

# **ANÁLISIS DEL RENDIMIENTO EN ORACLE**

**Alejandro Campos Paredes**

# ÍNDICE

<b>VISTAS DINÁMICAS DE RENDIMIENTO-----</b>	<b>3</b>
<b>STATSPACK-----</b>	<b>5</b>
<b>INCIDENCIAS QUE AFECTAN AL RENDIMIENTO DE LA INSTANCIA-----</b>	<b>8</b>
<b>ANÁLISIS DEL INFORME GENERADO POR STATSPACK-----</b>	<b>10</b>
<b>HERRAMIENTAS EN ORACLE ENTERPRISE MANAGER-----</b>	<b>19</b>

## VISTAS DINÁMICAS DE RENDIMIENTO

Las vistas dinámicas más importantes que nos dan información sobre el rendimiento de la instancia son:

### V\$SYSTEM\_EVENT.

Recoge información sobre las esperas totales por evento y el tiempo de estas esperas.

### V\$SYSSTAT

Recoge las estadísticas básicas acumuladas de la instancia, como el uso total de commits o de rollbacks, o los bloques totales de redo leídos.

### V\$SGAINFO.

Recoge información sobre el tamaño (en bytes) de todos los elementos componentes de la SGA (Shared pool, Large pool, etc). Además nos dice cual de estos elementos son redimensionables. Un ejemplo de redimensionable sería el Shared Pool y un ejemplo de no redimensionable sería el tamaño máximo de la SGA.

### V\$SGASTAT.

Recoge la información detallada de los elementos que componen la SGA. Si V\$SGAINFO nos mostraba el tamaño total de cada uno de estos componentes, V\$SGASTAT nos muestra el tamaño de todos los elementos que componen cada uno de los componentes.

### V\$BUFFER\_POOL\_STATISTICS.

Recoge información sobre las estadísticas de la caché de datos, como el número de buffers escritos o el número de buffers escaneados.

### V\$LIBRARYCACHE.

Proporciona información sobre el rendimiento de la library cache (caché de secuencias SQL). Por ejemplo nos da información de cuantas veces se solicitaron las sentencias sql y cuantas fueron rechazadas.

## V\$FILESTAT

Contiene información acerca de las estadísticas de los ficheros de datos escritos y leídos, como el número de veces que es requerido el DBWR en ese fichero.

## V\$LATCH

Proporciona información sobre los latches. Los latches son un mecanismo que protege la estructura de datos de la SGA contra los accesos simultáneos. Limitan la cantidad de tiempo y espacio en los que un proceso puede mantener un recurso en un instante dado.

## V\$WAITSTAT

Muestra estadísticas relacionadas con la contención de bloques de la base de datos .

## V\$SQL

Recoge información sobre las sentencias SQL en ejecución, incluyendo el consumo de memoria.

## V\$PROCESS

Recoge información acerca de los procesos que se encuentran activos en ese momento, como el usuario que lo esta usando, con el programa que se esta utilizando, en que archivo, o el tamaño de la memoria de pga usado.

## V\$BGPROCESS

Recoge información sobre los procesos en segundo plano, como la descripción de estos y los errores encontrados en ellos.

## STATSPACK

### **¿Qué es Statspack?**

Es una herramienta que contiene Oracle, que recopila información de las vistas más importantes del rendimiento. Este además analiza dichas estadísticas y genera un reporte con el diagnóstico global de rendimiento. Además muestra la información recopilada en un formato legible para el administrador de la Base de Datos.

### **¿Como es el reporte que genera?**

El reporte que genera cuenta con una estructura definida:

En primer lugar nos muestra el diagnóstico en sí, en el cual se encuentran los indicadores de desempeño (medición de las principales variables), y a continuación, los eventos que inciden en el rendimiento.

En segundo lugar el reporte nos presenta todos los eventos de la base de datos.

Luego las sentencias SQL más consumidoras de recursos ordenadas de 4 formas:

- SQL ordenadas por llamadas.
- SQL ordenadas por lecturas.
- SQL ordenadas por ejecuciones.
- SQL ordenadas por llamadas analizadas.

A continuación, un listado de todas las estadísticas de la base de datos.

Continúa con un listado de los tablespaces y los datafiles ordenados por la suma de lecturas y escrituras.

Las siguientes estadísticas son las de uso de :

- Buffer pool
- PGA
- Undo Segments
- Latches
- Shared pool

Por último nos enseña parámetros del init.ora

## ¿Cómo se instala?

Antes de instalar Statspack, es recomendable la creación de un tablespace de un mínimo de 500M, ya que al instalarlo nos pedirá un tablespace que designar para la utilización de Statspack.

Para instalar Statspack, tan solo tendremos que ejecutar en sql plus, con cualquier usuario que funcione como sysdba, el siguiente script:

```
@?/rdbms/admin/spcreate.sql
```

Este script creará también el usuario perfstat.

En la instalación se nos pedirá que elijamos el password del usuario perfstat, así como el tablespace por defecto, y el tablespace temporal.

## ¿Cómo se recopilan las estadísticas?

Para tomar una foto en un momento determinado se debe de ejecutar el siguiente comando como usuario perfstat en sql plus:

```
exec statspack.snap;
```

Sin embargo es recomendable automatizar la ejecución anterior, de forma que se fija la ejecución del statspack cada hora exacta (7:00,8:00,9:00,...). Para ello se debe ejecutar el siguiente script:

```
@?/rdbms/admin/spauto.sql
```

## ¿Cómo se genera un reporte?

Para generar un reporte hay que ejecutar el siguiente script como usuario perfstat:

```
@?/rdbms/admin/spreport.sql
```

Se pedirá que indiques la instantánea inicial y la final de la que quieres hacer el reporte. También la ubicación del fichero.

## 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 que se indique.

-spauto.sql: Permite automatizar la recolección de estadísticas.

### Mantenimiento (como usuario perfstat):

sppurge.sql -> Permite borrar un rango de snapshots

sptrunc.sql -> Vacía todas las tablas, borrando todos los snapshots

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

## INCIDENCIAS MÁS FRECUENTES QUE AFECTAN AL RENDIMIENTO DE LA INSTANCIA.

### **Contención en los latches de library cache**

La razón por la que ocurre esta incidencia, es por la acumulación de muchas peticiones a un mismo contenido de la caché de datos. Muchas de ellas se quedarán en espera. Para analizar si tenemos este problema podemos mirar la vista dinámica V\$LATCH. Existen varias latches que afectan a la library cache (library cache, library cache pin allocation, library cache lock,...) Existirán problemas de rendimiento si existen fallos (misses) con esperas(sleeps). Para evitar este problema habría que ampliar el tamaño de la Shared Pool.

### **Contención en los latch de shared pool**

La razón por la que ocurre esta incidencia, es por la acumulación de muchas peticiones a un mismo contenido de la shared pool. Muchas de ellas se quedarán en espera. Para analizar si tenemos este problema podemos mirar la vista dinámica V\$LATCH. El latch encargado de la shared pool se llama 'shared pool' Existirán problemas de rendimiento si existen fallos (misses) con esperas(sleeps) en este latch. Para evitar este problema habría que ampliar el tamaño de la Shared Pool.

### **Altos tiempos de CPU para compilar.**

Las estadísticas de esta incidencia se miraría en la fila “parse time cpu” de la vista V\$\$SYSSTAT

### **Muchas recompilaciones (reloads) en la caché de datos**

Las recompilaciones (reloads) son cada petición de metadatos (PIN) que no se encuentran en memoria por que los ha sacado el algoritmo LRU (ejecuciones que requieren recompilar sentencia). Se miran en la vista \$LIBRARYCACHE y para el correcto funcionamiento de la instancia debe ser prácticamente 0 en los namespaces sql/area, table/procedure,body y trigger. Si este valor no se acerca a 0 deberemos subir el tamaño de la shared pool.

### **Muchas llamadas de compilación.**

Las estadísticas de este incidencia se miraría en la fila “parse count(total)”, “parse count(hard)”, parse count(failures) de la vista V\$\$SYSSTAT. Si existen muchas sentencias analizadas con fracaso(“parse count(failures)”) con respecto al total (“parse count(total)”) el rendimiento de la base de datos no está siendo el correcto.



## **Contención en el latch 'cache buffers lru chain'.**

Como las anteriores contenciones de latches, tiene que ver con la acumulación de peticiones. Se puede comprobar en la vista V\$LATCH. Se dará si existen fallos (misses) con esperas (sleeps). La solución es evitar las lecturas innecesarias o los índices poco selectivos.

## **Mucho tiempo empleado en la espera "write complete waits" y mucho tiempo empleado en la espera "free buffer waits" .**

Podemos ver si hay esperas para estas dos incidencias en V\$BUFFER\_POOL\_STATISTICS. Pueden evitarse evitando lecturas innecesarias o los índices poco selectivos

## **Contención en latches de redo.**

Los latches de redo en los que tendremos que comprobar que hay contención, es decir, si existen fallos (misses) con esperas (sleeps), son 'redo copy' y 'redo allocation'. Una posible solución es la subida de la caché de Redo (log\_buffer).

## **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 número de esperas al escribir el redo a disco, por que se ha llenado el fichero redolog.

## **Contención en los segmentos de rollback**

Habrà contención en los segmentos de rollback, si el count de 'undo header' de la columna class de la vista V\$WAITSTAT, es mayor que 0. Para solucionarlo tendremos que crear un mayor número de segmentos de rollback.

## **Cuellos de botellas**

Los cuellos de botellas aparecen cuando hay una demanda excesiva simultánea sobre un recurso.

## ANÁLISIS DEL INFORME GENERADO POR STATSPACK

Anteriormente ya explique la estructura de Statspack, ahora veremos la información mas relevante recogida por este informe.

En primer lugar, Statspack nos da la información sobre la eficiencia de la actividad de la instancia. Dentro de este apartado también aparecen las estadísticas de la Shared Pool.

```
Instance Efficiency Indicators
~~~~~
      Buffer Nowait %: 100.00          Redo NoWait %: 100.00
      Buffer Hit %: 99.95           Optimal W/A Exec %: 100.00
      Library Hit %: 94.40          Soft Parse %: 89.21
      Execute to Parse %: 70.93      Latch Hit %: 100.00
Parse CPU to Parse Elapsed %: 22.58  % Non-Parse CPU: 95.07

Shared Pool Statistics          Begin   End
-----
      Memory Usage %: 89.20      89.11
      % SQL with executions>1: 60.35 63.18
      % Memory for SQL w/exec>1: 76.55 79.37
```

-Buffer Nowait: % de veces que se acceden a los buffers de datos sin tiempos de espera (Buffer Nowait).

-Buffer Hit: % de veces que los bloques de buffer se encontraban en la memoria, sin tener que ejecutar una operación de lectura.

-Library Hit: % de ocasiones en el que las sentencias sql se encontraban en la shared pool.

-Execute to Parse: % de utilización de sentencias sql ya analizadas.

-Parse CPU to Parse Elaps: Proporción de CPU dedicado a analizar las sentencias sql.

-Redo Nowait: % de tamaño suficiente en los buffers de redo.

-Soft Parse: % de sesiones en las que se utilizó sentencias sql que ya estaban en la shared pool.

-Latch Hit: Frecuencias de uso de latches sin esperas.

-Non-Parse CPU: % de cantidad de recursos de la CPU destinados a la ejecución de sql.

Las estadísticas de la Shared Pool, nos indican la memoria usada, y porcentaje de sentencias sql ejecutadas más de una vez.

El informe de Statspack trae la información de los 5 eventos que tienen un mayor peligro de constituir un posible cuello de botella en la base de datos, ya que son los que tienen un mayor número de esperas.

```

Top 5 Timed Events
~~~~~
Event                               Waits      Time (s)    Avg %Total
                                wait      (s)         wait   Call
                                (ms)     Time       (ms)   Time
-----
CPU time                             1           1           72.7
control file sequential read         470          0            1      17.5
control file parallel write          95           0            1       4.0
db file sequential read                5           0            9       2.2
os thread startup                      3           0           13       1.9

```

También podemos ver las estadísticas de memoria usada en la base de datos.

```

Memory Statistics
~~~~~
                Begin      End
-----
Host Mem (MB):    1,527.5    1,527.5
SGA use (MB):     407.1      407.1
PGA use (MB):     55.6       52.6
% Host Mem used for SGA+PGA:    30.3      30.1
-----

```

En la siguiente estadística también podremos ver en que gasta el tiempo la base de datos.

```

Time Model System Stats  DB/Inst: ORCL/orcl  Snaps: 1-2
-> Ordered by % of DB time desc, Statistic name

```

```

Statistic                               Time (s)  % DB time
-----
sql execute elapsed time                 1.0       88.6
DB CPU                                   0.7       58.4
parse time elapsed                       0.5       45.5
hard parse elapsed time                  0.5       42.8
PL/SQL compilation elapsed time          0.3       22.6
PL/SQL execution elapsed time            0.2       19.2
failed parse elapsed time                 0.2       17.9
connection management call elapsed       0.0        .7
repeated bind elapsed time                0.0        .2
DB time                                  1.1
background elapsed time                  0.6
-----

```

En este caso vemos que lo que más se hace en la base de datos (en lo que más tiempo emplea) es la ejecución de las sentencias sql).

Podemos ver los eventos de procesos, tanto los que están en primer plano, como los que están en segundo plano, para los que ha habido mas esperas.

```
Background Wait Events DB/Inst: ORCL/orcl Snaps: 1-2
-> Only events with Total Wait Time (s) >= .001 are shown
-> ordered by Total Wait Time desc, Waits desc (idle events last)
```

Event	Waits	%Tim out	Total Wait Time (s)	Avg wait (ms)	Waits /txn	%Total Call Time
control file sequential read	162	0	0	2	2.6	13.9
control file parallel write	95	0	0	1	1.5	4.0
os thread startup	3	0	0	13	0.0	1.9
.						
.						
.						

```
Foreground Wait Events DB/Inst: ORCL/orcl Snaps: 1-2
-> Only events with Total Wait Time (s) >= .001 are shown
-> ordered by Total Wait Time desc, Waits desc (idle events last)
```

Event	Waits	%Tim out	Total Wait Time (s)	Avg wait (ms)	Waits /txn	%Total Call Time
control file sequential read	308	0	0	0	4.9	3.6
db file sequential read	5	0	0	9	0.1	2.2
log file sync	9	0	0	0	0.1	.2

Estudiamos las sentencias sql en el informe. Podremos ver las sentencias que consumen más cpu.

```
SQL ordered by CPU DB/Inst: ORCL/orcl Snaps: 1-2
-> Total DB CPU (s): 1
-> Captured SQL accounts for 73.8% of Total DB CPU
-> SQL reported below exceeded 1.0% of Total DB CPU
```

CPU Time (s)	Executions	CPU per Exec (s)	%Total	Elapsd Time (s)	Buffer Gets	Old Hash Value
0.27	103	0.00	40.5	0.76	2,955	1206412919

```
Module: OEM.SystemPool
BEGIN EMD_NOTIFICATION.QUEUE_READY(:1, :2, :3); END;
```

Sentencias que tienen un mayor tiempo de ejecución

```
SQL ordered by Elapsed time for DB: ORCL Instance: orcl Snaps: 1 -2
-> Total DB Time (s): 1
-> Captured SQL accounts for 100.3% of Total DB Time
-> SQL reported below exceeded 1.0% of Total DB Time
```

Elapsed Time (s)	Executions	Elap per Exec (s)	%Total	CPU Time (s)	Physical Reads	Old Hash Value
0.76	103	0.01	67.5	0.27	0	1206412919

```
Module: OEM.SystemPool
BEGIN EMD_NOTIFICATION.QUEUE_READY(:1, :2, :3); END;
```

### Sentencias con un mayor número de lecturas lógicas.

```
SQL ordered by Gets DB/Inst: ORCL/orcl Snaps: 1-2
-> End Buffer Gets Threshold: 10000 Total Buffer Gets: 9,458
-> Captured SQL accounts for 24.1% of Total Buffer Gets
-> SQL reported below exceeded 1.0% of Total Buffer Gets
```

Buffer Gets	Executions	Gets per Exec	%Total	CPU Time (s)	Elapsd Time (s)	Old Hash Value
3,469	3	1,156.3	36.7	0.17	0.18	2689373535

```
DECLARE job BINARY_INTEGER := :job; next_date DATE := :mydate;
broken BOOLEAN := FALSE; BEGIN EMD_MAINTENANCE.EXECUTE_EM_DBMS_J
OB_PROCS(); :mydate := next_date; IF broken THEN :b := 1; ELSE :
b := 0; END IF; END;
```

### Sentencias con un mayor número de lecturas de disco físicas.

```
SQL ordered by Reads DB/Inst: ORCL/orcl Snaps: 1-2
-> End Disk Reads Threshold: 1000 Total Disk Reads: 5
-> Captured SQL accounts for 80.0% of Total Disk Reads
-> SQL reported below exceeded 1.0% of Total Disk Reads
```

Physical Rds	Executions	Rds per Exec	%Total	CPU Time (s)	Elapsd Time (s)	Hash Value
1	2	0.5	20.0	0.00	0.01	336764478

```
select /*+ index(idl_char$ i_idl_char1) */ piece#,length,piece
from idl_char$ where obj#=:1 and part=:2 and version=:3 order by
piece#
```

### Sentencias con un mayor número de ejecuciones.

```
SQL ordered by Executions DB/Inst: ORCL/orcl Snaps: 1-2
-> End Executions Threshold: 100 Total Executions: 2,391
-> Captured SQL accounts for 59.9% of Total Executions
-> SQL reported below exceeded 1.0% of Total Executions
```

Executions	Rows Processed	Rows per Exec	CPU per Exec (s)	Elap per Exec (s)	Old Hash Value
198	198	1.0	0.00	0.00	635752601

```
Module: OEM.SystemPool
SELECT MGMT_TASK_SEQUENCE.NEXTVAL FROM DUAL
```

### Sentencias que han sido mas veces analizadas.

```
SQL ordered by Parse Calls DB/Inst: ORCL/orcl Snaps: 1-2
-> End Parse Calls Threshold: 1000 Total Parse Calls: 695
-> Captured SQL accounts for 51.9% of Total Parse Calls
-> SQL reported below exceeded 1.0% of Total Parse Calls
```

Parse Calls	Executions	% Total Parses	Old Hash Value
91	103	13.09	1206412919

```
Module: OEM.SystemPool
BEGIN EMD_NOTIFICATION.QUEUE_READY(:1, :2, :3); END;
```

Statspack también nos da información de como ha actuado y operado la instancia durante el periodo que va entre fotografía y fotografía.

Instance Activity Stats DB/Inst: ORCL/orcl Snaps: 1-2

Statistic	Total	per Second	per Trans
DB time	52,303	275.3	830.2
DBWR checkpoint buffers written	155	0.8	2.5
DBWR checkpoints	0	0.0	0.0
DBWR transaction table writes	6	0.0	0.1
DBWR undo block writes	101	0.5	1.6
deferred (CURRENT) block cleanout			

Otro de los apartados importante son los de las estadísticas de entrada/salida de las tablespaces y de los ficheros de datos. Este punto también incluye las estadísticas del buffer pool.

Tablespace IO Stats DB/Inst: ORCL/orcl Snaps: 1-2

->ordered by IOs (Reads + Writes) desc

Tablespace

Tablespace	Av Reads	Av Reads/s	Av Rd (ms)	Av Blks/Rd	Writes	Av Writes/s	Buffer Waits	Av Buf Wt (ms)
SYSAUX	0	0	0.0		34	0	0	0.0
UNDOTBS1	0	0	0.0		15	0	0	0.0
SYSTEM	5	0	10.0	1.0	4	0	0	0.0

File IO Stats DB/Inst: ORCL/orcl Snaps: 1-2

->Mx Rd Bkt: Max bucket time for single block read

->ordered by Tablespace, File

Tablespace

Filename

Tablespace	Av Reads	Av Reads/s	Av Rd (ms)	Mx Rd Bkt	Av Blks/Rd	Writes	Av Writes/s	Buffer Waits	Av BufWt (ms)
SYSAUX	0	0				34	0	0	
SYSTEM	5	0	10.0	16	1.0	4	0	0	
UNDOTBS1	0	0				15	0	0	

```

Buffer Pool Statistics DB/Inst: ORCL/orcl Snaps: 1-2
-> Standard block size Pools D: default, K: keep, R: recycle
-> Default Pools for other block sizes: 2k, 4k, 8k, 16k, 32k
-> Buffers: the number of buffers. Units of K, M, G are divided by 1000

```

P	Pool	Buffer	Physical	Physical	Free	Writ	Buffer
	Buffers	Hit%	Gets	Reads	Buffer	Comp	Busy
					Waits	Wait	Waits
D	13K	100	9,605	5	155	0	0

También reporta información sobre la memoria de los procesos.

```

Top Process Memory (by component) DB/Inst: ORCL/orcl Snaps: 1-2
-> ordered by Begin/End snapshot, Alloc (MB) desc

```

PId	Category	Alloc (MB)	Used (MB)	Freeabl (MB)	Max Alloc (MB)	Hist Alloc (MB)	Max (MB)
B	10 LGWR	9.2	4.4	.0	9.2	11.4	11.4
	Other	9.2			9.2	11.4	
	PL/SQL	.0	.0		.0	.0	
	32 SHAD	3.8	2.8	.0	3.8	7.5	7.5
	Other	3.6			3.6	3.6	
	SQL	.1	.1		.1	4.1	4.1
	PL/SQL	.0	.0		.0	.1	
	34 SHAD	2.9	2.3	.0	2.9	2.9	2.9
	Other	2.7			2.7	2.7	
	PL/SQL	.1	.0		.1	.1	
	SQL	.1	.0		.1	.3	
	25 SHAD	2.9	2.4	.0	2.9	34.2	34.2
	Other	2.7			2.7	24.9	24.9
	SQL	.1	.0		.1	9.3	9.3
	PL/SQL	.1	.0		.1	.1	
	9 DBW0	2.7	2.2	.0	2.7	5.3	5.3
	Other	2.7					

Por supuesto también nos da información sobre los latches. Recordamos que habrá contención de latches cuando haya fallos con esperas (Slps/Miss).

```

Latch Activity DB/Inst: ORCL/orcl Snaps: 1-2
->"Get Requests", "Pct Get Miss" and "Avg Slps/Miss" are statistics for
willing-to-wait latch get requests
->"NoWait Requests", "Pct NoWait Miss" are for no-wait latch get requests
->"Pct Misses" for both should be very close to 0.0

```

Latch	Get Requests	Pct Get Miss	Avg Slps /Miss	Wait Time (s)	NoWait Requests	Pct NoWait Miss
active checkpoint queue	114	0.0		0	0	
active service list	417	0.0		0	69	0.0
archive control	5	0.0		0	0	
ASM db client latch	126	0.0		0	0	
ASM map operation hash t	1	0.0		0	0	
buffer pool	1	0.0				

También nos ofrece detalles de como está funcionando la caché de datos.

```
Dictionary Cache Stats DB/Inst: ORCL/orcl Snaps: 1-2
->"Pct Misses" should be very low (< 2% in most cases)
->"Final Usage" is the number of cache entries being used in End Snapshot
```

Cache	Get Requests	Pct Miss	Scan Reqs	Pct Miss	Mod Reqs	Final Usage
dc_awr_control	3	0.0	0		0	1
dc_database_links	4	0.0	0		0	1
dc_global_oids	274	1.8	0		0	146
dc_histogram_defs	1,605	37.6	0		0	4,127
dc_object_grants	68	10.3	0		0	117
dc_objects	780	0.4	0		0	3,080
dc_profiles	6	0.0	0		0	2
dc_rollback_segments	43	0.0	0		0	22
dc_segments	229	0.0	0		0	941
dc_tablespaces	1,522	0.0	0		0	15
dc_users	2,357	0.0	0		0	273
global database name	131	0.0	0		0	1

```
Library Cache Activity DB/Inst: ORCL/orcl Snaps: 1-2
->"Pct Misses" should be very low
```

Namespace	Get Requests	Pct Miss	Pin Requests	Pct Miss	Reloads	Invalidations
SQL AREA	208	20.7	3,273	7.4	5	1
TABLE/PROCEDURE	1,036	0.3	2,258	4.3	44	0
BODY	84	0.0	445	0.0	0	0
TRIGGER	47	0.0	56	0.0	0	0

Con la siguiente tabla, podremos ver si es aconsejable ampliar el tamaño de la Shared Pool, ya que nos ofrece algunas estadísticas por cada tamaño de la Shared Pool.

```
Shared Pool Advisory DB/Inst: ORCL/orcl End Snap: 2
-> SP: Shared Pool Est LC: Estimated Library Cache Factr: Factor
-> Note there is often a 1:Many correlation between a single logical object
in the Library Cache, and the physical number of memory objects associated
with it. Therefore comparing the number of Lib Cache objects (e.g. in
v$librarycache), with the number of Lib Cache Memory Objects is invalid
```

Shared Pool Size (M)	SP Size (M)	Est LC Size (M)	Est LC Mem Obj	Est LC Time Saved (s)	Est LC Time Saved Factr	Est LC Load Time (s)	Est LC Load Time Factr	Est LC Mem Obj Hits
132	.8	8	1,236	6,651	1.0	280	1.9	56,155
152	.9	28	3,614	6,701	1.0	230	1.6	136,485
172	1.0	48	5,075	6,787	1.0	144	1.0	138,366
192	1.1	68	6,459	6,816	1.0	115	.8	138,926



212	1.2	86	8,368	6,826	1.0	105	.7	139,053
232	1.3	96	9,891	6,827	1.0	104	.7	139,064
252	1.5	96	9,891	6,827	1.0	104	.7	139,064
272	1.6	96	9,891	6,827	1.0	104	.7	139,064
292	1.7	96	9,891	6,827	1.0	104	.7	139,064
312	1.8	96	9,891	6,827	1.0	104	.7	139,064
332	1.9	96	9,891	6,827	1.0	104	.7	139,064
352	2.0	96	9,891	6,827	1.0	104	.7	139,064

-----

Java Pool Advisory DB/Inst: ORCL/orcl End Snap: 2

Java Pool Size (M)	JP Size Factr	Est LC Size (M)	Est LC Mem Obj	Est LC Time Saved (s)	Est LC Time Saved Factr	Est LC Load Time (s)	Est LC Load Time Factr	Est LC Mem Obj Hits
12	1.0	3	118	29	1.0	144	1.0	68
16	1.3	4	156	29	1.0	144	1.0	85
20	1.7	4	157	29	1.0	144	1.0	85
24	2.0	4	157	29	1.0	144	1.0	85

Proporciona información sobre el tamaño de la memoria de los componentes de la SGA.

SGA Memory Summary DB/Inst: ORCL/orcl Snaps: 1-2

SGA regions	Begin Size (Bytes)	End Size (Bytes) (if different)
Database Buffers	117,440,512	
Fixed Size	1,333,648	
Redo Buffers	6,086,656	
Variable Size	301,991,536	
sum	426,852,352	

Por último da a conocer los parámetros de init.ora en el momento en el que se hizo la fotografía.

init.ora Parameters DB/Inst: ORCL/orcl Snaps: 1-2

Parameter Name	Begin value	End value (if different)
audit_file_dest	C:\APP\USUARIO\ADMIN\ORCL\ADUMP	
audit_trail	DB	
compatible	11.1.0.0.0	
control_files	C:\APP\USUARIO\ORADATA\ORCL\CONTR OL01.CTL, C:\APP\USUARIO\ORADATA\ ORCL\CONTROL02.CTL, C:\APP\USUARI O\ORADATA\ORCL\CONTROL03.CTL	
db_block_size	8192	
db_domain		
db_name	orcl	
db_recovery_file_dest	C:\app\Usuario\flash_recovery_are a	
db_recovery_file_dest_size	2147483648	
diagnostic_dest	C:\APP\USUARIO	
dispatchers	(PROTOCOL=TCP) (SERVICE=orclXDB)	
job_queue_processes	1000	
memory_target	427819008	
open_cursors	300	

# HERRAMIENTAS MÁS INTERESANTES EN ORACLE ENTERPRISE MANAGER

Las herramientas más interesantes que presenta Oracle Enterprise Manager con respecto al rendimiento de la base de datos son:

## **ADDM (Monitor de Diagnóstico de la Base de Datos Automático)**

El ADDM realiza un análisis del sistema, identifica los posibles problemas y sus causas potenciales, y por último plantea recomendaciones para solucionarlos.

La información que analiza el ADDM es:

- Cuellos de botella en la CPU
- Gestión ineficiente de conexiones
- Bloqueos
- Operaciones de entrada/salida
- Tamaño de las estructuras de memoria
- Carga de sentencias sql.
- Tiempo de ejecución de procedimientos PL/SQL y Java

Es muy fácil de generar. Tan solo tendremos que seleccionar el botón “ejecutar ADDM ahora” de la pestaña “Rendimiento”.



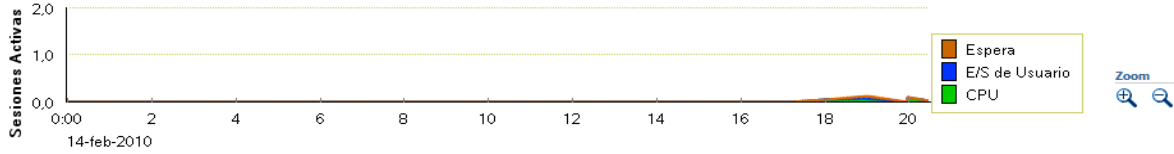
Al realizar el análisis nos da un resultado.

### Actividad de la Base de Datos

Ejecutar ADDM

Historial de Conclusiones

El icono seleccionado situado debajo del gráfico identifica el período de análisis de ADDM. Haga clic en un icono diferente para seleccionar otro período de análisis.



CONSEJO Para obtener una explicación de los iconos y símbolos utilizados en esta página, consulte [Clave de Icono](#)

### Análisis de Rendimiento de ADDM

Nombre de la Tarea **ADDM:1232777566\_1\_107**

Filtros

Ver Instantáneas

Ver Informe

Propietario de la Tarea **SYS**

Media de Sesiones Activas **0**

Hora de Inicio de Período **14-feb-2010 20H28' CET**

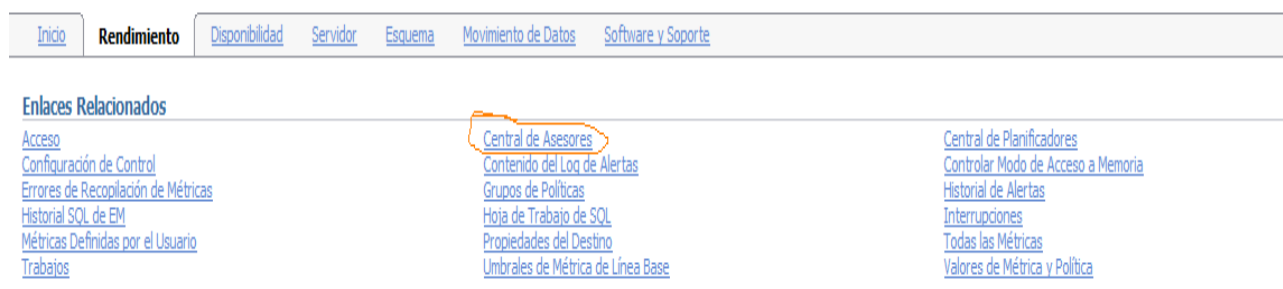
Duración del Período (minutos) **13,2**

Impacto (%)	Conclusión	Incidencias (Últimas 24 Horas)
ADDM no ha encontrado ningún problema durante este período		

En este caso no ha encontrado ningún problema.

## La central de asesores

Para acceder a ese apartado tendremos que irnos a la parte inferior de la pestaña rendimiento.



[Base de Datos](#) | [Configurar](#) | [Preferencias](#) | [Ayuda](#) | [Desconexión](#)

copyright © 1996, 2007, Oracle. Todos los Derechos Reservados.

Oracle, JD Edwards, PeopleSoft y Retek son marcas comerciales registradas de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

www.oracle.com

Y la pestaña que aparece nos ofrece las siguientes posibilidades:

**Asesor de memoria:** Analiza el tamaño de la SGA y de la PGA, optimizando el uso de la memoria global de la instancia. Si tienes activada la opción gestión automática de la memoria, la base de datos definirá automáticamente la distribución óptima de memoria. El botón consejo nos da una gráfica con el porcentaje de mejora en función de los distintos tamaños.

### **Asesores de sql:**

Se divide en tres partes :

-**Asesor de acceso sql** : Analiza las consultas realizadas y puede indicar si es conveniente crear índices o vistas materializadas para mejorar los tiempos de respuesta.

-**Asesor de ajustes sql**: Analiza las sentencias SQL y ofrece optimizaciones sobre las mismas.

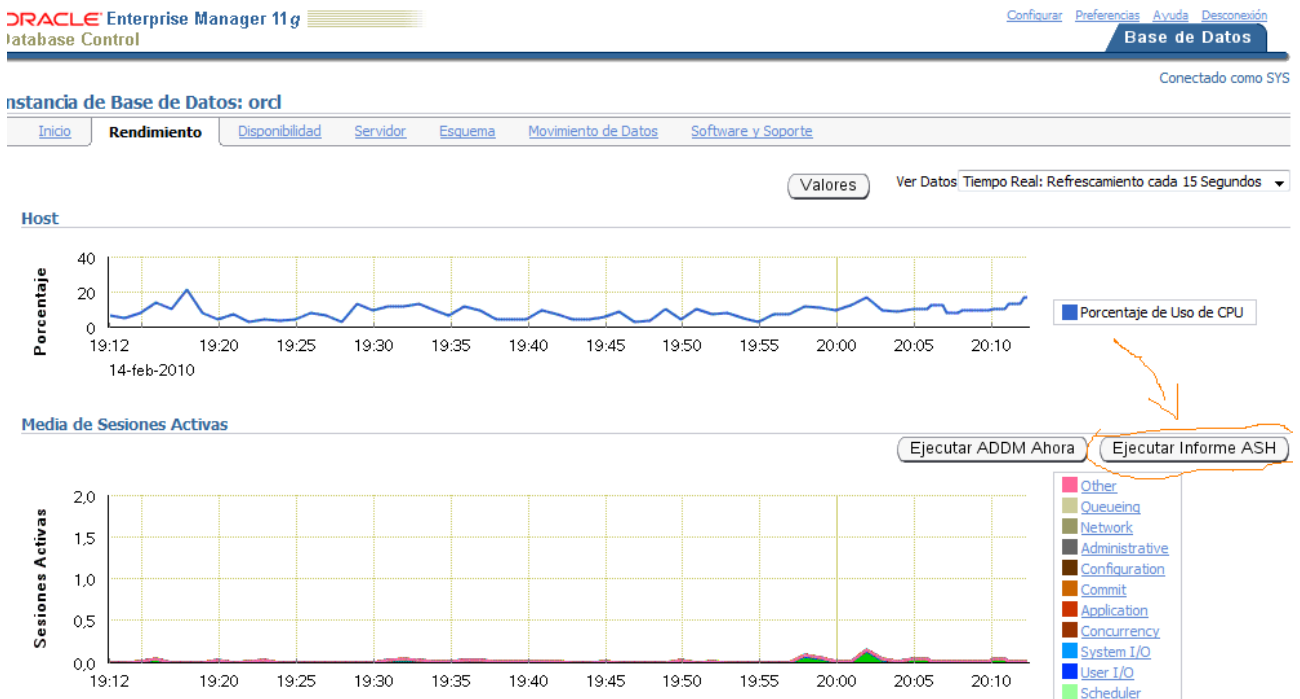
-**Resumen de resultados de ajustes SQL automáticos (SQL Tuning Advisory)**: Permite activar los ajustes automáticos de SQL. Con esta opción activada la base de datos buscará formas para mejorar los planes de ejecución de sentencias SQL de mucha carga. Ofrece estadísticas sobre esta tarea.

**Asesor de segmentos:** Proporciona información útil para el dimensionado de los segmentos (tablespaces, tablas,...) y para detectar aquellos que deben ser comprimidos.

**Gestión automática de deshacer (undo):** Proporciona información sobre el tamaño del tablespace de undo y del tiempo de retención de los datos en él.

# El informe ASH

El informe ASH es un informe muy parecido a Statspack, que se genera en Oracle Enterprise Manager solamente con seleccionar el botón “ejecutar informe ASH” en la pestaña de rendimiento.



El informe presenta lo mas utilizado, o lo que mas consume en cada una de las partes de la base de datos (eventos mas frecuentes, secuencias SQL ejecutadas con mayor frecuencia, etc). Es una herramienta fundamental para intentar detectar posibles casos de cuellos de botella.