# Real Time Statistics - Oracle Database 19c

Por Francisco Riccio

## Introducción

Una de las principales causas a los problemas de desempeño que tienen las bases de datos es contar con estadísticas desfasadas de sus tablas e índices, teniendo como consecuencia planes de ejecución ineficientes.

Oracle Database 10g ha realizado esfuerzos por mantener las estadísticas de base de datos de manera actualizada, para ello implementó tareas automatizadas que recolectan estadísticas en horas definidas, también recolecta estadísticas a una tabla particular cuando se crea algún índice sobre ella. La versión Oracle Database 12c añadió una extensión permitiendo recolectar estadísticas automáticamente cuando se crean tablas a partir de una consulta (create table as select).

Hoy la versión Oracle Database 19c permite recolecta estadísticas cuando una operación DML convencional o una importación ocurre, pero la información recolectada es una información "ligera" y se complementará cuando ocurra una recolección completa a través del paquete DBMS_STATS. Aún esta información "ligera" es válida para entregarle al optimizador mejor información sobre los datos de una tabla y optar por un mejor plan de ejecución dando como resultado un beneficio de desempeño.

## Implementación

La funcionalidad se encuentra habilitada por defecto y lo que se realizará a continuación es un análisis y validación.

Para realizar nuestra validación se cuenta con una tabla llamada TEST con 2 millones de filas y se procederá a insertar 500,000 filas de manera convencional.

```
SQL> set serveroutput on
declare
 v_inicio number;
 v_fin number;
begin
 v_inicio:=DBMS_UTILITY.get_time;
 for i in 2000001..2500000 loop
  insert into TEST values (i,'T'||to_char(i));
 end loop;
 commit;
 v_fin:=DBMS_UTILITY.get_time;
 DBMS_OUTPUT.put_line('Duracion: '||to_char((v_fin-v_inicio)/100)||' seg');
end;
/
Duracion: 23.66 seg

PL/SQL procedure successfully completed.
```

Se revisará algunas vistas del sistema que permitirán validar la información recolectada:

```
SQL> select NOTES,NUM_ROWS,STALE_STATS
     from DBA_TAB_STATISTICS
     where owner='FRICCIO' and table_name='TEST';

NOTES                          NUM_ROWS STALE_S
------------------------------ ---------- -------
                                 2000000 YES
STATS_ON_CONVENTIONAL_DML        2500000
```

**Se puede apreciar que la base de datos ha recolectado información sobre la tabla y tiene información que ahora el número de registros es de 2.5 millones y no 2 millones**.

También es importante recordar que en versiones anteriores la cantidad de operaciones DML que ocurría sobre una tabla podía ser obtenida por la vista: DBA_TAB_MODIFICATIONS.

```
SQL> select INSERTS,UPDATES,DELETES
     from dba_tab_modifications
     where TABLE_OWNER='FRICCIO' and TABLE_NAME='TEST';

   INSERTS     UPDATES     DELETES
---------- ---------- ----------
    500000          0          0
```

Lo más interesante es si corremos una consulta sobre la tabla, **podremos validar que el optimizar tomó ventaja de esta nueva funcionalidad para obtener un plan de ejecución más eficiente** cuando ocurra un hard parse.

```
SQL> select CAMPO2 from TEST where campo1=217002;

CAMPO2
------------------------------
T217002

SQL> select * from table(dbms_xplan.display_cursor(format=>'TYPICAL'));

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------------------------
SQL_ID  35q795p6a3dhm, child number 0
-------------------------------------
select CAMPO2 from TEST where campo1=217002

Plan hash value: 480040

-------------------------------------------------------------------------------------------------------------
| Id  | Operation                             | Name                 | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                      |                      |       |       |     4 (100)|          |
|   1 |  TABLE ACCESS BY INDEX ROWID BATCHED| TEST                 |     1 |    13 |     4   (0)| 00:00:01 |

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------------------------
|*  2 |   INDEX RANGE SCAN                    | SYS_AI_ayv3vppgrvmw8 |     1 |       |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("CAMPO1"=217002)

Note
-----
   - dynamic statistics used: statistics for conventional DML
```

Esta nueva funcionalidad no incluye un overhead sobre las operaciones DML, para validarlo volveremos a cargar 500 mil filas sin la opción habilitada, previa actualización de estadísticas completa.

```
SQL> execute dbms_stats.gather_table_stats('FRICCIO','TEST');

PL/SQL procedure successfully completed.

SQL> alter system set "_optimizer_gather_stats_on_conventional_dml"=false;

System altered.

SQL> alter system set "_optimizer_gather_stats_on_load"=false;

System altered.
```

Con la modificación de los parámetros ocultos se desactiva la funcionalidad pero esto no debe ser aplicado en algún entorno salvo sea una recomendación de Oracle Support. Se ha realizado estos cambios para propósito de este ejemplo.

Iniciamos el ingreso de 500 mil filas de manera convencional y tendremos el siguiente resultado:

```
SQL> set serveroutput on
declare
 v_inicio number;
 v_fin number;
begin
 v_inicio:=DBMS_UTILITY.get_time;
 for i in 2500001..3000000 loop
  insert into TEST values (i,'T'||to_char(i));
 end loop;
 commit;
 v_fin:=DBMS_UTILITY.get_time;
 DBMS_OUTPUT.put_line('Duracion: '||to_char((v_fin-v_inicio)/100)||' seg');
end;
/
Duracion: 22.06 seg

PL/SQL procedure successfully completed.

SQL> select NOTES,NUM_ROWS,STALE_STATS
     from DBA_TAB_STATISTICS
     where owner='FRICCIO' and table_name='TEST';

NOTES                          NUM_ROWS STALE_S
------------------------------ ---------- -------
                                2500001 YES
```

Se apreciar que las operaciones DML convencionales no se vieron afectadas en el tiempo de duración y que la base de datos ya no recolectó estadísticas sobre dichas operaciones en la tabla.

Asimismo al ejecutar consultas sobre la tabla, el optimizador no tomará alguna ventaja al no contar con la información

```
SQL> alter system flush shared_pool;

System altered.

SQL> set autotrace on
SQL> select CAMPO2 from TEST where campo1=217002;

CAMPO2
------------------------------
T217002


Execution Plan
----------------------------------------------------------
Plan hash value: 480040

-------------------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name                 | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |                      |     1 |    15 |     4   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID BATCHED| TEST                 |     1 |    15 |     4   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN                  | SYS_AI_ayv3vppgrvmw8 |     1 |       |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("CAMPO1"=217002)
```

## Conclusión

Esta nueva funcionalidad permite a los Administradores de Base de Datos mantener una base de datos más estable en temas de rendimiento sin alguna intervención manual, además de no generar un overhead significativo a las operaciones DML convencionales e importaciones.

**Publicado por:**

**Francisco Riccio,** actualmente se desempeña como Arquitecto de Soluciones en Oracle Perú y es instructor de cursos oficiales de certificación Oracle. Es un Oracle Certified Professional en productos de Oracle Application, Base de Datos, Cloud & Virtualización.