

Guide to Database Tuning: Row Cache Hints and Tricks

A feature of Oracle Rdb

By Norm Lastovica
Oracle Rdb Relational Technology Group
Oracle Corporation

Guide to Database Tuning: Row Cache Hints and Tricks

With a number of sites running row cache in production, Oracle has gained some significant experience in running Oracle Rdb systems with the row cache feature. Here is a small list of some of the hints and tricks that we've learned along the way. Presumably, this'll help you to be successful with row cache. The list is in no particular order.

1. There are significant benefits from upgrading OpenVMS to at least version 7.3-1. Starting with this release, more S0 space is available for caches and their control structures. OpenVMS 7.3-1 also offers a faster lock manager along with a wide range of other performance improvements.
2. Sequential scans gain no real benefit from row cache (no I/O savings). You may wish to not cache tables that are always sequentially scanned. Instead, you should evaluate adding an index.
3. Tables that are insert-only typically will not gain from row caches (and may actually experience reduced performance). Sorted indexes on such tables will likely benefit from being cached.
4. When the database has snapshots enabled, very few modified rows will remain in cache. For these databases, checkpoint caches to the database rather than to backing store. A future Oracle Rdb release is expected to include the ability to include snapshots in row caches.
5. To cause an online flush (checkpoint) of modified rows from cache back to the database, use the `RMU /SERVER RECORD_CACHE /CHECKPOINT` command.
6. Remember that `RMU /BACKUP` forces a checkpoint of all modified rows from cache back to the database. This might take a while if there are many modified rows in caches.
7. When caching sorted indexes, make the cache slot size equal to the size of an index node (not a duplicate node).
8. Round slot count up to the next higher power of 2 if the current count is getting close to a power of two (for example, use 4096 slots instead of 4000). This technique, in some cases, can help reduce hash table collisions.

9. Remember that fragmented records are never stored in a row cache.
10. Try caching the RDB\$SYSTEM storage area in a physical cache and checkpoint that cache to the database. This speeds database attaches. Use the `RMU /ANALYZE /AREA` command to determine the total number of records in RDB\$SYSTEM. The number of cache slots should be equal to the number of records. Start with a slot size of 750 bytes
11. Avoid storing any user tables or indexes in the RDB\$SYSTEM area by using the `CREATE DATABASE ... DEFAULT STORAGE AREA IS ...` command to store unmapped objects in your own default storage area. Use a lists storage map to place your list of byte varying data elsewhere.
12. Use `RMU /SET ROW_CACHE [/ENABLE/DISABLE]` to quickly enable or disable row cache for a closed database. This can be handy when making cache changes or performing some maintenance tasks.
13. Set databases to `OPEN IS MANUAL`. Open and close the database by using the `RMU /OPEN /WAIT` and `RMU /CLOSE /WAIT` commands.
14. Use RCS & DBR log files (the logical names are `RDM$BIND_RCS_LOG_FILE` and `RDM$BIND_DBR_LOG_FILE`). Limit the size of an RCS log file with the logical name `RDM$BIND_RCS_LOG_REOPEN_SIZE` (define it to a value of perhaps 10,000 blocks).
15. Use reasonable RCS and fast commit checkpoint intervals (on the order of 15 minutes). Make sure to explicitly specify both. A longer interval generally results in better run-time performance, while a short interval generally results in faster database recovery from process or node failure. Current Rdb 7.1 releases include prestarted transaction timeout capabilities to help reduce recovery times.
16. Node failure recovery duration is related to the last completed fast commit checkpoint. You might want to run a batch job that does a `RMU /CHECKPOINT /WAIT` every 15 minutes to make sure that all processes do actually checkpoint.
17. Keep read-write transactions relatively short. Otherwise, they can block the RCS process from doing checkpoints with record locks.
18. Enable database operator notifications using `ALTER DATABASE ... NOTIFY IS ENABLED`. You can select which operators to alert in most cases.

19. If you observe very frequent database recovery processes (DBRs) - more than perhaps one per hour, consider increasing `RDM$BIND_RCS_CLEAR_GRICS_DBR_CNT` from the default of 25.
20. If you use `LARGE MEMORY (VLM)` for a number of caches, consider reducing the VLM window count from the default of 100. A window count of 25 is generally plenty and will save virtual address space.
21. System space buffers (`SHARED MEMORY IS SYSTEM`) and VLM (`LARGE MEMORY IS ENABLED`) require `PHYSICAL` memory. Make sure that you do not over-allocate the system and starve it for memory.
22. System space buffers require contiguous system (S0) address space. Make sure that you have enough (OpenVMS V7.2 or later helps here).
23. Rows deleted from the cache are not removed from their database pages until RCS flushes this change back to the database. For areas where space reclamation is important to you, make sure to checkpoint back to the database and not to the backing files.
24. The logical name `RDM$BIND_RCACHE_INSERT_ENABLED` disables cache inserts on a per-process basis. You might want to do this for processes that do massive deletes or updates.
25. Use `RMU/Show Statistics` with event notifications.
26. Define the system logical `RDM$BIND_RCS_INITIAL_MAP_ALL_CACHES` to "1" (Rdb V7.0.4 and later). This causes the RCS process to create and map all defined row caches for the database. The RCS will sort the cache definitions for 32-bit address space usage from largest to smallest before the caches are created.

For more information:

- Oracle Rdb7 Guide to Database Performance and Tuning
- Oracle Rdb7 Guide to Database Design and Definition
- Oracle Rdb7 Guide to Database Maintenance
- Oracle Rdb SQL Reference Manual
- Oracle Rdb Internals Course
- Oracle Rdb7 for OpenVMS Release Notes
- MetaLink
- Oracle Rdb home page www.oracle.com/rdb
- E-mail infordb@us.oracle.com

ORACLE

Oracle Rdb
Guide to Database Tuning: Row Cache Hints and Tricks
June 2003

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software
that powers the Internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2003 Oracle Corporation
All rights reserved.