

Oracle12c result cache: the good, the bad and the ugly cases

Jérôme Witt
dbi services
Basel, Switzerland

Keywords

Oracle Database 12c, Performance, Tuning, Result Cache

Introduction

Caching is a mechanism well-known for relational databases. Memory is way faster than disks so the idea to keep data in memory is not new. The goal is to keep frequently used data in memory to save access time. But to build a result for a SQL query, there is not only the time to access data but some time is also consumed to work on that data for join operation or aggregation. The idea behind Result Cache is also to save the time and resource consumed to build the result set by putting directly the results in memory. So the engine can return directly a result without having to read data even in cache. This concept doesn't exist in all relational databases, SQL Server and PostgreSQL doesn't have a result cache mechanism built-in whereas MySQL supports this feature for a long time.

Result Cache overview

Result cache has been introduced in Oracle 11g in EE. Without going into the details of parsing and cursor creation in Oracle, we can summarize the execution of a SQL query in three big phases:

- 1- Get data from the storage to the cache (for data not already in cache)
- 2- Run the operation like joins, aggregations and sorts
- 3- Send back the results to the client

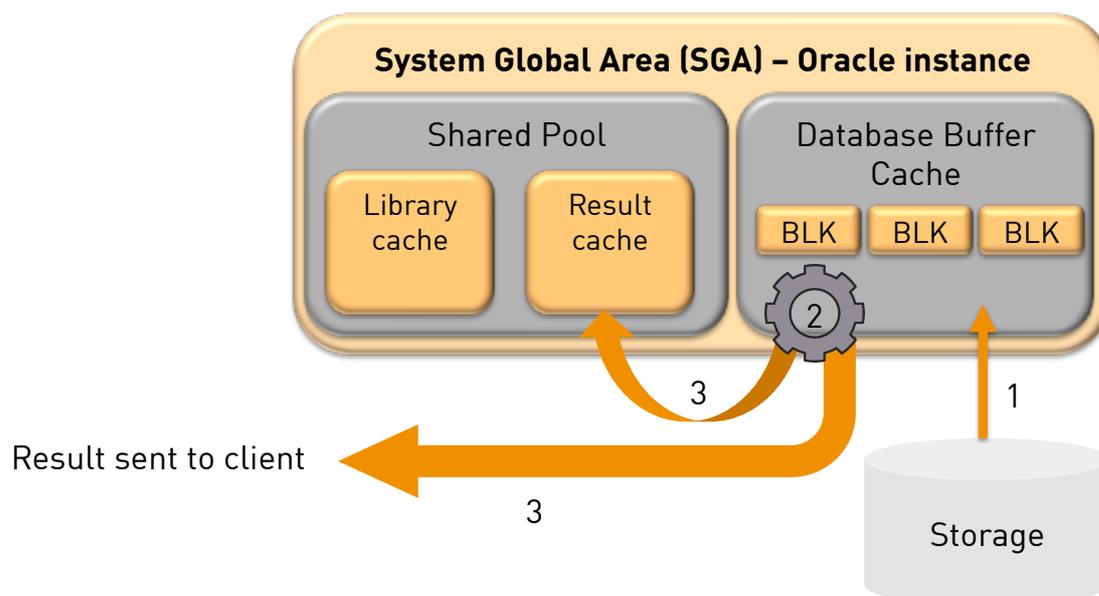


Abb. 1: Result Cache concept

When Result Cache is used, the result is not only sent to the client but it's also kept in a specific memory pool within the shared pool.

So if any client starts a similar query that needs the same result, Oracle sends directly the result from the Result Cache pool and skips the steps 1 and 2. The performance gain is obvious.

The result cache feature is controlled by several init. parameters to define the size of the memory pool allocated to keep the result sets, controls the way the feature is triggered (manually or automatically if set to FORCE) and the maximum size of a single result within the result cache in percentage (in that case 5% of 5920K).

NAME	TYPE	VALUE
-----	-----	-----
result_cache_max_result	integer	5
result_cache_max_size	big integer	5920K
result_cache_mode	string	MANUAL
result_cache_remote_expiration	integer	0

By default the result cache is enabled but not used automatically (result_cache_mode set to MANUAL). It means that any developer may use that feature.

They can set a flag at the table level, if all tables in the query have the flag set to force the query will go through the result cache.

```
alter table customers result_cache (mode force);
```

They can also use the RESULT_CACHE / NO_RESULT_CACHE hints to force a query to use (or not) the result cache

```
SELECT /*+ RESULT_CACHE | NO_RESULT_CACHE */ dummy from dual;
```

By default Oracle takes care of invalidation of result sets stored in the result cache if an underlying table is changed (update, insert or delete). The goal is that clients doesn't get stale data from the Result cache. The cache is automatically populated when the query is run and then invalidated when a DML is committed. The cache is available for all sessions connected to the database.

There are several ways to monitor and control the result cache usage :

- The package DBMS_RESULT_CACHE
- Dynamic performance views starting with v\$RESULT_CACHE

```

SQL> set serveroutput on
SQL> exec DBMS_RESULT_CACHE.Memory_Report();
Result Cache Memory Report
[Parameters]
Block Size           = 1K bytes
Maximum Cache Size  = 4M bytes (4K blocks)
Maximum Result Size = 204K bytes (204 blocks)
[Memory]
Total Memory = 5180 bytes [0.004% of the Shared Pool]
... Fixed Memory = 5180 bytes [0.004% of the Shared Pool]
... Dynamic Memory = 0 bytes [0.000% of the Shared Pool]
PL/SQL procedure successfully completed.

```

```

V$RESULT_CACHE_OBJECTS: Information about cached objects
V$RESULT_CACHE_DEPENDENCY: Dependency relationships
V$RESULT_CACHE_MEMORY: Memory usage statistics
V$RESULT_CACHE_STATISTICS: Result cache statistics

```

Deep dive and limits

Result cache can be used with views. The behavior will be like a materialized view, a query can join data from the regular buffer cache to a result coming from the result cache. Oracle will prevent view merging to take advantage of the result cache. The result cache query cannot be correlated to the outer query.

It allows building intermediate result on “static” tables to build some reports with pre-aggregated data for example.

```

SQL> WITH ch_cust AS ( select /*+RESULT_CACHE*/ ... )
SELECT ... FROM orders o, ch_cust v ...

```

The join is done against the result set coming from memory if the query has been already started.

Id	Operation	Name	Rows
0	SELECT STATEMENT		
1	HASH GROUP BY		8746
* 2	FILTER		
* 3	HASH JOIN		13633
4	VIEW		12506
5	RESULT CACHE	bqygnntud3aarcf35m0rc3dx59	
* 6	TABLE ACCESS FULL	CUSTOMERS	12506
* 7	TABLE ACCESS FULL	ORDERS	268K

The performance improvement is significant if the base tables are almost static so the same result sets can be re-used for several users and reports.

But serialization has to happen when the cache has to be updated. The goal is to prevent a session from accessing a result that is not null yet or that should be invalidated.

This serialization is done through latches and there is one specific latch to protect the result cache memory pool.

```
SQL> select * from v$latchname where name = 'Result Cache: RC Latch';
```

LATCH#	NAME	HASH	CON_ID
559	Result Cache: RC Latch	1054203712	0

Reading the result cache is done taking a shared lock against the latch but modify the result cache (for adding a new result set or invalidating an existing one) needs an exclusive lock to assure nobody can access it at the same time.

Things starts to become bad because all result sets are protected by a single latch. Adding a new result set into the cache prevents other sessions from reading other results sets. Using the result cache for queries on table frequently updated will introduce a contention on that single latch protecting the result cache.

The situation can be even worse if result cache starts to be used by almost all queries on some objects. Oracle 12c Adaptive Dynamic Statistics has a side effect because it use the RESULT_CACHE hint. So by default all result from ADS will be stored in the result cache putting pressure on the RC latch. In case of high concurrency on objects having a SQL Plan Directive enabling ADS the RC latch will be a big contention point.

Conclusion

Result cache is a step beyond materialized views, the result is not stored anymore and is accessed in memory. Result cache is an interesting feature to improve performance but like many performance features it shouldn't be enabled for all database activity.

Kontaktadresse:

Jérôme Witt
dbi services
Voltastrasse, 104
CH-4056 Basel

Telefon: +41 (0) 79 961 27 73
Fax: +41 (0) 32 422 96 15
E-Mail: jerome.witt@dbi-services.com
Internet: www.dbi-services.com