

# Result Cache en Oracle Database 11gR2

Por Francisco Riccio 

## Introducción

A partir de la versión Oracle Database 11g se cuenta con un espacio de memoria RAM llamado Result Cache que se encuentra alojado dentro del pool Shared Pool con la finalidad de almacenar resultados de queries y funciones PL/SQL que se ejecuten en el servidor de base de datos.

Este nuevo sub pool es muy útil para almacenar resultados de aquellos queries que analizan grandes cantidades de filas y retornan pocas filas; también para funciones PL/SQL que son frecuentemente utilizadas y cuyo resultado depende de información que no cambia mucho en el tiempo.

## Implementación

### Configuración del Result Cache

Por default el Result Cache almacena una cantidad de memoria RAM por default acorde a la siguiente regla:

- a) Si está configurado la base de datos con el parámetro MEMORY\_TARGET, el Result Cache será configurado con el 0.25% del valor del parámetro MEMORY\_TARGET automáticamente.
- b) Si está configurado la base de datos con el parámetro SGA\_TARGET, el Result Cache será configurado con el 0.5% del valor del parámetro SGA\_TARGET automáticamente.
- c) Si está configurado el parámetro SHARED\_POOL\_SIZE, el Result Cache será configurado con 1% del tamaño del valor del SHARED\_POOL\_SIZE automáticamente.

Si en caso queramos definir un tamaño al Result Cache de forma predeterminada debemos configurar un valor al parámetro **RESULT\_CACHE\_MAX\_SIZE**. Si el parámetro es configurado con el valor de 0 se deshabilita el sub pool Result Cache.

Asimismo la cantidad máxima de memoria que se puede albergar como resultado de un query o función PL/SQL se define por el parámetro: **RESULT\_CACHE\_MAX\_RESULT**, cuyo valor por default es de 5% respecto al tamaño del Result Cache.

### Utilización para setencias SQL

Si deseamos que el resultado de un query se almacene dentro del Result Cache debemos utilizar el hint result\_cache, ejemplo:

```
SQL> select /*+ result_cache */ count(*) from producto;
```

Al colocarle el hint de result\_cache hace que el resultado del query se guarde en el Result Cache y una vez almacenado recién se devuelva el resultado al usuario.

Si nosotros ejecutamos el query por segunda vez, veremos en el plan de ejecución del query, que el resultado es obtenido del Result Cache haciendo que no sea nuevamente calculado:

```
SQL> select /*+ result cache */ count(*) from producto;
```

```
COUNT (*)
-----
          3
```

#### Execution Plan

```
Plan hash value: 4118974845
```

Id	Operation	Name	Rows	Cost (%CPU)	Time
0	SELECT STATEMENT		1	2 (0)	00:00:01
1	RESULT CACHE	ds86zn9knw21ac9k2d9v6r5bfa			
2	SORT AGGREGATE		1		
3	TABLE ACCESS FULL	PRODUCTO	3	2 (0)	00:00:01

Toda operación DML (insert/delete/update) ó DDL (alter table) sobre tablas que influye sobre los resultados almacenados en el Result Cache ocasiona que dichos resultados sean invalidados. En este escenario, el query tendrá que volver a calcular el resultado porque su resultado en el Result Cache fue invalidado.

Asimismo el parámetro **RESULT\_CACHE\_REMOTE\_EXPIRATION** permite invalidar el resultado de un query que haga llamadas a objetos remotos. Por default tiene el valor de 0, es decir no se almacena resultados que se calculan en base a información de objetos remotos. El valor del parámetro es expresado en segundos.

Si nosotros queremos que el resultado de todo query o función PL/SQL se almacene en el Result Cache sin utilizar el hint result\_cache configuramos el parámetro **RESULT\_CACHE\_MODE** con el valor de FORCE; por default este parámetro tiene el valor de MANUAL. En caso lo configuremos con el valor de FORCE podemos utilizar el hint **no\_result\_cache** si queremos que no se almacene en el Result Cache.

**Nota 1:** Si deseamos que el Result Cache a pesar de estar configurado sea omitido podemos ejecutar el siguiente procedimiento: **DBMS\_RESULT\_CACHE.BYPASS(true)** y colocándole el valor de false volvemos a dejar el Result Cache en modo normal.

**Nota 2:** No podemos usar Result Cache en sentencias SQL si estas utilizan los siguientes objetos: Tablas temporales, tablas del diccionario de datos, funciones no determinísticas como currval, nextval, sysdate, etc.

## Utilización para funciones PL/SQL

Una función PL/SQL debe tener la cláusula RESULT\_CACHE si queremos que su resultado se almacene en el Result Cache, ejemplo:

```
SQL> create or replace function getTotalPU return number
result_cache
is
  v_pu number:=0;
begin
  select sum(pu) as total
  into v_pu from producto;
  return v_pu;
end;
/
```

Function created.

Nota: En la versión Oracle Database 11gR1 se requería la cláusula RESULT\_CACHE RELIES ON (<nombre\_tabla>), donde la cláusula RESULT\_CACHE indica a Oracle Database que invalide el resultado de la función almacenada en el Result Cache para cualquier operación DML o DDL que ocurra sobre la tabla que se liste en el cláusula. En **My Oracle Support (MOS) Nota: 1142314.1 (Processes Terminate With ORA-7445 [\_\_intel\_new\_memcpy()]) Errors After Upgrade to 11.2**, se especifica que la cláusula RELIES ON ya no debe ser especificada en versiones Oracle Database 11gR2 hacia adelante.

Las restricciones que tiene las funciones PL/SQL sobre Result Cache es que ellas no deben ser funciones pipelined, no pueden tener parámetros out/in out y si son parámetros in no pueden ser de tipo LOB, ref cursor, objetos y colecciones.

## Reportes & Validación

- a) Si deseamos obtener un reporte del espacio utilizado en el Result Cache podemos utilizar la función: **DBMS\_RESULT\_CACHE.MEMORY\_REPORT**.

```
SQL> set feedback 1
SQL> set serveroutput on
SQL> execute dbms_result_cache.memory_report;
Result Cache Memory Report
[Parameters]
Block Size           = 1K bytes
Maximum Cache Size  = 10496K bytes (10496 blocks)
Maximum Result Size = 524K bytes (524 blocks)
[Memory]
Total Memory = 158272 bytes [0.003% of the Shared Pool]
... Fixed Memory = 12768 bytes [0.000% of the Shared Pool]
... Dynamic Memory = 145504 bytes [0.003% of the Shared Pool]
..... Overhead = 112736 bytes
..... Cache Memory = 32K bytes (32 blocks)
..... Unused Memory = 30 blocks
..... Used Memory = 2 blocks
..... Dependencies = 1 blocks (1 count)
..... Results = 1 blocks
..... SQL = 1 blocks (1 count)

PL/SQL procedure successfully completed.
```

- b) La vista **V\$RESULT\_CACHE\_STATISTICS** permite visualizar cuantos resultados fueron almacenados en el Result Cache y asimismo cuantas veces un query obtuvo sus resultados del cache.

```
SQL> col name format a55
SQL> select * from v$result_cache_statistics;
```

ID NAME	VALUE
1 Block Size (Bytes)	1024
2 Block Count Maximum	10496
3 Block Count Current	32
4 Result Size Maximum (Blocks)	524
5 Create Count Success	1
6 Create Count Failure	0
7 Find Count	2
8 Invalidation Count	0
9 Delete Count Invalid	0
10 Delete Count Valid	0
11 Hash Chain Length	1
ID NAME	VALUE
12 Find Copy Count	2
13 Global Hit Count	0
14 Global Miss Count	0

14 rows selected.

- c) Si deseamos ver los queries que tienen resultados guardados en el Result Cache y además de visualizar con que tablas tiene dependencias, estos resultados consultamos la vista: **V\$RESULT\_CACHE\_OBJECTS**.

```
SQL> select type, namespace, status, scan_count, name
      from v$result_cache_objects;
```

TYPE	NAMES	STATUS	SCAN_COUNT	NAME
Dependency		Published	0	FRICCIO.PRODUCTO
Result	SQL	Published	2	select /*+ result_cache */ count(*) from producto

La vista mediante el campo status nos puede entregar si un resultado está publicado, expirado (expired) ó invalidado (invalid).

- d) Para obtener un reporte de cada resultado almacenado en el Result Cache con que objetos de la base de datos tiene dependencia consultamos la vista: **V\$RESULT\_CACHE\_DEPENDENCY**.

```
SQL> select RESULT_ID,OBJECT_NO from v$result_cache_dependency;
```

RESULT_ID	OBJECT_NO
1	16375996

- e) Si deseamos limpiar la cache del Result Cache ejecutamos el siguiente procedimiento:

```
SQL> execute DBMS_RESULT_CACHE.FLUSH;
```

### Result Cache Client

Oracle Database 11g también nos da la posibilidad de utilizar la memoria del lado del cliente para almacenar los resultados de las sentencias SQL que ejecute la aplicación cliente.

Al habilitar el Result Cache Client evitamos viajes innecesarios al servidor de Base de Datos además de ahorrar espacio en él ya que el resultado se almacena en la capa del cliente y no en el caché del servidor. El resultado almacenado en el Result Cache Client es independiente al Result Cache Server, es decir el Result Cache Client puede ser configurado sin que esté configurado el Result Cache en el servidor.

La base de datos es responsable de invalidar un resultado almacenado en el Result Cache Client cuando una operación DML o DDL ha ocurrido sobre un objeto que influye en el resultado de una operación almacenada.

Result Cache Client está habilitado para aplicaciones Oracle OCI, JDBC OCI, ODP.NET, Pro\*C, Pro\*Cobol y ODBC y no está habilitado por default. El cliente Oracle debe tener una versión mínima a 11gR1 y no está soportado para conexiones beaquerth.

Nota: Subqueries, vistas, objetos remotos, flashback queries, queries que incluyen funciones PL/SQL y queries que referencian políticas VPD en una tabla no son enviados al cache.

Parámetros relacionados a su configuración:

- a) **CLIENT\_RESULT\_CACHE\_SIZE**, determina el máximo tamaño del resultado de un query que un cliente puede almacenar en su cache.

Si su valor es 0 entonces el Result Cache Client está deshabilitado.

- b) **CLIENT\_RESULT\_CACHE\_LAG**, indica el número de segundos que el cliente Oracle validará contra el servidor la consistencia de la información en cache, no es obligatorio su configuración.

Para obtener las métricas del trabajo realizado por el Result Cache Client podemos consultar la vista: **CLIENT\_RESULT\_CACHE\_STATS\$**.

## Conclusión

Concluimos que Oracle Database 11g ha diseñado un nuevo sub pool llamado Result Cache que nos ayuda almacenar resultados de nuestros queries y funciones PL/SQL en ventaja de que si son nuevamente invocados dichas operaciones, el trabajo no tenga que ser nuevamente calculado, evitando un gasto innecesario de recursos en la base de datos y por el lado del usuario es una mejora considerable en el tiempo de respuesta.

Publicado por Ing. Francisco Riccio. Es un IT Specialist en IBM Perú e instructor de cursos oficiales de certificación Oracle. Está reconocido por Oracle como un Oracle ACE y certificado en productos de Oracle Application & Base de Datos.

e-mail: [francisco@friccio.com](mailto:francisco@friccio.com)

web: [www.friccio.com](http://www.friccio.com)