



PERFORMANCE EVALUATION AND TUNING IN AN ORACLE DBMS

BOICEA, A[lexandru]; CRIVAT, A[lexandru]; RADULESCU, F[lorin] & POPA, G[eorge]

Abstract: *The current paper presents the main ways of monitoring and tuning of an Oracle instance. The issues discussed are : Requirements for tuning, description of oracle objects and utilities used and methods for SQL, Shared Pool, Buffer Cache and Redo Log Buffer optimization.*

Key words: *oracle, database, tuning, performance evaluation*

1. INTRODUCTION

Before beginning Oracle specific tuning one should check for bottlenecks on the operating system level. In particular, determine if there is excessive paging to the swap file and whether other applications are contending for system resources (such as RAM, CPU and disks) with Oracle. The following aspects should be taken into consideration:

Database Design : Many times it will not be possible to participate at this level but when possible, being able to lay down a good design initially will help prevent a lot of performance nightmares from ever occurring.

Memory tuning : One must make sure the SGA is tuned correctly. The SGA is composed of three parts: 1) the data block buffer cache, 2) the redo log cache and 3) the shared pool. All three need to be tuned properly for optimal performance.

Disk I/O: One must make sure the tablespaces are laid out correctly so that we don't have one type of tablespace adversely affecting another. An example of this is putting a rollback segment tablespace on the same disk as a data tablespace. Also verify that the storage parameters are set optimally.

Internal Memory Structure Contention : Verify one internal process isn't waiting upon another for long periods of time. This is done by monitoring and adjusting latches correctly. (Troy technologies USA, 2001)

2. PERFORMANCE EVALUATION OF THE DATABASE

Information about the status of the database can be found in trace files and system views.

Trace files are Oracle log files which give the status of Oracle events. The main overall trace file which monitors the instance is called the alert.log. It is located in the directory specified by the BACKGROUND_DUMP_DEST initialization parameter. Trace files (except for alert.log) usually end with a .trc extension. User process trace files go in the directory specified by the USER_DUMP_DEST initialization parameter.

The V\$ views are dynamic performance views based on the X\$ tables. X\$ tables are internal tables which hold information about the instance. Both are owned by SYS and populated at instance startup. V\$ views can be seen by anyone with the "SELECT ANY TABLE" object privilege while X\$ tables can only be viewed by SYS. The views which show information about processes waiting for resources are: V\$SYSTEM_EVENT, V\$SESSION_EVENT and V\$SESSION_WAIT.

V\$SYSSTAT is the main view for system performance on the instance level. It can also be used to monitor client-server traffic. V\$SESSION gives connection information for all user sessions. Based on the system views there are a set of scripts integrated

in oracle that generate performance reports: ULTBSTAT.SQL and UTLESTAT.SQL. There is also a set of GUI applications that come with the Oracle Enterprise Edition version of Oracle Enterprise Manager called the Oracle Performance Pack. These programs help monitor and optimize database performance (Oracle Expert - assists in configuring and tuning the database, Lock Manager - monitors locks, Performance Manager - monitors real time performance statistics and views them in various ways, Tablespace Manager - monitors segment storage in tablespaces. Also performs the defragment of tablespaces (know as coalescing, TopSessions - like the Unix 'top' command, allows the monitoring of the top resource intensive user sessions, Oracle Trace - monitors the performance within user applications). (Troy technologies USA, 2001)

3. SQL TUNING

There are two optimizers built into Oracle: Rules Based and Cost Based. The Rules Based Optimizer (RBO) is a set of 15 rules Oracle uses to determine the fastest path of execution for a given SQL statement. The Cost Based Optimizer (CBO) determines all possible paths of execution and assigns a cost to each one. It chooses what it finds to be the least expensive path. ANALYZE must be run on all tables and indexes to generate statistics in order to use the Cost Based Optimizer.

It is very important to update statistics on a regular basis to provide meaningful data for the CBO to work with. Performance can suffer tremendously by using outdated statistics. The CBO goal can be set to either; 1) fastest overall throughput for a SQL statement or, 2) fastest initial response time for a SQL statement. The optimizer can be set at the session level by issuing the command:

ALTER SESSION SET OPTIMIZER GOAL=*optimizer mode*

The EXPLAIN PLAN FOR *sql statement* can be used to see how the optimizer is executing the statement. This puts the execution plan in the PLAN_TABLE.

SQL Trace is used to gather user session statistics. It generates a trace file in the USER_DUMP_DEST location. SQL Trace can be turned on at the instance level by setting the initialization parameter SQL_TRACE=TRUE. SQL Trace can also be turned on for an individual session by issuing: ALTER SESSION SET SQL_TRACE=TRUE. TKPROF is used to read the output of a trace file created by SQL Trace. The DBMS_APPLICATION_INFO package is created by the DBMSUTIL.SQL. It allows the DBA to track resource usage and performance data for PL/SQL procedures. DBMSUTIL.SQL is called by the CATPROC.SQL script so it should already exist in the database. (Kilpatrick et al., 2001)

4. TUNING THE SHARED POOL

The shared pool is composed of the library cache, row cache and if MTS is used, the User Global Area (UGA). The size of the shared pool is set by the initialization parameter SHARED_POOL_SIZE (in bytes).

The library cache contains shared parse information for SQL statements. This is the main area to monitor in the shared pool. The initialization parameter SHARED_POOL_RESERVED_

SIZE specifies how much of the shared pool has to be set aside for the reserved list. This is an area of the library cache to store large objects in. Objects smaller than the value specified by the initialization parameter SHARED_POOL_RESERVED_MIN_ALLOC will not be allowed on the reserved list.

The User Global Area (UGA) is an additional third area of the shared pool when Oracle runs in Multithreaded Server (MTS) mode. When running in MTS mode, user information stored in the PGA in dedicated server mode is stored in the UGA instead. Hence, the UGA needs to be taken into consideration for overall sizing of the shared pool. Specifically one needs to calculate the additional amount of memory required by the shared server sessions and open cursors.

5. TUNING THE BUFFER CACHE

The size of the buffer cache is determined by: DB_BLOCK_SIZE * DB_BLOCK_BUFFERS.

DB_BLOCK_SIZE cannot be changed without recreating the database. The buffer cache contains the dirty buffer write queue, which holds dirty block buffers (i.e., modified blocks) until they can be written out to disk. A least recently used (LRU) algorithm is used to decide which buffers to move out of the buffer cache so that new buffers can be read in.

The size of the default buffer pool is not explicitly defined. Instead it is equal to the size of the entire buffer cache (value of DB_BLOCK_BUFFERS) minus the number of blocks allocated to the keep buffer pool and/or the recycle buffer pool. The buffer blocks for the keep and recycle buffer pools are taken from the buffer cache. Likewise, their LRU latches are taken from the latches allocated for the entire buffer cache as specified by DB_BLOCK_LRU_LATCHES. Therefore, neither the number of buffer blocks nor the number of LRU latches allocated to the keep and recycle buffer pools can, taken together, equal or exceed the values allocated to the buffer cache as a whole. If we over-allocate either DB blocks or LRU latches by accident, the database will not mount.

The CACHE hint (/*+ CACHE */) is used in the SQL statements to ensure that the selected table will be cached. Also the CACHE clause can be used when creating a table to make sure it will always be cached. Only small, frequently used tables should be cached. By default, large tables will fill the buffer cache (unless assigned to the keep or recycle buffer pool).

V\$CACHE shows what objects are currently in the buffer cache. CATPARR.SQL is run to create this view. While CATPARR.SQL is for Parallel Server environments, V\$CACHE is useful in single instance environments as well.

6. TUNING THE REDO LOG BUFFER

The LOG_BUFFER initialization parameter specifies how big (in bytes) the redo log buffer is. V\$SYSTEM_EVENT is queried to determine if there are waits for 'log buffer space'. If so, the redo log buffer size should be increased. This information can also be obtained from V\$SYSSTAT by looking at the number of 'redo buffer allocation retries' there are for a given user process.

7. DATABASE CONFIGURATION

The LOG_CHECKPOINTS_TO_ALERT initialization parameter has to be set to TRUE so that checkpoint beginning and ending times are logged to the alert.log file.

The DB_BLOCK_CHECKPOINT_BATCH specifies the maximum number of blocks that a DBWR process can write in a single batch during a checkpoint. Increasing this value can

speed up checkpoint times, but making it too large can give poor response times as well.

The initialization parameters DISK_ASYNC_IO and TAPE_ASYNC_IO specify whether the operating system supports asynchronous I/O for hard drives and tape drives respectively (most do). The default value is TRUE.

The SYSTEM tablespace should only contain data dictionary objects, PL/SQL packages, triggers and the initial rollback segment. The database should have a minimum of six tablespaces. Heavy I/O on the SYSTEM tablespace should be avoided. (Powell, 2004)

8. USING ORACLE BLOCKS EFFICIENTLY

PCTUSED is relevant for deletes. If there are a lot of inserts and deletes on a table, PCTUSED has to be set high. On DSS systems PCTUSED is irrelevant. PCTFREE and PCTUSED taken together should be less than 100.

The highwater mark for a table is increased 5 blocks at a time. Full table scans read up through the highwater mark. DELETE does not reduce the highwater mark count on a table, but TRUNCATE does. So does ALTER TABLE tablename DEALLOCATE UNUSED. (Stuns et al., 2005)

9. OPTIMIZING SORT OPERATIONS

The following clauses cause sorting to occur: ORDER BY, GROUP BY, DISTINCT, UNION, INTERSECT and MINUS. Oracle will sort in memory if the sort can fit in the value specified by the initialization parameter SORT_AREA_SIZE (in bytes). If not, Oracle will break the sort into multiple sort runs. MAXEXTENTS is not a valid storage parameter for temporary tablespaces.

The initialization parameters SORT_WRITE_BUFFERS and SORT_WRITE_BUFFER_SIZE specify how much memory to allocate for direct write sorts. The SORT_WRITE_BUFFERS should be between 2 and 8. SORT_WRITE_BUFFER_SIZE should be between 32K and 64K.

10. CONCLUSIONS

The paper presents in few details most of the necessary steps that need to be taken to assure that an oracle DBMS performs optimal. The future work will include a more detailed approach on each step of the optimization process. The Database used will be an oracle version 10g. It will run on a virtual machine with limited resources. Time and resource consuming tasks will be chosen and run before and after the tuning of the database parameters. Performance evaluation will be run before and after the alterations of the database and the results will be compared to measure the improvements.

11. REFERENCES

- Stuns D.; Buterbaugh T. & Bryla B. (2005), *Oracle 10g™ Administration II* ISBN: 0-7821-4368-7, San Francisco
- Powell G. (2004), *Oracle High Performance Tuning for 9i and 10g*, Digital Press
- Kilpatrick P., Raman S. & Womack J., *Oracle 9i Performance Tuning*, Student Guide, Volume 1, July 2001, Oracle Corporation.
- Troy technologies USA, *Oracle 8: Performance Tuning* (2001). Available from <http://www.troytec.com> Accessed on: 2009-10-05
- Mark Gurry (2002). *Oracle SQL Tuning Pocket Reference*, O'Reilly, ISBN : 0-596-00268-8