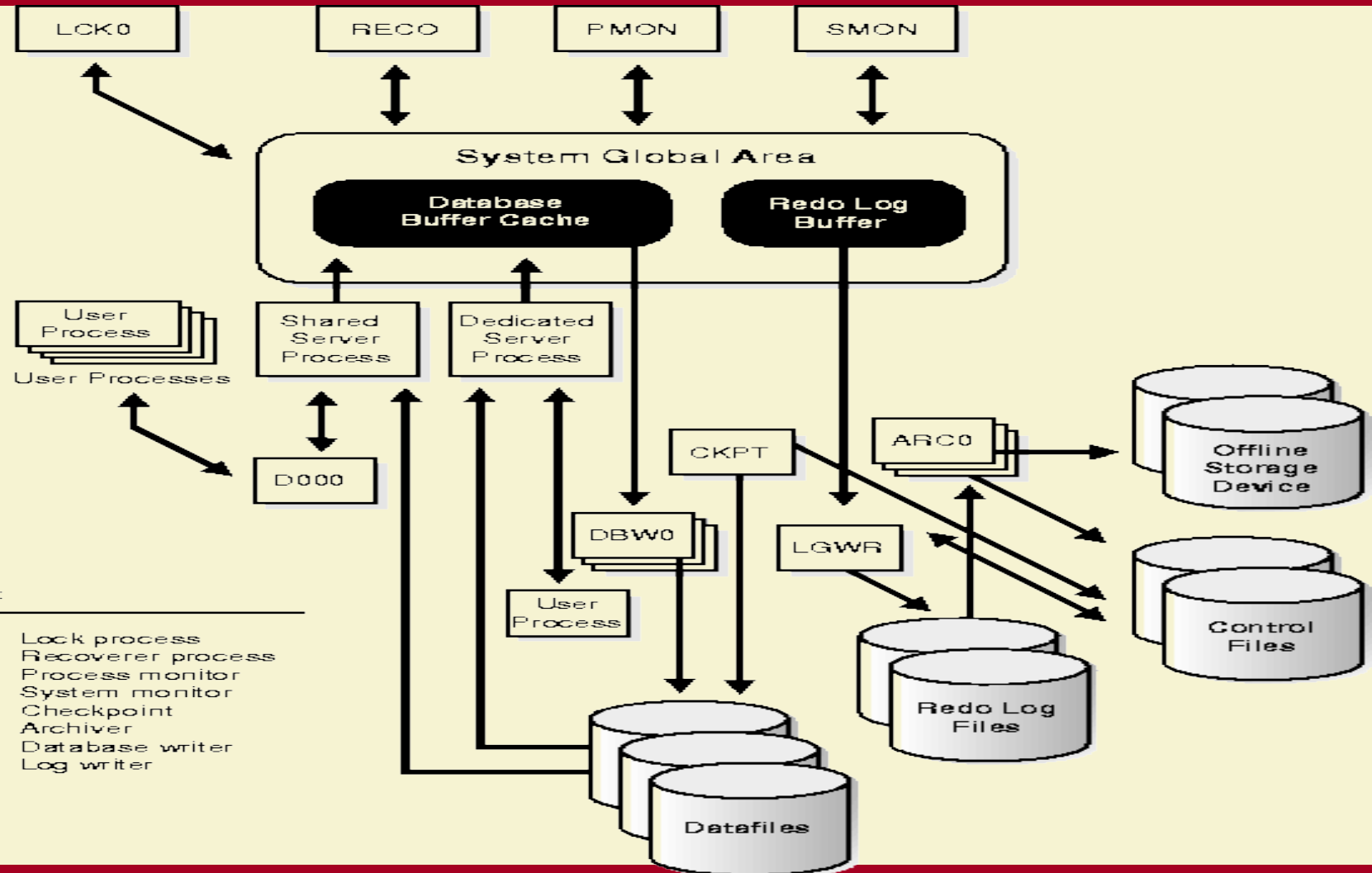
APENDICE B.

Arquitectura Oracle

Servidor Oracle

- Instancia
 - Procesos Background
 - DBWR
 - LGWR
 - SMON
 - PMON
 - CKPT
 - RECO, ARCH, Dispatchers, Servers, etc.
 - System Global Area (SGA)
 - Shared Pool
 - » Library Cache
 - » Dictionary Cache
 - Database Buffer Cache
 - Redo Log Buffer
 - Java Pool y Large Pool.
- Base de datos
 - Ficheros de datos
 - Ficheros redo log
 - Ficheros de control

APENDICE B. Arquitectura Oracle



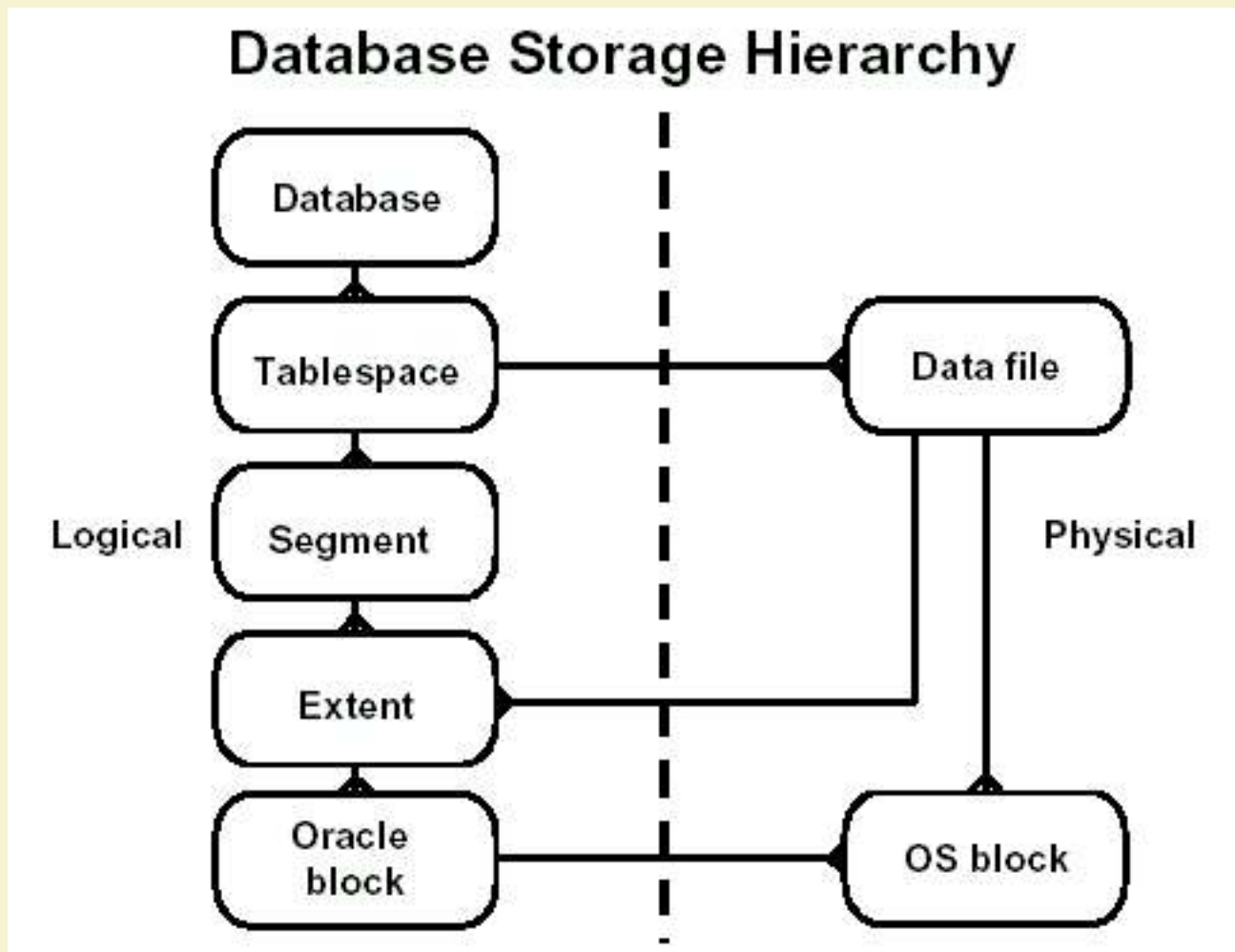
APENDICE C.

Estructura Física/Lógica de la BD

- Estructura lógica de la BD
 - Tablespaces
 - Segmentos
 - Extensiones
 - » Bloques Oracle
- Estructura física de la BD
 - Ficheros de datos
 - Bloques del SO

APENDICE C.

Estructura Física/Lógica de la BD

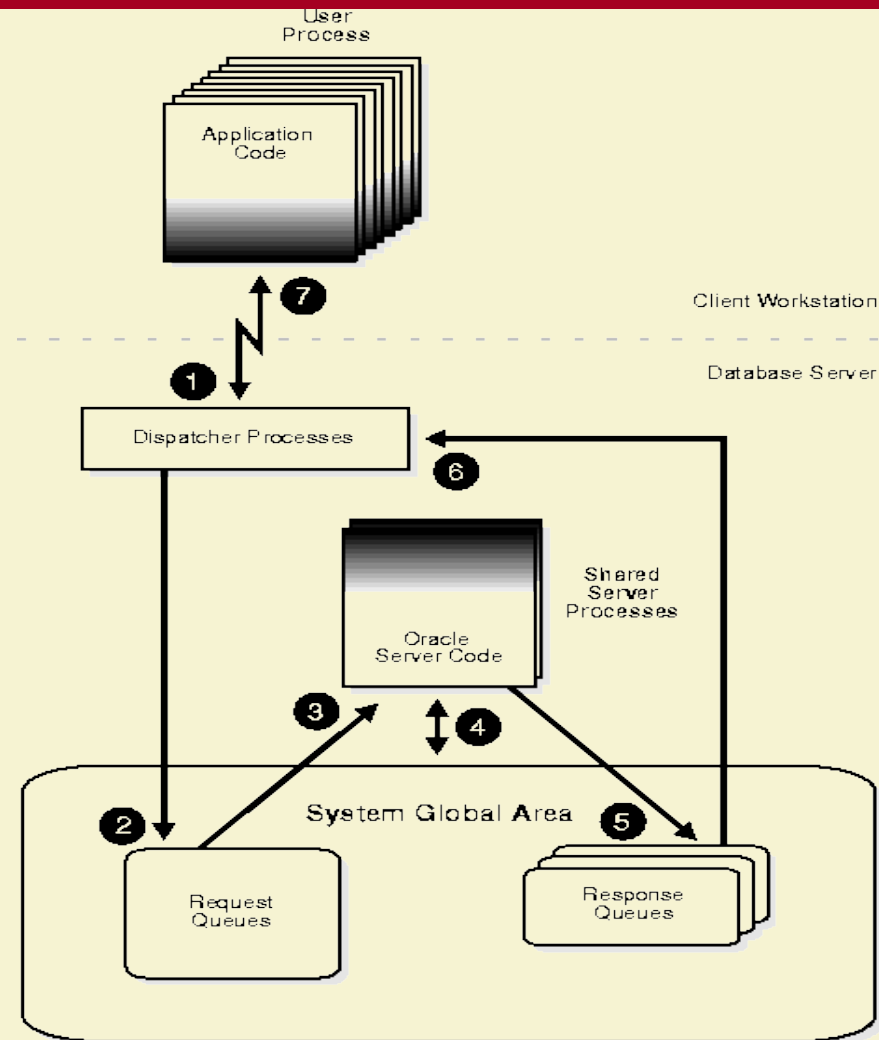
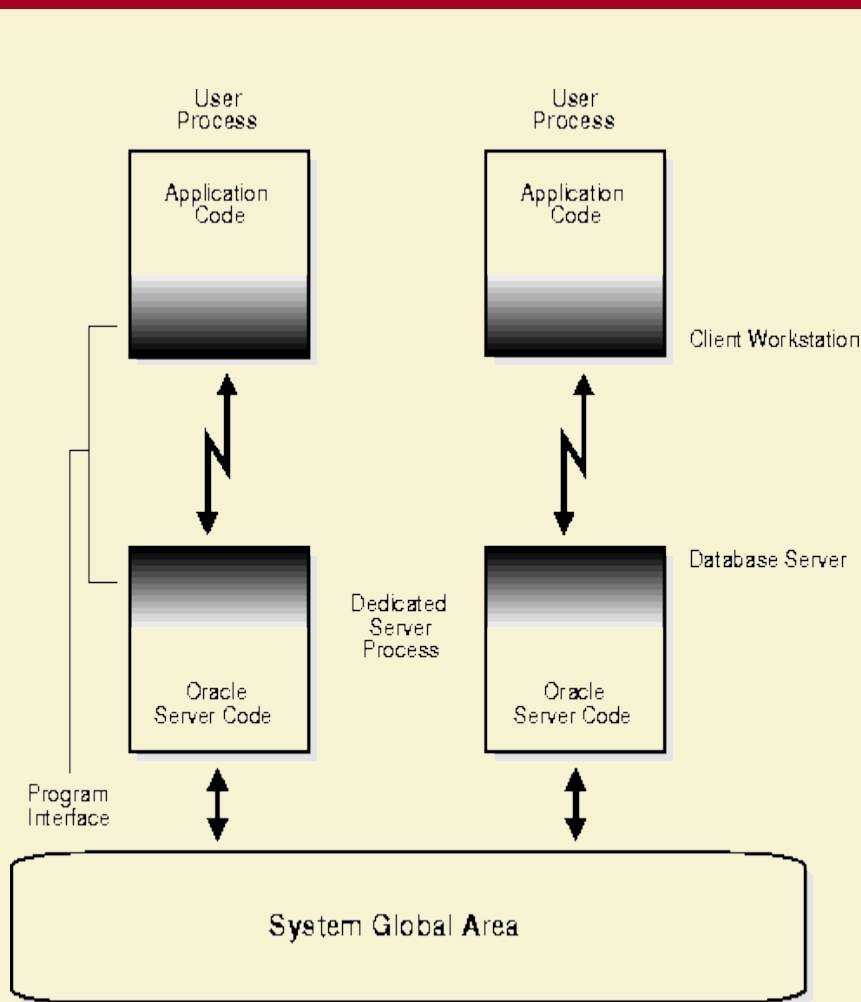


APENDICE D.

Conexión a la BD y consulta

- Proceso de Usuario: es la aplicación cliente que inicia la conexión (p.e. Programa de Facturación q se arranca en un PC).
 - Envía las sentencias SQL al proceso servidor para su ejecución,
 - y muestra en pantalla los datos devueltos por el mismo.
- Proceso Servidor: se arranca uno (en el servidor) para cada sesión que se conecta a la BD (servidor dedicado).
 - Compila y ejecuta las sentencias SQL q le llegan desde el proceso de usuario (las busca en la Shared Pool y si ya está compilada, la ejecuta directamente).
 - Accede a los datos en la Database Buffer Cache (y si no los encuentra, en los ficheros de datos, llevándolos a las Database Buffer Cache).
 - Devuelve los datos al proceso de usuario q inició la conexión.

APENDICE D. Conexión a la BD



APENDICE E.

Actualización de datos

- El proceso servidor de la sesión busca los bloques de datos a modificar en la caché de datos. Si no los encuentra, los buscará directamente en los ficheros de datos de la BD.
- Se guarda una copia del dato (antes del cambio) en un segmento de Rollback (se mantiene hasta q se haga Commit o Rollback).
- Se modifican los bloques de datos en la Database Buffer Caché (caché de datos). El DBWR los llevara a los ficheros de datos cuando suceda un checkpoint (se llena el buffer de redo al 90%).
- Se guardan en el Buffer Redo Log las "redo entries" (vector de cambios de cada bloque modificado) necesarias para registrar el cambio q se va a hacer (el LGWR vuelca el buffer al fichero redo log activo, cuando se hace commit o cada 3 segundos).