Oracle 12c Result Cache: the good, the bad and the hidden cases
About me (us)

Nicolas Jardot
Senior Consultant
nicolas.jardot@dbi-services.com
www.dbi-services.com

Jérôme Witt
Senior Consultant
Mobile +41 79 961 27 73
jerome.witt@dbi-services.com
www.dbi-services.com
Who we are
dbi services

Experts At Your Service
> 40 specialists in IT infrastructure
> Certified, experienced, passionate

Based In Switzerland
> 100% self-financed Swiss company
> Over CHF 6 mio. turnover

Leading In Infrastructure Services
> More than 100 customers in CH, D, & F
> Over 40 SLAs dbi FlexService contracted

dbi services is hiring in Basel & Zürich (career@dbi-services.com)
Agenda

1. Introduction
2. Result cache overview
3. Deep dive and limits
4. Conclusion
Introduction

- Why caching?
- Caching results in Oracle
- Equivalence in other RDBMS
Introduction

Why caching?

**Loading data is time expensive**
- Disks are way slower than memory access
- Same data may be accessed often
- Data cached to get it faster (i.e. buffer cache, in-memory)
- Data cached to avoid loading again same data

**Building result set is expensive too**
- Join data from different sources
- Data may need to be filtered/sorted or transformed (i.e. aggregate)
- Caching directly the result to avoid additional work on data

**Two main objectives for caching**
- Save time
- Save resource
Materialized views allow to store result of a query

- Data stored physically in a table
- Result set data can be cached like regular data
- Query rewrite allows to create materialized views that will be substituted from the original tables (Enterprise Edition)
- Save resources used to compute the result set

Cache problematic: When to refresh?

- Materialized views can be refreshed automatically or on demand
- Not a real cache, data is stored and changed on disk
- Share same cache as regular tables and may need to be read from disk
Introduction
Caching results in Oracle – 2

Result cache introduced in Oracle 11g Release 1
- Available only for Oracle Enterprise Edition (EE) without option

Store result sets instead of data blocks
- Result of SQL queries
- Values returned by PL/SQL function
- Automatic result set invalidation when data changes (11gR2)

Result sets stored only in memory
- Stored within the shared pool
- Shared across all sessions (public)
Introduction

Result cache equivalence in other RDBMS - 1

MySQL

MySQL 4.0 New Feature “result cache”
Stores result + SQL text
Automatic invalidation

PostgreSQL

Available as extension:
“PostgreSQL Query Cache”
Automatic invalidation

2003

2007

2011

Addendum:
2005 : Oracle acquires Innobase
2009 : Sun buys MySQL AB
2010 : Oracle acquires Sun (aka. MySQL)
Introduction
Result cache equivalence in other RDBMS - 2

Does Microsoft SQL Server caches query results?
> Of course, it caches pages containing data and query plan

MSSQL 2016 New Feature Query Store
"aka. Oracle SQL Plan management"
Result cache overview

- Concept
- Configure and monitor usage
- PL/SQL result cache
Result cache overview
Concept

What happens when a client runs a SQL query?

System Global Area (SGA) – Oracle instance

Shared Pool
- Library cache
- Result cache

Database Buffer Cache
- BLK
- BLK
- BLK

Storage

Result sent to client

Oracle 12c Result Cache
Result cache overview

Concept

What happen when a client run again the same SQL query?

![Oracle Result Cache Diagram]

- System Global Area (SGA) – Oracle instance
  - Shared Pool
    - Library cache
    - Result cache
  - Database Buffer Cache
    - BLK

Result sent to client

Oracle 12c Result Cache
Result cache overview
Configure and monitor usage – 1

Result cache is controlled by dynamic parameters at instance or session level

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_cache_max_result</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td>result_cache_max_size</td>
<td>big integer</td>
<td>5920K</td>
</tr>
<tr>
<td>result_cache_mode</td>
<td>string</td>
<td>MANUAL</td>
</tr>
<tr>
<td>result_cache_remote_expiration</td>
<td>integer</td>
<td>0</td>
</tr>
</tbody>
</table>

> Not active but enabled by default
> Can be controlled at statement level using hint

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

> Can be enabled for queries on specific tables

```sql
alter table customers result_cache (mode force);
```
Result cache overview
Configure and monitor usage – 2

Oracle maintains result sets dependencies
- Invalidate automatically results in case of changes (since Oracle 11gR2)

Each instance has its own Result cache in RAC environment
- However result sets may be transferred through cache fusion

Limitations
- Only stored in memory, not persistent like materialized view
- Need to run almost the same query to take advantage of cached results
- Doesn’t cache results for objects stored in SYS/SYSTEM
- No cache for statement using functions related to current context
“Similar” query can re-use the cached result
> Query text is case insensitive
> Oracle removes spaces and comments
> Oracle removes some SQL feature (e.g. FETCH FIRST N ROWS)

Oracle adds an operation and identify the result set with a generated key (CACHE_ID)

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RESULT CACHE</td>
<td>73ck5un44fx578yyd6amtq7yw5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SORT AGGREGATE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>INDEX FAST FULL SCAN</td>
<td>CUSTOMERS_PK</td>
<td>1000K</td>
</tr>
</tbody>
</table>
Monitor result cache objects and usage

- V$RESULT_CACHE_OBJECTS: Information about cached objects

<table>
<thead>
<tr>
<th>TYPE</th>
<th>STATUS</th>
<th>NAME</th>
<th>INVALID</th>
<th>ROW_COUNT</th>
<th>SCAN_COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>Published</td>
<td>SOE.CUSTOMERS</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Result</td>
<td>Published</td>
<td>select ...</td>
<td>0</td>
<td>12506</td>
<td>0</td>
</tr>
</tbody>
</table>

- V$RESULT_CACHE_DEPENDENCY: Dependency relationships
- V$RESULT_CACHE_MEMORY: Memory usage statistics
- V$RESULT_CACHE_STATISTICS: Result cache statistics

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>VALUE</th>
<th>CON_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Create Count Success</td>
<td>3059889</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Find Count</td>
<td>9181640</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>Invalidation Count</td>
<td>57290</td>
<td>0</td>
</tr>
</tbody>
</table>
Result cache overview
Configure and monitor usage – 5

Use DBMS_RESULT_CACHE package to manage from PL/SQL
> FLUSH: Clear the cache and release memory
> STATUS: Get result cache status
> BYPASS: Temporarily disable result cache usage
> INVALIDATE: Manually invalidate results
> MEMORY_REPORT: Get memory usage report

SQL> set serveroutput on
SQL> exec DBMS_RESULT_CACHE.Memory_Report();
Result Cache Memory Report
[Parameters]
Block Size          = 1K bytes
Maximum Cache Size  = 3872K bytes (3872 blocks)
Maximum Result Size = 193K bytes (193 blocks)
[Memory]
Total Memory = 169392 bytes [0.029% of the Shared Pool]
... Fixed Memory = 5440 bytes [0.001% of the Shared Pool]
... Dynamic Memory = 163952 bytes [0.028% of the Shared Pool]
PL/SQL procedure successfully completed.
Result cache overview

PL/SQL result cache - 1

Stores output of PL/SQL function in Result cache

- Function code not executed once cached
- Keeps track of input parameters and manage invalidation automatically
- Cache used to cache package global values for the whole instance instead of a session

CREATE OR REPLACE FUNCTION f_abcd
RETURN NUMBER
RESULT_CACHE
AS ...

Not the same as deterministic
- Returned value may depend on DML

Not the same as caching SQL
- Doesn’t cache the SQL starting the function but the function call itself
Result cache overview

PL/SQL result cache - 2

**Ugly case(s)**: Oracle 11gR1/11gR2 + 12cR1

**Bug 21484570 - HINT RESULT_CACHE CAUSES FUNCTION TO EXECUTE TWICE**
- S.R ongoing
- Non result-cached function

**Bug 21907155 : PL/SQL RC FN SHOWS ONE SESSION THE CONTENTS OF ANOTHER SESSION’S GTT**
- https://jonathanlewis.wordpress.com/2015/09/22/result-cache/

*Do not mixup PL/SQL result cache with database queries*
Deep dive and limits

- Inline views
- Latch and invalidation
- Adaptive Dynamic Statistics
Deep dive and limits
Inline views – 1

Result cache can be used to store results from views
> A regular view definition (using the hint)
> An inline view specified in FROM clause
> An inline view created using WITH

Can’t cache results for view using correlated column

Using result cache prevents Oracle from merging the view to the outer query
> Query rewrite partially disabled to maximize cached result usage
Deep dive and limits
Inline views – 2

Build intermediate result on most static table
- Join the view to the less static part of the SQL statement

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

Cached result is used to do the join

```
<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>HASH GROUP BY</td>
<td></td>
<td>8746</td>
</tr>
<tr>
<td>* 2</td>
<td>FILTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 3</td>
<td>HASH JOIN</td>
<td></td>
<td>13633</td>
</tr>
<tr>
<td>4</td>
<td>VIEW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RESULT CACHE</td>
<td>bqygntud3aarcf35m0rc3dx59</td>
<td>12506</td>
</tr>
<tr>
<td>* 6</td>
<td>TABLE ACCESS FULL</td>
<td>CUSTOMERS</td>
<td>12506</td>
</tr>
<tr>
<td>* 7</td>
<td>TABLE ACCESS FULL</td>
<td>ORDERS</td>
<td>268K</td>
</tr>
</tbody>
</table>
```
Deep dive and limits
Latch and invalidation – 1

Serialization is necessary to guaranty data consistency
  > Need to prevent users having stale results

Latch used to prevent distinct sessions to modify/access memory structures at the same time
  > Impact application scalability

Result cache is protected by latches
  > Only at CBD$ROOT level for multitenant environments

```sql
SQL> select * from v$latchname where name = 'Result Cache: RC Latch';
LATCH#   NAME                   HASH       CON_ID
------- ---------------------- ---------- ----------
  559 Result Cache: RC Latch  1054203712    0
```
Deep dive and limits
Latch and invalidation – 2

**Shared mode** used to access result sets (since 11gR2)
- Allow concurrent access to different result sets

**Exclusive mode** required to update the result cache
- Single latch used to serialize access to result cache
- No access possible when cache is invalidated or populated
- Invalidating one result set prevent access to all result sets

Result cache may be source of contention in several cases
- Generating lots of result sets (i.e. statement having different bind values)
- High invalidation rate (i.e. statement on frequently updated table)
Oracle 12c new feature
- Enabled if `optimizer_dynamic_sampling` is set to level 11
- New sampling level
- Backported to 11.2.0.4
- Can be triggered by SQL Plan Directives

Scan tables to obtain better estimations for current predicates
- If missing statistics
- When there is a complex expression
Deep dive and limits
Adaptive Dynamic Statistics – 2

Result cache used to store results from Adaptive Dynamic Statistics
> Avoid re-execution of sampling queries for different parse operation

Sampling queries use RESULT_CACHE hint
> Cached by default as RESULT_CACHE_MAX_SIZE > 0

Invalidation after time specified in 12.1.0.2
> Dependency tracking disabled (DEPEND_COUNT=0)

SQL> SELECT name, scan_count as “scnt”, depend_count
      FROM v$result_cache_objects WHERE status='Published'
      AND scan_count<>0
      ORDER BY scan_count DESC FETCH FIRST 1 ROWS ONLY;

NAME                                SCAN_COUNT
-----------------------------------------------
SELECT /* DS_SVC */ /*+ dynamic_sampling(0) no_sql_tune
no_monitoring optimize ... result_cache(snapshot=3600) 76
Ugly case: AWR latch statistics section show high contention on 'Result Cache: RC Latch' (Doc ID 2002089.1)

SQL Plan Directives meaning there is not enough statistics
> Object used by lots of OLTP queries
> Adaptive Dynamic Statistics started too often to get better cardinalities

Side effect: put pressure on result cache latch
> Result cache may be filled by output invalidated before re-use

Hidden parameter `_optimizer_ads_use_result_cache` (BOOLEAN) control result cache usage for Adaptive Dynamic Statistics
Conclusion
Conclusion
Summary

Result cache allows to keep results in memory instead of data

Result cache provides instant access to result sets
> Once cached, no need to do some work again for unchanged data
> Useful if users repeat some queries when there is no/few data changes

Result cache invalidation managed automatically

Maintaining data consistency has a cost
> Invalidation temporarily locks access to result cache
> Don’t use result cache on table with frequent invalidations
> Enable result cache on specific “static” targets
Any questions? Please do ask.

Nicolas Jardot
Senior Consultant
nicolas.jardot@dbi-services.com

Jérôme Witt
Senior Consultant
Mobile +41 79 961 27 73
jerome.witt@dbi-services.com

Looking forward to seeing you at booth 242!