11g Cache Features
Put to the Test

Dave Anderson
SkillBuilders.com/11gCache
Agenda

- Introduce the 11g Caches
- Things you should know about the Query Result Cache
- Demonstrations
- Introducing the PL/SQL Function Result Cache
- Demonstration
Oracle 11g: New Caches

- **Server Result Cache**
  - Query Result Cache
  - PL/SQL Result Cache

- **Client Result Cache**
  - OCI-based database-aware client result cache
  - Cache query result sets in client memory
  - Automatic syncing with server
New in Shared Pool: Server Result Cache

Contains SQL Query Result and PL/SQL Result Caches
The SQL Query Result Cache

- Query Result Cache holds query result sets
  - Subsequent execution of equivalent* query uses cached result
  - Reduce I/O, sorts and computations

- Intended use
  - OLAP, DW, aggregation-type queries
  - Read many rows, output few
  - Against read-mostly, read-only data
  - Frequent execution of the query

- Automatically refreshed
  - Updates on dependents invalidate cached results
  - New result cached upon next query execution

- One might think about it this way...
  - "Dynamic, memory-only, materialized view"
The Query Result Cache

- Query needs to be the "equivalent" to use cached result
  - Must produce same hashed cache id
  - Result must be in cache,
    - and marked "Published"
- Different table alias or column aliases cause mismatch
- But whitespace, line breaks don't matter
- Result Cache Materialized Views (RCMV) help
  - More on this later
Configuring the Server Result Cache

- New initialization parameters
- parameters.sql

  - `result_cache_mode` MANUAL | FORCE | (AUTO?)
  - `result_cache_max_size` Default < 1% of shared pool. Disable RC by setting to 0.

  - `result_cache_max_result` Default is 5%
  - `result_cache_remote_expiration` Default is 0

- Also several undocumented parameters
  - e.g. `_result_cache_block_size`
Using the Query Result Cache

- Query or query block hint
  ```sql
  select /*+ result_cache */ customer_id, sum(sale_amount)
  from orders
  group by customer_id
  having sum(sale_amount) > 5000
  order by sum(sale_amount) desc
  ```

- Table Attribute (a.k.a. "annotation")
  ```sql
  alter table customers result_cache(mode force);
  ```

- Session-level
  ```sql
  alter session set result_cache_mode=force;
  ```

- Result Cache Materialized Views (RCMV)
  - Result cache for query rewrite
DBMS_RESULT_CACHE

- **BYPASS**
  - Disables use of cache - instance wide

- **FLUSH**
  - Flushes all objects from cache
  - Flushing shared pool does not flush result cache

- **MEMORY_REPORT**
  - Memory used by the result cache

- **STATUS**
  - Status of the result cache, ENABLED or BYPASS

- **INVALIDATE**
  - Invalidates all cached results based on object passed in

- **INVALIDATE_OBJECT**
  - Invalidates dependents in the cache
Dictionary and the Server Result Cache

- **V$RESULT_CACHE_DEPENDENCY**
  - Records relationship between "results" and "dependents"
  - dependencies.sql

- **V$RESULT_CACHE_MEMORY**
  - Each row describes an RC cache memory block
  - In-use? By what?

- **V$RESULT_CACHE_OBJECTS**
  - A row for each cached result and each dependent object
  - objects.sql

- **V$RESULT_CACHE_STATISTICS**
  - Instance-level RC statistics
  - For example, number of "creates", number of "finds"
  - cache_stats.sql
Query Result Cache Restrictions

► No active transaction on object in current session
  o Unless flashback query used
► NOT supported
  ▪ Temporary tables
  ▪ Tables in SYS or SYSTEM schemas
  ▪ Caching subquery results
  o Sequence CURRVAL and NEXTVAL pseudo columns
  o Date time functions
    ▪ SYSDATE, SYS_TIMESTAMP
    ▪ CURRENT_DATE, CURRENT_TIMESTAMP
    ▪ LOCAL_TIMESTAMP
  o USERENV / SYS_CONTEXT with non-constant variables
  o SYS_GUID
► All function calls must be deterministic
Query Result Cache Demo's

1. demo1.sql
2. autotrace.sql
3. latches.sql
4. alex.sql
5. flashback.sql
6. syntax.sql
7. table_attribute.sql
8. auto_result_caching.sql
9. active_trans.sql
10. subquery.sql
11. rcmv.sql
12. objects.sql
13. dependencies.sql
14. cache_stats.sql
Query Result Best Practices

► Look for queries that:
  • Process lots of rows and are considered expensive
  • Return few rows
  • Are executed frequently
  • Have limited number of possible bind variable values
  • Are based on read-mostly or read-only tables
  • Release 1 - have limited number of concurrent users
  • Release 2 - major enqueue and latching problems appear to be resolved

► Monitor
  • Create to Find ratio
  • Memory use, e.g. size, in-use vs free cache blocks
  • Contention
PL/SQL Function Result Cache

- Shared cache of function results
- Better than packaged-based collections
- Instance-wide sharing
- "Automatic" refresh
- Add one new keyword to get great performance benefit
Function Result Cache - Demo's

- Demo's scripts used by permission, Steven Feuerstein
- 11g_frc_demo.sql
Summary

► Query result cache
  • dynamic, memory-based, auto-refreshed mview – type functionality

► Function result cache
  • Instance-wide sharing of function results
  • Auto-refreshed

► OCI Client Result Cache
  • Client-side result caching
  • Auto-sync'd
Resources

- Alex Fatkulin, Pythian.com
- Julian Dyke, juliandyke.com
- Adrian Billington, oracle-developer.net
- Tom Kyte, oracle.asktom.com
- Christian Antognini, http://antognini.ch/blog/
Thank You

- Dave@skillbuilders.com

- SkillBuilders.com
  - Oracle Database Services
  - APEX Services
  - Weekly online, free webinars
  - Consulting
  - Remote, Onsite, Micro to Long-term
  - Training
  - Mentoring