

Monitorización y ajuste del rendimiento

Administración de Bases de Datos

Curso 2018-2019

Jesús Correas, Mercedes G. Merayo, Yolanda García

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid



Sobre este tema

- Estas transparencias están basadas en los siguientes documentos:
 - ▶ J. Watson [OCA Oracle Database 11g: Administration I Exam Guide \(Exam 1Z0-052\)](#). Oracle Press - McGraw-Hill, 2008.
 - ▶ B. Thomas [OCA Oracle Database 11g Administrator Certified Associate Study Guide](#). Wiley Publishing, 2009.

Monitorización y ajuste del rendimiento

- Es necesario realizar tareas de **mantenimiento** de una BD.
 - ▶ La **calidad de los índices se va degradando** cuando se modifican las tablas y se pueden hacer inutilizables.
 - ▶ El uso de la BD puede ir variando y deben ajustarse los **tamaños de ficheros de datos y estructuras en memoria**.
 - ▶ La **eficiencia de las consultas** depende de información estadística reciente y fiable.
- Hay dos formas fundamentales de monitorización:
 - ▶ **reactiva**: se realiza cuando hay problemas de funcionamiento de la BD;
 - ▶ **proactiva**: se realiza de forma preventiva, antes de que ocurran problemas en la BD.
- Herramientas que se proporcionan para monitorizar una BD:
 1. Estadísticas del optimizador, almacenadas en el **diccionario de datos**.
 2. Estadísticas de rendimiento (**AWR**, *Automatic Workload Repository*).
 3. **Advisory Framework**.
 4. Alertas y umbrales.
- Además se genera el **Automatic Diagnostic Repository** que se utiliza para mtmto. reactivo.

1. Estadísticas del optimizador de SQL

- El **optimizador de consultas** SQL de Oracle calcula para cada consulta el **plan de ejecución más eficiente**.
- Utiliza **información estadística** para calcular el **coste** de cada plan.
- Utiliza datos almacenados en el **diccionario de datos** de distintos objetos de la BD:
 - ▶ Tablas (vista **DBA_TABLES**): número de filas, núm. bloques (utilizados y no utilizados nunca), espacio libre en bloques utilizados, longitud media de las filas, filas partidas en distintos bloques.
 - ▶ Columnas (**DBA_TAB_COLUMNS**): número de valores distintos, mínimo y máximo, número de nulos, longitud media.
 - ▶ Índices (**DBA_INDEXES**): profundidad del árbol, número de valores distintos, *factor de agrupamiento* (relación entre el orden físico de las filas y el orden de las entradas).
 - ▶ Índices (**INDEX_STATS**): entradas de índice de filas existentes/borradas.
- Si esta información no existe o es incorrecta, **el rendimiento se puede degradar**.

1. Estadísticas del optimizador de SQL

- La información estadística no se obtiene en tiempo real, porque haría más lentas las consultas: se debe ejecutar *periódicamente*.
- Lo que se necesita es que la información sea *suficientemente precisa*:
 - ▶ Que tenga una información lo más precisa posible, pero no calculada *en tiempo real*.
 - ▶ Si no hay mucha variación en los datos de una tabla/schema, se puede hacer con menos frecuencia.
 - ▶ También se puede indicar un porcentaje que indica **cuántas filas se analizan** de las tablas para **estimar** la información estadística.
- Se puede recoger información estadística en tres niveles según el parámetro **STATISTICS_LEVEL**: BASIC, TYPICAL, ALL.
- La información estadística **se puede generar manualmente**:
 - ▶ Comando ANALYZE, Ej: ANALYZE TABLE... COMPUTE STATISTICS;
 - ▶ Mediante procedimientos PL/SQL del paquete DBMS_STATS, o
 - ▶ Mediante *Enterprise Manager Database Control*.

1. Estadísticas del optimizador de SQL

- Lo normal es que el análisis se ejecute diariamente como una *tarea automática de mantenimiento* del **planificador de trabajos de Oracle** (*Scheduler*).
- El planificador define **ventanas (de tiempo) de mantenimiento**: horario en el que se pueden lanzar tareas que consumen recursos del sistema. Por defecto:
 - ▶ 4 horas de lunes a viernes (desde las 22:00),
 - ▶ 20 horas sábados y domingos (desde las 06:00).
- El *manejador de recursos* (**Resource Manager**) garantiza que las **ventanas de mantenimiento** consumen como máximo un porcentaje de los recursos a estas tareas.
- Por defecto, algunos *advisors* también están planificadas como *tareas automáticas de mantenimiento*:
 - ▶ Las estadísticas del optimizador de SQL,
 - ▶ SQL *advisor* y
 - ▶ *Segment advisor*.

2. Estadísticas de rendimiento: AWR

- La instancia de Oracle genera una gran cantidad de información sobre el rendimiento del sistema.
- Normalmente la información se acumula en la SGA.
- periódicamente se escribe en el tablespace `SYSAUX` en el *Automatic Workload Repository*, **AWR**.
- Por defecto, se guarda un **snapshot** en disco **cada 60 minutos** por el proceso *background* **MMON**.
- La información de los *snapshots* se guarda por defecto durante 8 días.
- Proporcionan información sobre la **evolución del estado de los objetos de la BD**.

2. Estadísticas de rendimiento: AWR

- Utiliza el mismo parámetro de configuración que las estadísticas de optimización: **STATISTICS_LEVEL** (BASIC, TYPICAL, ALL).
- **¿Qué información proporciona AWR?** AWR contiene datos estadísticos que nos proporciona métricas como:
 - ▶ Accesos a disco por segundo, por transacción, por sesión.
 - ▶ Commits o rollbacks por segundo, por sesión.
 - ▶ Tiempo de ejecución de SQL por segundo/ transacción/ sesión, etc.
- Se pueden definir determinados (pares de) *snapshots* como referencia para comparar el funcionamiento de la BD: se denominan **baseline**.
- Se pueden guardar **baselines** para distintos escenarios: carga diaria del sistema, cierre mensual, etc.

2. Estadísticas de rendimiento: AWR

- Para consultar la información de AWR lo mejor es utilizar **Enterprise Manager**.
- La información resultante de AWR no se puede consultar directamente desde `SQL*Plus` mediante vistas (solo la actual).
- Para el mantenimiento de AWR sí se pueden utilizar procedimientos del paquete **DBMS_WORKLOAD_REPOSITORY** y algunas vistas.
- **Ejemplos:**

- ▶ para mostrar los snapshots:

```
select snap_id, begin_interval_time  
from dba_hist_snapshot order by snap_id desc;
```

- ▶ para crear un snapshot:

```
exec dbms_workload_repository.create_snapshot;
```

- ▶ para generar el informe entre dos snapshots:

```
$ORACLE_HOME/rdbms/admin/awrrpt.sql
```

3. Advisory Framework

- Oracle incluye una serie de *advisors* (asesores), programas y opciones de *Enterprise Manager* que evalúan el estado de la BD.
- Destaca **Automatic Database Diagnostic Monitor (ADDM)**.
 - ▶ Consultar los informes que genera ADDM debe ser parte de las tareas habituales del administrador.
 - ▶ Detecta problemas en la BD y sugiere soluciones.
- Para utilizar los *advisors* desde *Enterprise Manager*: enlace **Advisor Central** desde **Related Links**.
- Los *advisors* son:
 - ▶ ADDM (*Automatic Database Diagnostic Monitor*)
 - ▶ SQL *advisor*
 - ▶ *Automatic undo advisor*
 - ▶ MTTR *advisor*
 - ▶ *Data Recovery Advisor*
 - ▶ *Segment advisor*
 - ▶ *Memory advisors*

3. *Advisory Framework*: ADDM

- Este *advisor* es el punto de inicio para analizar el rendimiento y resolver problemas en una BD.
- ADDM se ejecuta automáticamente cada vez que se toma un *snapshot*.
- Utiliza la información de AWR entre el *snapshot* actual y el anterior (se puede modificar). Se elimina a los 30 días.
- ADDM puede realizar recomendaciones:
 - ▶ Cambios de *hardware*.
 - ▶ Configuración de la BD.
 - ▶ Cambios del schema, de la aplicación.
 - ▶ Es habitual que recomiende la ejecución de otros *advisors*.

3. Advisory Framework: SQL advisors

Hay tres *advisors*:

- **SQL access advisor:** A partir de la carga generada por sentencias SQL sugiere la modificación de objetos de la BD.
 - ▶ Creación o eliminación de índices, vistas materializadas, uso de segmentos particionados, etc.
- **SQL tuning advisor:** Analiza sentencias individuales.
 - ▶ Como el anterior, puede recomendar cambios en el schema.
 - ▶ Puede solicitar la generación de otras estadísticas para optimizar SQL
- **SQL repair advisor:** Analiza los casos en los que ocurren errores internos de Oracle para forzar la utilización de planes de ejecución seguros.

3. Advisory Framework: Automatic Undo Advisor

- El tablespace de *undo* debe estar dimensionado para garantizar la consistencia de consultas muy largas:
 - ▶ Las consultas deben utilizar el estado de los datos **en el instante de inicio de la consulta**.
 - ▶ Si es necesario, deben utilizar información de *undo*.
 - ▶ Si el tablespace de *undo* no es suficiente y se reutilizan los bloques de *undo* antes de terminar la consulta, se puede producir un error “*Snapshot too old*”.
- Este *advisor* utiliza la información estadística de generación de datos de *undo* y la longitud de las consultas para **recomendar un tamaño mínimo del tablespace de undo**.

3. *Advisory Framework: MTTR, Data recovery advisors*

- **MTTR advisor:**

- ▶ Hemos visto anteriormente que se puede fijar un tiempo límite para la recuperación de una caída de la instancia.
- ▶ La instancia de Oracle escribe en disco bloques modificados en la SGA, aunque haya espacio disponible en la SGA, para alcanzar el MTTR.
- ▶ Este *advisor* calcula el tiempo de recuperación según la carga de trabajo actual de la BD.

- **Data recovery advisor (DRA):**

- ▶ Si los ficheros de la BD se han dañado (datafiles, controlfile), se deben recuperar de un backup.
- ▶ Hay que determinar qué ficheros se han dañado y en qué bloques.
- ▶ También debe determinarse qué backup utilizar.
- ▶ DRA utiliza un sistema experto para proporcionar la mejor alternativa.

3. Advisory Framework: Segment advisor, Memory advisor

- **Segment advisor:**

- ▶ Los segmentos de datos (tablas, índices) crecen automáticamente al insertar nuevos datos.
- ▶ Pero no disminuyen cuando se borran o se modifican
- ▶ **Segment advisor** estudia su estado actual y las estadísticas históricas de los objetos de la BD y recomienda la reorganización cuando sea necesario.
- ▶ Por defecto este *advisor* está planificado como una *tarea automática de mantenimiento*.

- **Memory advisor:**

- ▶ Predicen el efecto de la modificación del tamaño dedicado a las estructuras en memoria de SGA y PGA:
 - ★ El tiempo de procesamiento que se puede reducir, y
 - ★ La actividad de disco.
- ▶ Se utilizan para **ajustar el rendimiento** de la BD.

4. Alertas y umbrales

- Una parte importante del trabajo de un administrador consiste en comprobar el estado de la BD para detectar lo antes posible situaciones excepcionales.
- Actualmente Oracle dispone de un sistema de **alertas y umbrales** que permiten automatizar gran parte de este trabajo.
- Por ejemplo, un caso típico es la **monitorización del espacio en los tablespaces**.
 - ▶ Hay que evitar que los tablespaces se llenen completamente (o que ocupen todo el disco) porque se producen errores en las consultas SQL.
 - ▶ Esto supone vigilar periódicamente el espacio ocupado para tomar medidas cuando se aproximen al límite.
 - ▶ Tradicionalmente, los administradores programaban utilidades para hacer esto o compraban sw externo.
 - ▶ Pero hay muchos casos particulares y condiciones que deben tenerse en cuenta.

4. Alertas y umbrales

- Hay dos tipos de alertas:
 - ▶ **Stateful:** producidas por condiciones que persisten en el tiempo y se pueden corregir (espacio en disco, sesiones bloqueadas, tiempo medio de ejecución de SQL...).
 - ▶ **Stateless:** Basadas en eventos que se producen de forma no persistente (interbloqueos, “*snapshot too old*”).
- El funcionamiento básico es el siguiente:
 - ▶ Primero se deben fijar **umbrales** (*thresholds*). Se almacenan en el AWR.
 - ▶ El proceso *background* `MMON` **monitoriza la base de datos en tiempo real y lanza las alertas que detecte.**
 - ▶ El mecanismo de alertas consiste en poner un mensaje en la **cola de alertas** (una tabla de mensajes).
 - ▶ Por defecto, *Enterprise Manager* desencola el mensaje y lo muestra en la página de inicio. Puede configurarse para enviar email o SMS.
 - ▶ Se pueden consultar las alertas en la vista `DBA_OUTSTANDING_ALERTS`.

4. Alertas y umbrales

- Hay más de 200 métricas sobre las que se pueden fijar umbrales:
`SELECT METRIC_NAME FROM V$METRICNAME;`
- Se pueden fijar umbrales mediante *Enterprise Manager (Metric and Policy Settings)* en la sección *Related Links* de la página de inicio).
- También mediante el paquete `DBMS_SERVER_ALERT` (procedimiento `set_threshold`).
- Algunos parámetros que se deben establecer:
 - ▶ Dos valores: el valor para que se produzca una alerta de tipo **warning** y el de una alerta de tipo **critical**.
 - ▶ El periodo de observación (en minutos).
 - ▶ El número de observaciones consecutivas para que se dispare la alerta.
- En algunos casos hay parámetros que no son aplicables.

4. Alertas y umbrales

- El sistema de notificación por defecto solo muestra las alertas en la página de inicio de *Enterprise Manager* y añade entradas en `DBA_OUTSTANDING_ALERTS`.
- Permanecen ahí hasta que se elimine el problema que lo ocasiona.
- Cuando se elimina una alerta, se pasa a la vista `DBA_ALERT_HISTORY`.
- Se pueden añadir otras notificaciones. Se deben configurar tres elementos:
 - ▶ Se debe configurar un **método de notificación**. Por ejemplo, por correo electrónico (servidor SMTP, etc.)
En *Enterprise Manager: Setup / Notification Methods*.
 - ▶ Se debe crear una **regla** para capturar el evento.
En *Enterprise Manager: Preferences / Rules*.
 - ▶ Un usuario administrador debe **suscribirse** a la regla.
En *Enterprise Manager: Setup / Administrators*.

Ajuste del rendimiento

Ajuste del rendimiento

- Hay dos aspectos que son fundamentales para ajustar el rendimiento de una BD:
 - ▶ El uso de la memoria del servidor.
 - ▶ El estado de diversos objetos de la BD.
- Respecto a la memoria, las últimas versiones de Oracle permiten **automatizar la gestión** tanto de la SGA como la de los procesos de servidor PGA.
- Todos los sistemas operativos de servidor utilizan **memoria virtual**:
 - ▶ Realizan paginación de memoria en disco de forma transparente a los procesos.
 - ▶ La instancia de Oracle **no puede saber** si la memoria que se utiliza está en RAM o en disco.
 - ▶ **La paginación puede reducir mucho la eficiencia y debe evitarse.**

Gestión automática de memoria: PGA

- Una sesión de usuario está formada por un proceso de usuario (en la máquina cliente) conectado a un proceso de servidor.
- La memoria PGA (*Program Global Area*) del proceso servidor se utiliza para:
 - ▶ Almacenar tablas temporales,
 - ▶ Ordenación de filas,
 - ▶ Mezcla de *bitmaps*,
 - ▶ Variables y la pila de llamadas.
- Todas las sentencias SQL necesitan utilizar la PGA, en distinto grado.
- Dependiendo de la cantidad de memoria PGA disponible, la asignación de memoria a una sentencia SQL puede ser de tres tipos:
 - ▶ **Optimal:** si puede contener toda la información necesaria para ejecutar la sentencia SQL.
 - ▶ **One-pass:** si la memoria no es suficiente y debe realizar una pasada adicional sobre los datos de la sentencia SQL.
 - ▶ **Multipass:** si debe realizar múltiples pasadas sobre los datos.

Gestión automática de memoria: PGA

- **Ejemplo:** Para ordenar un conjunto de filas:
 - ▶ La asignación es **óptima** si todas las filas caben en la PGA.
 - ▶ Si no, es **one-pass** si se pueden ordenar grupos de filas en memoria y después se realiza una pasada adicional para mezclar los grupos ordenados.
 - ▶ Si no, es **multipass** si tanto la ordenación como el mezclado necesitan acceso a disco.
- Si se van a ordenar 10Gb de datos:
 - ▶ La asignación óptima consume 10Gb en memoria.
 - ▶ Con 40Mb se puede ejecutar con una pasada adicional.
 - ▶ Con menos memoria se debe hacer en múltiples pasadas.
- Muchas veces no se puede utilizar la asignación óptima.
- **Pero la asignación multipass se debe evitar siempre que sea posible.**

Gestión automática de memoria: PGA

- La instancia de Oracle puede gestionar automáticamente la PGA, con el parámetro `WORKAREA_SIZE_POLICY` (con valor `AUTO`).
- A continuación, se fija un objetivo máximo para el tamaño de la PGA de **todas las sesiones** (parámetro `PGA_AGGREGATE_TARGET`).
- La instancia de la BD distribuye la memoria total de PGA entre las distintas sesiones **bajo demanda**.
- Cuando una sesión deja de utilizar memoria de PGA, se puede asignar a otra sesión.
- Por defecto, el total de PGA es el máximo entre 10Mb y el 20 % de la SGA, pero **se debe ajustar para mejorar el rendimiento**.
- Para ajustarlo se utilizan los **memory advisors**.

Gestión automática de memoria: SGA

- La **SGA** contiene varias estructuras de datos:
 - ▶ **Shared pool**
 - ▶ **DB Buffer cache**
 - ▶ **Large pool**
 - ▶ **Streams pool**
 - ▶ **Java pool**
 - ▶ **Log buffer**
- Se pueden dimensionar independientemente, pero se recomienda habilitar la gestión automática (parámetro `SGA_TARGET`).
 - ▶ Automáticamente se redimensionan los componentes que necesitan más memoria.
- También se pueden gestionar SGA y PGA automáticamente con un solo parámetro: `MEMORY_TARGET`.
 - ▶ En este caso, la instancia transfiere memoria entre SGA y PGAs automáticamente.
- Incrementar innecesariamente la SGA puede tener un **efecto negativo en el rendimiento**.

Uso de *memory advisors*

- Los **memory advisors** utilizan la información en el AWR para calcular el efecto de cambiar el tamaño de SGA y PGA en el rendimiento.
- Para activar *memory advisors* deben generarse estadísticas en AWR (mediante el parámetro `STATISTICS_LEVEL`).

- Se puede estudiar el efecto de modificar la PGA con la consulta:

```
SELECT pga_target_factor, estd_extra_bytes_rw  
FROM v$pga_target_advice;
```

- ▶ `pga_target_factor` es el factor de incremento de la PGA.
- ▶ `estd_extra_bytes_rw` es una estimación del acceso a disco necesario.

- Respecto a la SGA:

```
SELECT sga_size_factor, estd_db_time FROM v$sga_target_advice;
```

- ▶ `sga_size_factor` es el factor de incremento de la SGA.
- ▶ `estd_db_time` es una estimación del tiempo de ejecución de SQL.

- El objetivo es minimizar `estd_extra_bytes_rw` y `estd_db_time`.

Estado de los objetos de la BD

- El estado de algunos objetos de la BD **puede afectar al rendimiento.**
- Algunos objetos de la BD pueden estar en un estado incorrecto:
 - ▶ Los objetos PL/SQL y las vistas pueden quedar en estado **inválido.**
 - ▶ Los índices se pueden ser **inutilizables.**
- Si un procedimiento PL/SQL utiliza una tabla o columna y esta se modifica después de compilar el procedimiento, queda en estado **inválido.**
 - ▶ Si se intenta utilizar, se procede a recompilarlo.
 - ▶ Pero la compilación **puede no terminar correctamente.**
- Lo mismo puede ocurrir con las vistas.
- Se pueden identificar los objetos inválidos de la BD:

```
SELECT object_name,object_type FROM dba_objects  
WHERE status='INVALID';
```
- Es conveniente recompilar los objetos afectados por cualquier cambio en la BD.

Estado de los objetos de la BD

- Los índices son pares que contienen el valor de la clave y el *rowid* de la fila.
- El *rowid* es la posición física de la fila.
- Si la fila cambia de posición, el índice se marca como **inutilizable**.
- Cuando se ejecuta una sentencia SQL, se utiliza un plan de ejecución que no utilice índices inutilizables. **Esto puede afectar al rendimiento.**
- Se pueden identificar los índices inutilizables:

```
SELECT owner,index_name FROM dba_indexes  
WHERE status='UNUSABLE';
```
- Para recrear el índice: `ALTER INDEX ... REBUILD`
- La **reconstrucción de índices** también se debe realizar en el funcionamiento normal: **se van haciendo ineficientes** cuando hay muchos borrados o modificaciones de los datos de la tabla.