Improving the Performance of PL/SQL Function Calls from SQL

Tim Hall

Oracle ACE Director

Oracle ACE of the Year 2006

OakTable Network

OCP DBA (7, 8, 8i, 9i, 10g, 11g)

OCP Advanced PL/SQL Developer

Oracle Database: SQL Certified Expert

http://www.oracle-base.com

Books

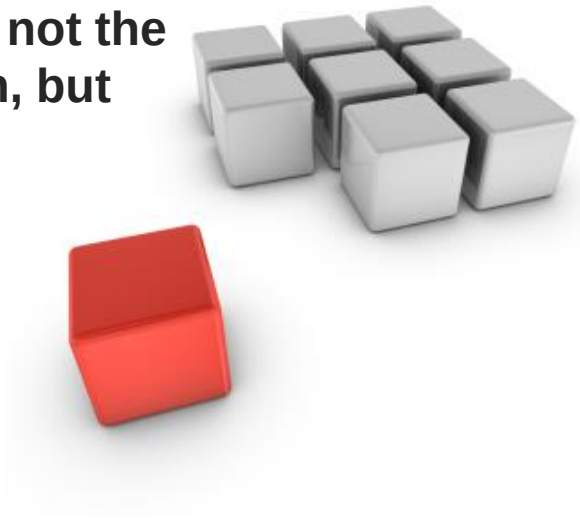
Oracle PL/SQL Tuning

Oracle Job Scheduling

What's the problem?

- We sometimes need to call PL/SQL functions in the select list of queries.
- By default, the function may be called for each row returned.
- If the function is called repeatedly with the same input parameters, this can represent a massive waste of resources.
- Sometimes we are able to change the query, but not the function. Sometimes we can change the function, but not the query.

(setup.sql)

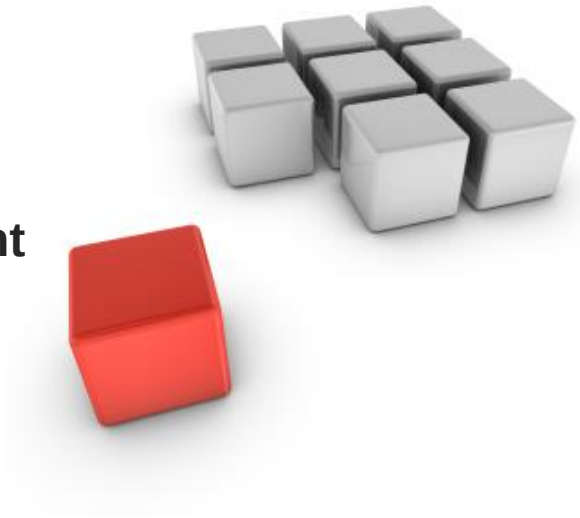


Scalar Subquery Caching

- Rewriting function calls as scalar subqueries allows Oracle to cache the results.

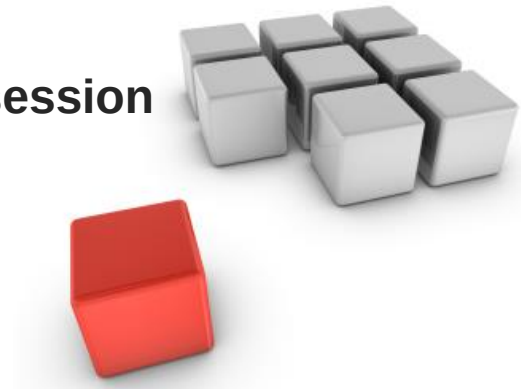
```
SELECT (SELECT slow_function(id) FROM dual)
FROM   func_test;
```

- Oracle sets up an in-memory hash table to cache results of scalar subqueries.
- The cache only last for the lifetime of the query.
- The cached values are not reusable in the current session or in other sessions.
- (scalar.sql)



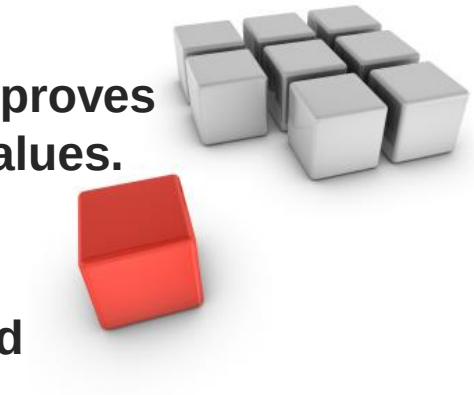
DETERMINISTIC Hint

- The DETERMINISTIC hint has been available for a long time, but didn't seem to do much until 10g.
- Oracle will optimize calls to functions marked as DETERMINISTIC to improve query performance.
- The caching is based on the array size of fetch, so mileage can vary.
- Cached return values only last for the lifetime of the call.
- The cached values are not reusable in the current session or in other sessions.
- (deterministic.sql)



11g Result Cache

- 11g introduced two new caching features.
- Both share the same pool of memory in the SGA, controlled using:
  - RESULT_CACHE_% parameters.
  - DBMS_RESULT_CACHE package.
  - V$RESULT_CACHE_% views. (result_cache.sql)
- The Query Result Cache improves performance of complex queries that return small number of rows. (query_result_cache.sql)
- The Cross-Session PL/SQL Function Result Cache improves performance of function calls by caching the return values. (plsql_result_cache.sql)
- Cached results can be reused in the same session and in other sessions.

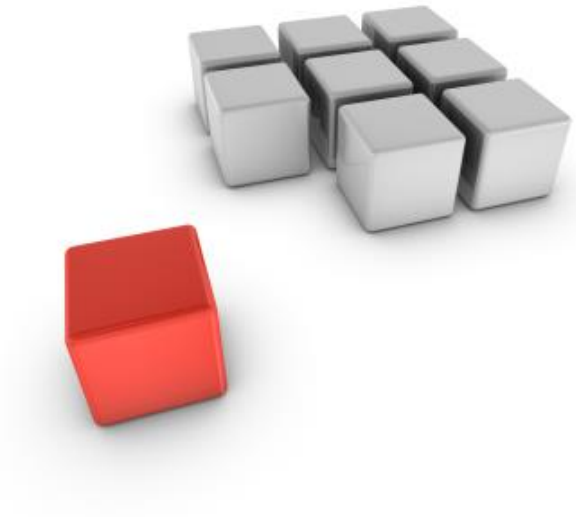


Manual Caching Using PL/SQL Collections

- Caching of function calls is nothing new.
- Caching using PL/SQL collections has been done for many years.
- Collections are session-specific, but cached values can be reused in multiple queries.
- (collection.sql)
- Manually caching can cause problems for volatile data.
- Remember, collections use memory. Don't go nuts!

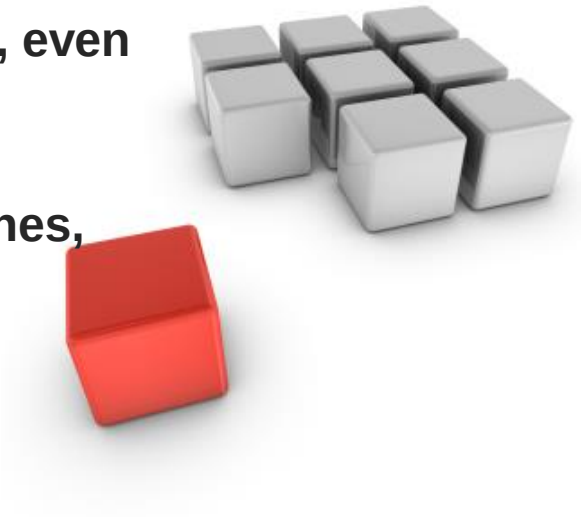
Manual Caching Using Contexts

- Manual caching using contexts is similar to using collections.
- It shares many of the same drawbacks, but can allow a shared cache between sessions.
- (context.sql)



Scalar Subquery Caching (Revisited)

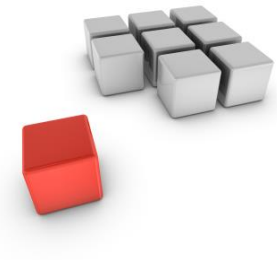
- Q: Do other caching methods make scalar subquery caching irrelevant?
- A: No. Alternative caching mechanisms don't reduce context switching between SQL and PL/SQL.
- (plsql_result_cache_2.sql)
- You should always use scalar subquery caching, even when using other caching mechanisms.
- Scalar Subquery Caching reduces context switches, but other methods have added benefits reuse between queries and sessions.



Is there anything relevant in 12c?

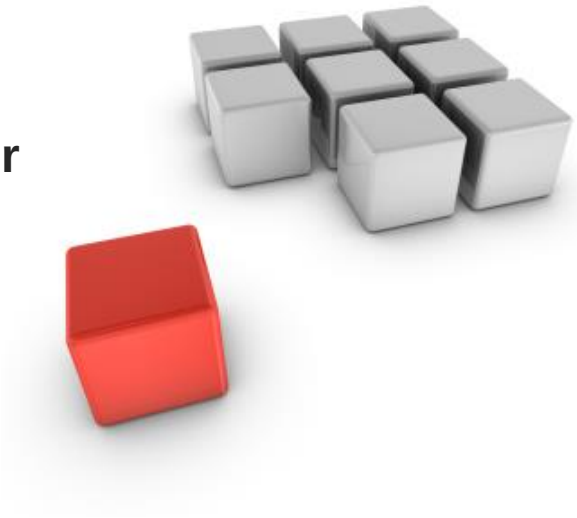
- Functions (and procedures) in the WITH clause. Reduces the context switching associated with PL/SQL calls, which improves performance.

```
WITH
  FUNCTION slow_function(p_id IN NUMBER) IS
  BEGIN
    RETURN p_id;
  END slow_function;
SELECT slow_function(id)
FROM   test_func;
```



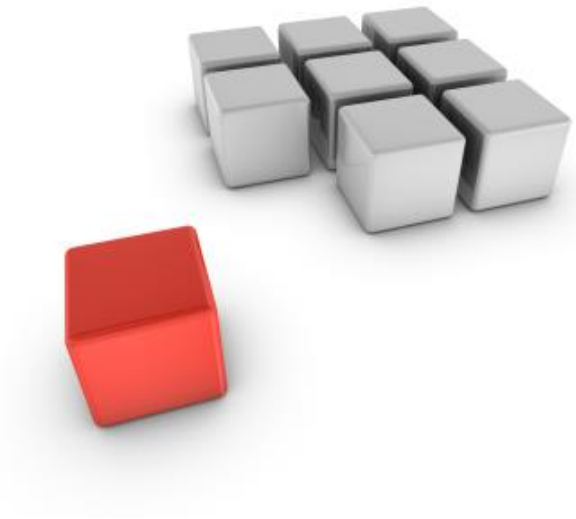
What about the FROM and WHERE clause?

- Function calls in the select list of an inline view follow the same rules as for the main select list.
- If you are using table functions, consider switching to pipelined table functions.
- Use Scalar Subquery Caching for functions in the WHERE clause if possible.
- Avoid functions on indexed columns, or consider function-based indexes.
  (fbi.sql)



What did we cover?

- Scalar Subquery Caching
- DETERMINISTIC Hint
- Cross-Session PL/SQL Function Result Cache
- Manual Caching Using PL/SQL Collections
- Manual Caching Using Contexts
- Function calls in the FROM and WHERE clause



The End…

- **Slides and Demos:**

  http://www.oracle-base.com/workshops

- **Questions?**