Oracle Rdb's Record Caching Model

Richard Anderson
Oracle Corporation
rjanders@us.oracle.com

Gopalan Arun
Oracle Corporation
garun@us.oracle.com

Richard Frank
Oracle Corporation
rfrank@us.oracle.com

1. ABSTRACT

In this paper we present a more efficient record based caching model than the conventional page (disk block) based scheme. In a record caching model, individual records are stored together in a section of shared memory to form the cache. Traditional relational database systems have individual pages that are stored together in shared memory to form the cache and records are then extracted from these pages on demand. The record cache model has better memory utilization than the page model and also helps reduce overheads like page fetches/writes, page locks and code path.

In May 1996, Oracle Rdb announced a record breaking number of 14227 tpmC on a Digital AlphaServer 8400. At the time, this was the best TPC-C performance achieved on a single SMP machine. A total of 15 record caches, caching 19.5 million records, consuming almost 7 GB of memory, formed the bulk of the shared memory.

2. RECORD CACHE OVERVIEW

A record cache is a section of shared memory that contains copies of records. Record caches provide a means to store records with very little memory overhead. When a record is requested from the database, Oracle Rdb checks to see if a record cache exists for the record. If a record cache exists, and the requested record is in it, the record is retrieved. If the record is not in the record cache, the page buffer pool is checked. If the record is not in the page buffer pool, a disk I/O is done to retrieve the page containing the record. The record is then extracted from the page and inserted into the record cache. The page buffer pool thus only serves as a staging area for the records before they are inserted into the record cache. When a record cache is not defined for a record, the record resides in the page cache. The flow of data through the main memory of a system with record cache is shown in Figure 1.

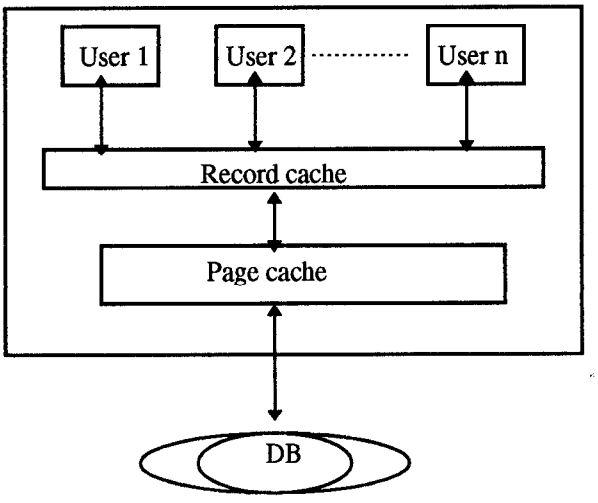

Figure 1

The record cache in Oracle Rdb is very flexible and can be configured to cache all database objects including b-tree nodes, hash buckets, metadata and data records. Properties like different replacement algorithms, size of record slots and number of record slots can be defined individually for each record cache. Each database can have one or more record caches.

2.1 Types of Record Cache

Oracle Rdb provides very effective data partitioning schemes. Tables and indexes can be horizontally and vertically partitioned. These partitions, each of which form a logical area, can then be assigned to physical storage areas (disk files). To provide maximum control over the database objects to be cached, Oracle Rdb provides two types of record caches;

Physical Area Record Cache (PARC) - These are caches defined for physical storage areas. A PARC can contain data from one or more physical storage areas.

Logical Area Record Cache (LARC) - These are caches defined for a specific table, index (or partition). Oracle Rdb automatically caches records from a logical area in a LARC when the name of the logical area matches the name of the LARC.

In order to keep the number of disk files within reasonable limits, physical storage areas often have records from more than one table. Obviously, it is now difficult to define one physical area record cache for this physical storage area and size the record slot in the record cache accurately. This problem can be solved by defining LARCs for tables that have widely differing record sizes and one PARC for all other tables in the physical storage area that have records of almost the same size.

2.2 Replacement Policy

Replacement policy can be set on an individual record cache basis. When a record cache has record replacement disabled, the data in the cache becomes memory resident. This behavior is useful, for example, to pin the non-leaf nodes of a large b-tree index. When a record cache with replacement disabled becomes full, records that cannot fit into that record cache reside in the page cache.

Enabling record replacement is beneficial when data access pattern is random. This policy ensures that the most frequently accessed records remain in memory. Record caches use a modified LRU replacement policy. Each database user process can protect from replacement, the last 10 rows it accessed in each cache. This group of records is called *Working Set*. Records belonging to a working set are considered to be referenced and thus are not eligible for replacement. Any record that is in the cache and is not in a working set is called an unreferenced record. Unreferenced records are eligible for replacement on a LRU basis.

2.3 Record Cache Server (RCS) Process

Oracle Rdb creates an RCS process when record caching is enabled on a database. The RCS process handles the task of writing modified records from the record cache to stable storage. The RCS process performs 2 distinct operations;

Checkpointing - The RCS process performs a novel, high speed checkpoint. In this scheme, records in the record cache that need to be checkpointed are batched together and written in large I/O chunks to an on-disk, sequential checkpoint file. Since the records are individually cached, the batching and sequential writing is easily achieved. Recovery after a node failure consists of reading the checkpoint file into the record cache and then performing conventional undo/redo. Thus, after recovery from a node failure, users begin with a warm record cache.

Sweeping - This refers to the task of writing modified records from the record cache to their regular physical storage areas in the database. The number of records the RCS process writes per unit of time can be modulated by means of a sweep threshold setting.

3. SUMMARY

In summary, record caches in Oracle Rdb are clearly superior to page caches. With the flexibility in partitioning that Oracle Rdb provides, any subset of records from any table or index can be cached/pinned in memory. As the benchmark result proved, it is possible to implement a record based caching scheme and have its algorithms scale well with large memory sections as well as high transaction throughput rates.

Both OLTP and datawarehousing applications will benefit from the better memory utilization of record caches. In addition, the performance benefit of shorter code path for data access, as compared to page caches, should also speedup these applications. The record cache feature has been fully integrated and tested and has been shipping with the Oracle Rdb7 database release.