LET'S TUNE ORACLE8 FOR NT

Marlene Theriault, DBAmentor.com

Abstract

When it comes to tuning an Oracle8.0 or 8i database, the Windows NT environment poses a whole new set of problem for the DBA. Since Oracle on Windows NT is a single executable with multiple threaded processes, how does the DBA approach tuning this machine and the databases that reside on it?

This paper will examine the approaches and tools available for tuning a database on Windows NT as well as some of the impacts of running more than one database on an NT server.

Scope

There are often many different approaches that can be used when setting out to write a paper for a conference. There are also many different levels of assumption that can be made (or not made) as to the level of knowledge of the reader. In this paper, I am going to assume:

- The reader knows enough about Windows NT to have used the Task Manager, Resource Kit, and has examined the Windows NT Registry (although I'll talk a bit about the registry in the paper)
- The reader understands that nomenclature like Start→|Settings→|Control Panel→|Services means to depress the Start Icon on the toolbar, select the Settings option and then the Control Panel option and, finally, double click on the Services option
- The reader has basic Oracle Database Administration (DBA) experience and background
- The reader has knowledge about how to verify and tune the various database areas like the library cache, buffer cache, latches and locks, etc.

Throughout the paper, I'll be including information about Net8 in a Windows NT environment since so much of the interaction with NT for Oracle is through the Net8 networking protocol.

Because this paper has grown very large just with NT specific information, I have NOT included basic tuning information that is valid across all Oracle databases. I've eliminated the determination and actions to use to tune the library cache, buffer cache, dictionary cache, sort area, redo log thread switches, redo latches, and various wait statistics. I have, however, provided an overview of V$WAIT_EVENTS view and how to use it to evaluate there the most major contention lies in your database.

A Bit about Basic Tuning Concepts

Although I am going to focus on Oracle tuning in a Windows NT environment, I wanted to take a few minutes up front to talk about tuning in general. There are many different levels at which you can tune in an Oracle environment including:

- Both the physical and logical design levels – disk layout (tablespace/datafiles) and object layout (tables, indexes, etc.)
- The application level – tuning the SQL
- The operating system level – OS resources like memory and disk storage capacity
- The network – both bandwidth and latency
- The database level – including the memory structures

At the heart of effective tuning lies the SQL used to communicate with your databases. If your applications aren't tuned and you have inefficient SQL statements, you can spend your life trying to tune everything else perfectly and you will never have a really well tuned system!

The other concept to consider is Pareto's Rule, which says that 20% of the work will consume 80% of the resources. This means that placing your efforts in attempting to alleviate the top 10 – 20% of resource contention may take a very few hours per week and provide the most benefit to your system. You must decide what acceptable performance is before you start to tune so that you will know when to STOP tuning.

Since all performance problems are resource contention related, you want to concentrate on where the real contentions are coming from – the database, operating system, network, etc. By determining the highest contentions on your system and relieving them, you should be able to enhance the overall performance substantially and quickly.

I'll talk more about these areas later with an overview of how to determine where contentions lie and what you can do to improve your performance. Your daytime performance may be very different than your nighttime task load. If you have a mix of both OLTP (rapid, frequent, small inserts and updates) and DDS (long, large data loads coupled with large, full table scans), when you gather your statistics may end up providing misleading results.

Remember, too, that tuning is an on-going process! The development database and new production database will require different tuning approaches than the database that has been running in production for a year or more. You will want to perform on-going statistic collection and analysis to enable you to trend for current and future resource needs.

A Look at Oracle on a Windows NT System

Let's talk a bit about how Oracle is configured in an NT environment so that you can work with the system more easily.

NT is a 32-bit operating system and supports both CISC and RISC architectures. Like Unix and OpenVMS, NT has adopted the global networking standards such as TCP/IP and SNMP. Oracle recommends that NT users use the TCP/IP communications protocol. There are two modes in which processes can execute: Kernel and User. **Kernel mode** is a highly privileged mode in which the code has direct access to memory – with operating system memory protected by the processor. **User mode** is used for application execution and APIs must be used to switch a process's threads from User to Kernel mode and back again; i.e., from lower privilege to higher and back again.

Processes Versus Threads

If you look at a Unix system, you will see that there are several identifiable processes. In the example shown in the following listing, the **–v** clause tells Unix to select everything that does not match the following string. The command says to look for all processes that contain the phrase "ora" but to exclude the words "LOCAL" and "grep". In the output, shown following the command, there are nine detached Oracle processes and the listener running on this Unix system.

```
ps -ef|grep ora|grep -v LOCAL|grep -v grep

oracle 221 1 0 Jun 15 ? 37:03
/opt/oracle/product/8.1.5/bin/tnslsnr LISTENER -inherit
oracle 258 1 0 Jun 15 ? 307:15 ora_dbw0_test
oracle 254 1 0 Jun 15 ? 0:01 ora_pmon_test
oracle 260 1 0 Jun 15 ? 0:40 ora_arch_test
oracle 262 1 0 Jun 15 ? 26:32 ora_lgwr_test
oracle 264 1 0 Jun 15 ? 1:43 ora_ckpt_test
oracle 266 1 0 Jun 15 ? 0:10 ora_smon_test
oracle 268 1 0 Jun 15 ? 0:02 ora_reco_test
oracle 272 1 2 Jun 15 ? 1991:58 ora_d000_test
oracle 19971 1 1 Jul 28 ? 197:17 ora_s000_test
```

The output shows the background processes for the TEST instance.

The NT architecture provides a built-in multithreading capability so a single executable can perform many different tasks at essentially the same time. In an NT environment for Oracle8i, each detached process runs as a concurrently running **thread** within a single executable called Oracle.EXE. (In Oracle8.0, the executable is named Oracle80.EXE.) Using the same executable with several threads, the threads all share the same code, memory space, and other structures.

In other operating systems, each separate process requires its own separate code implementation. By having multiple threads within one executable, Oracle reduces the number of code and data segments and stacks required for processing, reducing the amount of memory required. However, any memory operations that are performed affect all of the threads.

You can use the Windows NT Task Manager to view the processes that are currently running on your NT system. You can access the Windows NT Task Manager by depressing the CTRL-ALT-DELETE keys at the same time and then selecting the Task Manager option from the displayed box. Once the Task Manager screen is displayed, choose the Processes tab.

As of Oracle8.0.4, you can have multiple ORACLE_HOME directories with different versions of the RDBMS running at the same time. This means that you can have v7.3.4, 8.0.5, and 8i databases all running on the same machine! Since there are multiple threads for each process, you will not be able to easily recognize what threads belong to which Oracle process. You will, therefore, need to rely on either using the Services window or commands issued at the DOS prompt to start and stop the Oracle processes and listeners on your system. You can actually have one listener for each version running on your system at the same time but should really only use the highest version one to connect to all of the databases on your machine.

You can see the Oracle threads by accessing the Performance Monitor option of the Task Manager. Sequential thread numbers are used to identify each thread, but no thread names are supplied. Oracle threads are assigned in the following order. Thread 0 is the dispatcher thread. It handles input from the Listener process (started through the Services). Thread 1 is the worker or execution thread that handles processing as requested by the dispatcher thread. The worker or execution thread usually handles launching another thread on behalf of the incoming connection on dedicated servers. Table 1 shows the basic thread assignments after the dispatcher and worker or execution thread assignments. Table 1 reflects the basic thread assignments used commonly in both Oracle8.0 and 8i.

Thread Name	Thread#
PMON	2
DBW0	3
LGWR	4
CKPT	5
SMON	6
RECO	7

Table 1: Basic Oracle Thread Assignments

If archive logging is enabled, the ARCH process will be assigned as thread 4 and the other threads will move down a number with LGWR becoming 5, CKPT becoming 6, etc. In Oracle8i, the heterogeneous service agents have been multithreaded in an approach similar to the Oracle multithreaded server (MTS) architecture and there is now a set of dispatchers to receive requests from Oracle database server processes and return results to them. There is also a pool of task threads to process the requests and compute results. In this way, a greater number of concurrent user sessions can be accommodated. Although individual thread names of Oracle processes are not visible from the NT operating system (neither from the Task Manager nor the Services window), you can find them in the alert log when the database is started. A sample section of the alert log shows the following:

```
PMON started with pid=2
DBW0 started with pid=3
LGWR started with pid=4
CKPT started with pid=5
SMON started with pid=6
RECO started with pid=7
```

You can also obtain the list of threads and their assignment through SQL*Plus via the following query:

```
select b.Name bkpr, s.Username spid, p.Pid
  from V$BGPROCESS b, V$SESSION s, V$PROCESS p
 where p.Addr = b.Paddr(+)
   and p.Addr = s.Paddr

BKPR  SPID                                 PID
----- ------------------------------ ---------
PMON                                         2
DBW0                                         3
LGWR                                         4
CKPT                                         5
SMON                                         6
RECO                                         7
SNP0                                         8
```

```
SNP1                                         9
      DBSNMP                                12
      SYS                                   13
```

Note: The listener and the dispatcher threads do not show up in this list.

In this way, you can identify which process is associated with each thread. However, as each process ends, the number is reassigned as needed. Thus, it can be hard to monitor a user by watching a thread. If you have it available on your machine, you can also use the TLIST command from the Resource Kit to observe the threads of a process.

About Oracle Services

Each Oracle instance on an NT system runs as a separate **service** or executable program that NT can keep track of and start up whenever NT is started. A service can run in the background or be configured for foreground interaction with users. A service can be started and remain running even when no users are logged on to the system. This is an important feature because you do not want your database server to rely on a specific process being logged on to the machine to keep the database or listener running.

To access the Services window, choose the Start→|Settings→|Control Panel option. Within the Control Panel, select the Services option (the icon is two gears and is labeled "Services"). In Oracle8.0, two separate service entries are displayed in the Services window, while in Oracle8i only one entry is displayed. To create a service for a new database, you can use the command ORADIM in 8i from a DOS command line or ORADIM80 for v8.0. Without any parameters, issuing this command will provide you with the command syntax.

The NT Registry

On a Windows NT server, a central repository is used to keep track of the installed hardware and software. This repository is called the **Registry**. Each Oracle database and all of the Oracle software paths will be listed in the registry. The important root keys of the registry for Oracle installations are as follows:

- HKEY_LOCAL_MACHINE (HKLM): contains information on Hardware, Security Accounts Manager, Security, Software, and System
- HKEY_USERS (HKU): contains a subkey for each user account on the local machine
- HKEY_CURRENT_CONFIG (HKCC): contains information about the entire system
- HKEY_CLASSES ROOT (HKCR): contains information about file associations and application-specific data for OLE
- HKEY_CURRENT_USER: contains a profile for the current user (actually a pointer to HKCU/SID)

You can get to the Registry by running REGEDIT.EXE or REGEDT32.EXE from Start→|Run and the program name. Why do you care about the Registry? The Registry enables you to see how many different ORACLE_HOMEs you have installed by looking in the HKEY_LOCAL_MACHINE \SOFTWARE \ORACLE\ALL_HOMES\idx structure. The ALL_HOMES values will include the number of ORACLE homes using the parameter HOME_COUNTER and the last ORACLE_HOME will be identified by the parameter LAST_HOME with a number. Each separate ORACLE_HOME will have an ID with a different number such as ID0 and ID1. In each IDx listing, there will be a PATH parameter with a directory location.

Under the HKEY_LOCAL_MACHINE\SYSTEM\CurrentcontrolSet\Services, is the entry for each service that is displayed from Start→|Settings→|Control Panel→|Services. This information can be vital if you need to "clean up" manually should an Oracle8i installation fails and you have to begin again!

You must move very cautiously when making any changes or taking any actions within the Registry since a "slip" could end up forcing you to rebuild your entire NT and Oracle systems (which is not a great way to spend your life!)

About CPUs

Monitor your CPU usage carefully. If you see the CPUs "pegged" at 100%, the CPUs are the bottleneck. You'll want to verify the percentage of time that they are pegged at 100%. If they are only pegged at 100% for brief periods of time, that's not a big bottleneck. However, if you see the CPU usage remaining at 100% for longer periods of time, you will want to begin budgeting for additional (faster) CPU resources.

The two options that you have if CPUs are the bottleneck are to either add more CPUs or replace the current ones with faster ones. Remember that replacing 4 – 200 Mhz Pentium Pro CPUs with 4 – 400 Mhz Xeon processors will result in a greater overall improvement than adding 4 additional 200 Mhz Pentium Pros.

Using Memory

Because NT is a 32-bit operating system, the maximum memory that is addressable in this environment is 4 gigabytes (GB). However, NT reserves 2GB for itself so only 2GB of memory are available for applications to use. With Service Pack 3, 3 GB of memory become available for applications.

NT is a **virtual memory system**. Management of the virtual memory is done in equally sized pieces called **pages** and the physical memory is managed in 4-kilobyte (KB) sections called **page frames**. Each process is allocated a quantity of page frames in physical (real) memory. A page that is not in physical memory is considered to be invalid. A **page fault** occurs when a thread of a process requests a page that is marked as invalid. The NT memory manager must get the page of information from disk and load it into a page frame in physical memory. When a page frame becomes free, the memory manager copies the pages back to disk to make room for new pages. Thus, when the CPU is looking for data and it's not found in physical memory, it is loaded from virtual memory (the page file). This action is called **paging**. If an entire process is moved out of physical memory, it's called **swapping**. In reality, NT does not perform swapping. The term is kept around for historical reasons. The movement of pages out of memory has been referred to as "swapping" with the file that receives the pages referred to as a "swap file."

A first in/first out (FIFO) system is used. Each process gets a guaranteed minimum number of pages in memory called the **minimum working set size**. If there is enough free memory available, a process's memory usage can grow to its **maximum working set size**. Memory allocated to a thread is called **reserved memory** and the portion of virtual memory that has space reserved on disk is called **committed memory**.

You can monitor the memory used by a process under the Column Memory Usage of the Task Manager Process tab. You can monitor the page fault rate on an NT system by using **pfmon** in the Resource Kit.

The default page size on a Windows NT system is 4KB. On a Compaq Alpha system, it's 8KB. For better performance, create your Oracle database with either a 4KB or 8KB block size to coincide with the NT page size. Because NT grows and shrinks the working set, portions of the SGA could be paged out. This action can adversely affect processes in a decision support system (DSS) because the buffers would need to be recreated each time information is needed. Therefore, you'll want to size the SGA to balance the Oracle hit ratio and the O/S paging. You can also add more physical memory to the server if your value is lower than the maximum allowed.

Check Disks

Watch the volume of disk reads and writes using the following script:

```
prompt This looks at overall i/o activity against individual files within a
prompt tablespace
prompt
prompt Look for a mismatch across disk drives in terms of I/O
prompt
prompt Also, examine the Blocks per Read Ratio for heavily accessed
prompt TSs - if this value is significantly above 1 then you may have
prompt full tablescans occurring (with multi-block I/O)
prompt
prompt If activity on the files is unbalanced, move files around to balance
prompt the load.  Should see an approximately even set of numbers across files.
prompt

set pagesize 100;
set space 1
column pbr format 99999999 heading 'Physical|Blk Read'
column pbw format 999999 heading 'Physical|Blks Wrtn'
column pyr format 999999 heading 'Physical|Reads'
column readtim format 99999999 heading 'Read|Time'
column name format a40 heading 'DataFile Name'
column writetim format 99999999 heading 'Write|Time'
ttitle center 'Tablespace Report' skip 2
compute sum of f.phyblkrd, f.phyblkwrt on report
rem
select fs.name name,f.phyblkrd pbr,f.phyblkwrt pbw,f.readtim,
f.writetim
from v$filestat f, v$dbfile fs
where f.file# = fs.file#
order by fs.name
/
```

You can monitor the datafile disk usage over time and modify the location of files on disks to better balance the disk access. You can add physical disks as needed or replace the disks or arrays with faster disk and/or array controllers. Adding memory can help too if the SGA hit ratios are below ideal. By increasing memory, you can reduce the amount of "trips back to disk". You can also perform additional application tuning to decrease the trips to disk by adding indexes where appropriate – reducing full table scans whenever possible.

Maximum Sizes

The following are the maximum sizes that you can have in an Oracle 8.0 database on Windows NT:

Database Feature	**Maximum Size**
Maximum Oracle block size	16KB
Maximum blocks per data file	4 million
Maximum data files per database	2KB – 20,000, 4KB – 40,000, 8KB – 65,535
Largest data file	64GB (assuming 16KB Oracle block size)
Largest database	4PB (Petabytes)

For better performance, use more files with fewer blocks per file to spread the data over a larger area.

The limit on the number of concurrent users on an NT system depends on whether you use Multi-threaded Server, connection pooling, or dedicated connections. You see, as more process threads are created to accommodate user connections, NT will have more elements to scan in the thread list – slowing down processing. Remember that Oracle on NT will launch a user thread for each connection. Thus, Oracle on NT tends to run out of memory before reaching about 500 concurrent connections in a dedicated connection environment. By using MTS or connection pooling, you can increase the number of users accessing the system but decrease the number of actual connections.

Configuring Windows NT as a Back-end Server

In the preceding sections, you have seen the basic tuning steps that you can take to improve performance on an NT system. In this section, you will see how to configure the NT machine as a more efficient and effective database server

On an NT system, you have a potential 4 GB of memory space available for addressing. Half of the addressable space is dedicated to the system services and the file cache. User programs, such as Oracle8i, can use the rest of the available space. Because most Windows NT systems do not actually have 4 gigabytes of real memory installed, the systems use a mechanism called *virtual memory* to balance the use of real memory among applications. The Windows NT system will use a virtual memory paging file to dynamically swap memory pages between physical RAM and the virtual paging area. Since Oracle8i is a memory intensive application, the amount of memory requirements can exceed the actual physical memory capacity of the machine.

There are several steps that you can take to configure Windows NT as a backend server. The first of these steps is to ensure that the machine is not used for any other activities like a primary or backup domain controller, a file or print server, a remote access server, or a router or firewall server. In the following sections, you will see descriptions of the other steps.

Reduce the priority of interactive foreground applications

The next action that you can take is to reduce the priority of interactive foreground applications on the server console since they are given priority over background processes. To reduce the priority of interactive foreground applications and, thus, increase the resources available for Oracle8i, perform the following steps.

1. Choose the System option from the Control Panel under the Settings menu.
2. Click the Performance tab.
3. Move the Application Performance Boost slider to None.
4. Ensure that the Virtual Memory paging file size is set to at least 300 MB.
5. Click OK on the dialog box.
6. Exit the Control Panel.

If you have more than one physical disk installed on your system, you can span your virtual memory paging file across the physical volumes. If you have logical, instead of physical, disks on your system, spanning your virtual memory paging file across them will not be worthwhile. If you have changed the virtual memory page file size, you must reboot your system for the change to take effect.

Reduce the Windows NT Server file cache

Normally, Windows NT reserved 41% of the available memory for file cache activities on the system. While this is a good use of resource for a machine acting as a file server, it is unnecessary for the Oracle8i database, which does its own caching via the SGA. The goal of reducing the file cache allocation is to provide more actual memory for the Oracle8i application. To reduce the file cache, perform the following steps.

1. Choose the Network option from the Control Panel under the Settings menu.
2. Click the Services tab.
3. Select the Server service and click on the Properties option.
4. Select the button for the network applications configuration and click OK.
5. Click OK on the Network dialog box.
6. Exit the Control Panel.

In order for the new setting to take effect, you must reboot your system.

Disable Unnecessary Services

To retrieve additional memory for the Oracle8i database, you can disable services that are not needed for core operating system functionality or Oracle8i. Remember that the recommendation for unnecessary services to disable is based on the assumption that the Windows NT box is not being used for any other functions besides Oracle8i. If this is the case, you can disable the following services: Plug and Play, Remote Access Autodial Manager, Remote Access Connection Manager, Remote Access Server, and Telephony Service. To disable these services, perform the following steps:

1. Choose the Services option from the Control Panel under the Settings menu.
2. Select each of the above-listed options in turn and select Startup.
3. Under Startup Type, select Disable and then click OK.
4. After resetting each of the options, click Close to exit the Control Panel.

Remove unused network protocols and reset bind order

Since not all of the protocols that may have been configured or allocated on the Windows NT machine are being used, processing time that could be used by critical protocols may be wasted. Therefore, you want to remove any network protocols that are not being used by the backend server. To remove unused network protocols, identify the protocol(s) that you are using and then follow these steps.

1. Choose the Network option of the Control Panel under the Settings menu.
2. Click on the Protocols tab.
3. Select each unneeded protocol in turn and click the Remove option.
4. Click Yes to confirm your choice.
5. Once you have eliminated all unneeded protocols, click OK to exit.

If you are using more than one protocol, you can reset the network protocol bind order so that the protocol that is used most often by the Oracle8i database is given the highest priority. To reset the bind order, perform the following steps.

1. Choose the Network option of the Control Panel under the Settings menu.
2. Click the Bindings tab.
3. Select All Services from the Show Bindings for display area.
4. Double click on the Server list to expand it.
5. Select the protocol that is used most by Oracle8i (if it is not at the top of the list already) and click on Move Up until it is at the top of the list. You will want to do this for each network interface card that is installed on your system.
6. Click OK and exit the Control Panel.

You will need to reboot your machine to enable the new settings to take effect.

Optimize for network throughput

By default, the setting for local processes is geared to enable resources to be more available and usable by applications. To improve performance, you can reset the default from "Optimize for file sharing" to "Optimize for network throughput." By modifying this setting, you can see between a 5 – 10% improvement in the Oracle instance. If you have not installed the server version for NT, this option will not be available to change and you'll get a message "Cannot configure the software component." To reset the value:

1. Choose the Network option of the Control Panel under the Settings menu.
2. Click on the Services tab then select the Server option.
3. Click on the Properties option.
4. Select the "Maximize throughput for network applications" option.

Reboot the machine to enable the new settings to take effect.

Other available configuration options

Although the actions listed in the preceding sections will yield the most resource return for your backend server, there are other actions that you can consider taking to help provide some (though not as much) benefit.

Periodically, Microsoft releases service packs with patches to provide bug fixes and product enhancements. Before applying any new service packs to your system, be sure to check with Oracle Support to ensure that the new service pack has been certified with your current hardware configuration. If it has been approved and no major bugs have been reported within the service pack, applying the pack may help improve your system's performance.

If you are using the Windows NT system as a backend server, you can close unnecessary foreground applications from the startup folder. Look in the Winnt\Profiles\All Users\Start Menu\Programs\Startup directory for applications that are unnecessary. Programs like the FindFast indexing and MS Office Shortcut Toolbar can be removed.

Complex screen savers can quickly saturate the CPU. If a screen saver must be used for long-running processes, use a blank or limited text screen saver to reduce the CPU impact. By eliminating programs and features that are unnecessary for Oracle, you can improve the performance of any database or database networking on the server.

Tuning the Database

Because the approach to tuning the database is not operating system specific, I am going to just briefly cover a nice way to help determine the major cause of contention in your database. The approach will hold true for any Oracle8 database on any platform. That's actually good news since it means that you can use any of the knowledge you've gained on other platforms to tune the NT system or knowledge from tuning your NT database on other systems as well.

A great place to begin investigating where problems lie in your system’s performance is by gathering information from the views V$SYSTEM_EVENT, V$SESSION_EVENT, and V$SESSION_WAIT. Each view gives you a different insight into whether you have performance problems and where they really are. The V$SYSTEM_EVENT view gives you a high-level perspective of the database wait events, while the V$SESSION_EVENT view gives you a more detailed perspective on a session-by-session level. Keep in mind, though, that the V$SESSION_EVENT view will give you statistics on a session basis only while that session is active. Sessions are given a session ID number for the period of time that they are connected to the database. If a session disconnects, a new session may be assigned the old session's ID. Thus, you must look at V$SESSION_EVENT interactively to gather meaningful information.

A script that you can use to look at the system wait events in order of impact is

```
col Event format a37 heading 'Event'
col Total_Waits format 99999999 heading 'Total|Waits'
col Time_Waited format 9999999999 heading 'Time Waitd|In Hndrds'
col Total_timeouts format 999999 heading 'Timeout'
col Average_Wait heading 'Average|Time' format 999999.999
set pages 999
select *
  from V$SYSTEM_EVENT
 where Time_Waited > 0
   and Event not in('pmon timer','smon timer','rdbms ipc message',
   'parallel dequeue wait','virtual circuit','SQL*Net message from client',
   'client message','Null event','dispatcher timer')
 order by Time_Waited desc
/
```

The values that are used in the "not in" clause are ones that can be ignored when trying to identify where contention is highest on your system.

The major general areas you will look at for database tuning are:

The buffer hit ratio a measure of the number of times that a requested data block is found in the RAM memory of the SGA. *In general, the larger the database buffer, the higher the buffer hit ratio*.

Redo log space requests the number of times that Oracle had to flush the redo log buffer because of high activity. *The higher the value of the log_buffer init.ora parameter, the less the number of redo log space requests.*

DB file read waits the number of times that Oracle was required to wait for a physical I/O. This value is stored in the V$SESSION_WAIT view.

Buffer busy waits the number of times that a transaction had to wait to access a data block that was already in the data buffer. This is generally caused by excessive demand for the same row, such as the segment header for a table when parallel tasks are inserting data rows into the same table.

Table fetch continued row counts the number of times that Oracle encountered a row that was fragmented onto a successive data block. This occurs when rows with VARCHAR columns expand as a result of SQL UPDATE activity. Continued row fetches will also occur when very large rows are stored in tables with a small db_block_size, such that the row length is greater than the block size.

Disk Sorts counts the number of times that Oracle must go to the temporary tablespace to perform a large sort. *In general, the larger the sort_area_size, the fewer disks sorts.*

Library cache hit ratio the number of times that a library cache member (PL/SQL block or SQL statement) was found in the RAM memory of the SGA shared pool. *In general, the higher the value of the shared_pool init.ora parameter, the better the library cache hit ratio.*

Bibliography

1. "Oracle8i DBA Handbook", by Kevin Loney and Marlene Theriault, Chapter 17, Oracle Press - Osborne McGraw-Hill, 1999.
2. "Oracle DBA 101", by Marlene Theriault, et. al, Oracle Press – Osborne McGraw Hill, 1999.

About the Author

Marlene Theriault has been an Oracle DBA for over 17 years working with Oracle products (since version 2.0 of the Oracle RDBMS). She is co-author **Oracle Security** with William Heney, O'Reilly and Associates, 1998; **The Oracle8i DBA Handbook** with Kevin Loney, Oracle Press (Osborne McGraw-Hill), 1999; and **Oracle DBA 101** with Rachel Carmichael and James Viscusi, Oracle Press (Osborne McGraw-Hill), 1999. Ms Theriault received the "Best User Presentation" award at EOUG’99 for her presentation at EOUG’98, an "Outstanding Speaker" award at ECO'98, and “Distinguished Speaker” awards for presentations at both ECO’95 and ECO’96. She is chair of the MAOP DBA SIG and authors an “Ask The DBA” column for the MAOP newsletter. Marlene can be reached at: mtheriault@mindspring.com.