Query Result Cache in Oracle Database 11g Release 1

Oracle 11g allows the results of SQL queries to be cached in the SGA and reused to improve performance.

Setup

Set up the following schema objects to see how the SQL query cache works.

```
CREATE TABLE qrc_tab (

  id  NUMBER

);

INSERT INTO qrc_tab VALUES (1);

INSERT INTO qrc_tab VALUES (2);

INSERT INTO qrc_tab VALUES (3);

INSERT INTO qrc_tab VALUES (4);

INSERT INTO qrc_tab VALUES (5);


CREATE OR REPLACE FUNCTION slow_function(p_id  IN  qrc_tab.id%TYPE)

  RETURN qrc_tab.id%TYPE DETERMINISTIC AS

BEGIN

  DBMS_LOCK.sleep(1);

  RETURN p_id;

END;

/

SET TIMING ON
```

The function contains a one second sleep so we can easily detect if it has been executed by checking the elapsed time of the query.

Test It

Query the test table using the slow function and check out the elapsed time. Each run takes approximately five seconds, one second sleep for each row queried.

```
SELECT slow_function(id) FROM qrc_tab;

SLOW_FUNCTION(ID)
-----------------
        1
        2
        3
        4
        5

5 rows selected.

Elapsed: 00:00:05.15
SQL>
```

Adding the RESULT_CACHE hint to the query tells the server to attempt to retrieve the information from the result cache. If the information is not present, it will cache the results of the query provided there is enough room in the result cache. Since we have no cached results, we would expect the first run to take approximately five seconds, but subsequent runs to be much quicker.

```
SELECT /*+ result_cache */ slow_function(id) FROM qrc_tab;

SLOW_FUNCTION(ID)
-----------------
        1
        2
        3
        4
        5

5 rows selected.

Elapsed: 00:00:05.20

SELECT /*+ result_cache */ slow_function(id) FROM qrc_tab;

SLOW_FUNCTION(ID)
-----------------
        1
        2
        3
        4
        5
```

```
5 rows selected.


Elapsed: 00:00:00.15

SQL>
```

RESULT_CACHE_MODE

The default action of the result cache is controlled by the RESULT_CACHE_MODE parameter. When it is set to MANUAL, the RESULT_CACHE hint must be used for a query to access the result cache.

```
SHOW PARAMETER RESULT_CACHE_MODE


NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
result_cache_mode                    string      MANUAL
SQL>
```

If we set the RESULT_CACHE_MODE parameter to FORCE, the result cache is used by default, but we can bypass it using the NO_RESULT_CACHE hint.

```
ALTER SESSION SET RESULT_CACHE_MODE=FORCE;
SELECT slow_function(id) FROM qrc_tab;
SLOW_FUNCTION(ID)
-----------------
                1
```

```
              2

              3

              4

              5


5 rows selected.


Elapsed: 00:00:00.14


SELECT /*+ no_result_cache */ slow_function(id) FROM qrc_tab;


SLOW_FUNCTION(ID)

-----------------

              1

              2

              3

              4

              5


5 rows selected.


Elapsed: 00:00:05.14

SQL>
```

Scalar Subquery Caching

The query result cache does not work with scalar subquery caching.

```
SELECT (SELECT /*+ result_cache */ slow_function(id) FROM dual) AS result FROM qrc_tab;

    RESULT
----------
         1
         2
         3
         4
         5

Elapsed: 00:00:05.03
SQL>

SELECT (SELECT /*+ result_cache */ slow_function(id) FROM dual) FROM qrc_tab;

    RESULT
----------
         1
         2
         3
         4
         5


Elapsed: 00:00:05.03

SQL>
```