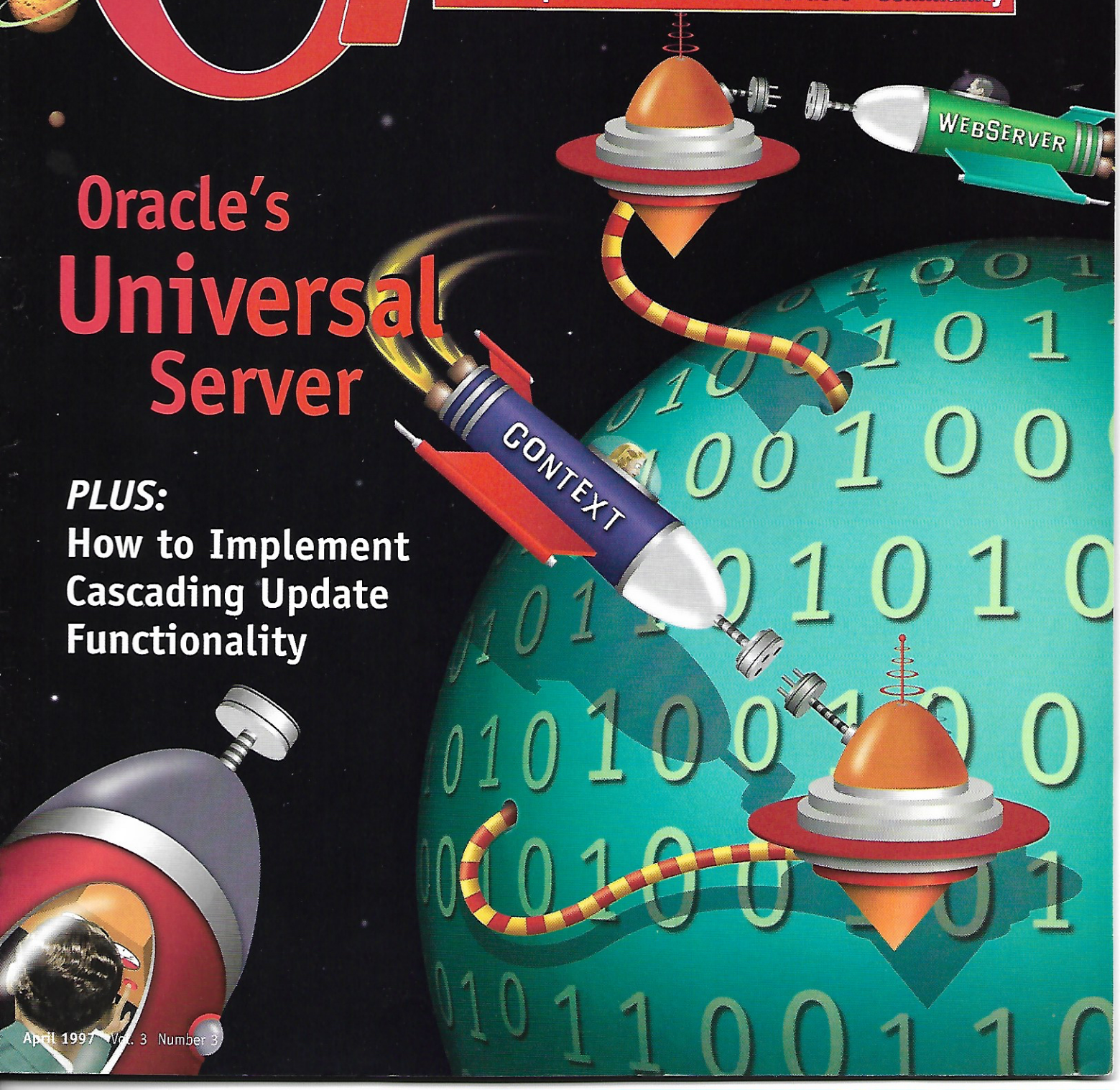
See What's Inside Larry's Brain This Month! page 48

The Magazine Formerly Known as *OTJ*

Oreview

The Independent Voice of the Oracle™ Community



Oracle's Universal Server

PLUS:
How to Implement Cascading Update Functionality

April 1997 Vol. 3 Number 3

www.oreview.com

Oreview

The Independent Voice of the Oracle Community

APRIL 1997 VOL. 3 NUMBER 3

Cover Illustration by John Ueland

editor'snote

By Kathleen O'Connor

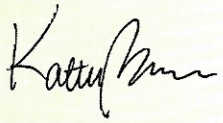
Technically Speaking

Welcome to the latest issue of *OReview*. For those of you who thought that our new redesign was going to translate to a less-technical magazine, rest assured that it does not. Just take a look at the table of contents and you'll see what I mean. This issue is probably one of the most technical that we've ever produced. It's full of down-and-dirty techniques that will enable you to get the most from your Oracle system.

You'll also notice when looking through the magazine that we've taken a lot of the code listings that normally appear in the magazine and put them on our Web site (www.oreview.com). We've done this because while we're trying to provide you with the maximum amount of technical information, we are limited by space. We have only so many pages of editorial content in each issue, and unfortunately code listings sometimes run several pages in length. To provide you with at least three or four beefy technical articles every month, we had to make a decision: Run fewer articles with the code listings printed in the magazine, or run more articles and put the listings on the Web site. If you don't have access to the Web, give me a call at 415.655.4243 and I'll send you a copy via email, fax, or snailmail.

Next month, look for our cover story on creating, implementing, and enforcing a security policy for your Oracle environment. We'll also have an article on how to migrate your applications from one database to another, the third and final part of Roger Snowden's application tuning series, an introduction to Oracle's WebServer and the PL/SQL agent, and a review of Platinum Technology's Plan Analyzer for Oracle. In addition, our May issue will feature some excerpts from an interview with Jerry Held (Oracle's senior VP of Server Technologies) and Ken Jacobs (VP of Product Strategy, Server Technologies, and Oracle8 Program Manager), who answer some questions about Oracle8. And, speaking of Oracle8, which shipped in second-beta mode on March 3: The June issue is currently scheduled to be data warehousing-focused, but may instead be Oracle8-focused, if Oracle8 ships for general availability in June.

On a sad note, I'd like to say good luck to the folks at *Oracle Informant* magazine, who decided to shut down operations (their April issue is their last). As one of the only remaining Oracle-specific magazines that is independent of Oracle Corp., we appreciate your continued interest and comments. You can reach us at oreview@mfi.com.



OReview
The Independent Voice of the Oracle Community

Editor in Chief ***Kathleen O'Connor***
(koconnor@mfi.com or 1-415-655-4243)
Technical Editor ***Steve Bobrowski***
Executive Managing Editor ***Annalisa Chamberlain***
Group Copy Editor ***Nadya Birnholz***
Contributing Editors
Suresh Aiyer, Michael R. Ault, Paul Dalberth, Ron Fleetwood, Steven Feuerstein, Steve Roti

Art Director ***Lynn LaRocca***
Illustrator ***Margaret Anderson***
Assistant Production Editor ***Michelle Yandrich***

Publisher ***HP Newquist***
Group Publisher ***Philip Chapnick***
Vice President/MIS-DP Division ***Steve Schneiderman***
National Advertising Sales Manager ***Jody Homelsky***
1-415-655-4271
Marketplace ***Michelle Wagner***
1-212-615-2814
Advertising Coordinator ***Stephanie DeBattista***
Reprints and Ancillary Products ***Gae Reck***
1-415-655-4245
Vice President Production ***Andrew A. Mickus***
Circulation Marketing Manager ***Lisa Eversole***
Circulation Assistant ***Jeff Laber***

Miller Freeman
A United News & Media publication

President/CEO ***Marshall W. Freeman***
Executive Vice President/COO ***Donald A. Pazour***
Senior Vice President/CFO ***Warren "Andy" Ambrose***
Senior Vice Presidents
H. Ted Bahr, Darrell Denny, David Nussbaum, Galen Poss, Wini D. Ragus, Regina Starr Ridley
Vice President/Circulation ***Jerry Okabe***

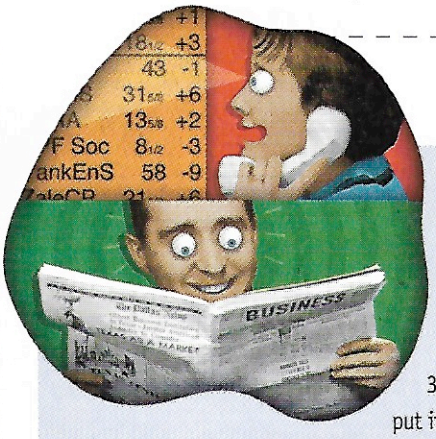
PRINTED IN THE USA

OReview: **The Independent Voice of the Oracle Community** (ISSN # 1084-7413) is published monthly except for combined January/February issue by Miller Freeman, Inc., 411 Borel Ave., Ste. 100, San Mateo, CA 94402, 415-358-9500. Please direct all advertising and editorial questions to this address.

Application to mail at Periodicals Postage Rate pending at San Mateo, CA and additional mailing offices. Subscriptions: For subscription inquiries, please call 800-250-2429 (within U.S.) or 847-647-5928; or write to: *OReview*, P.O. Box 1181, Skokie, IL, 60076-8181. Subscriptions are available at no charge to qualified persons.* Within the U.S. Paid subscriptions are available at the following rates: U.S – $29.95 for one year (eleven issues), Canada/Mexico –add $10.00 for postage, all other countries – add $40.05 for air mail postage. International subscriptions must be prepaid in U.S. dollars on a U.S. bank. Postmaster – Send address changes to: *OReview*, P.O. Box 1181, Skokie, IL, 60076-8181. *Publisher reserves the right to determine qualification.

businessnews April 1997

By Kathleen O'Connor



3Com Plans to Buy U.S. Robotics

In the largest acquisition ever involving a Silicon Valley company, 3Com, the company that put its name on Candlestick Park, has agreed to buy U.S. Robotics, the leading maker of modems. This $6.6 billion deal would establish a single source for consumers and large corporations.

3Com makes hardware that connects personal computers; with revenues of $3 billion, it's the world's second-largest networking company, behind San Jose-based Cisco Systems. U.S. Robotics, which recently released a supercharged modem that runs at twice the speed of other models, is the world's largest maker of modems, with $2 billion in annual sales. The marriage calls for 3Com chief executive Eric Benhamou to be chairman and chief executive officer of the combined company, which will keep the name 3Com. The 3Com-U.S. Robotics merger is scheduled to close this summer.

If this deal goes through, it will make 3Com a stronger competitor to Cisco Systems. In fact, many people are saying the 3Com-U.S. Robotics deal is a countermove to Cisco's recent buying binge, which included a $4.5 billion merger with Stratacom last April.

Pixar/Disney Deal

Steve Jobs' *other* company, Pixar, just made a sweet deal with Disney. Pixar Animation Studios recently announced a decade-long pact with Walt Disney Co., its partner in the 1995 blockbuster film "Toy Story." Disney and Pixar have agreed to codevelop and jointly finance five more animated films under the Disney-Pixar name and share equally in the profits from the films and related follow-on products such as videos, sequels, games, and merchandising (which itself is a billion-dollar market). This deal will give Pixar a heftier share of profits of future jointly developed films.

Whereas "Toy Story" was marketed and known as a Disney film, the five new features will all be branded as the properties of both Pixar and Disney. The same is true for any businesses flowing from the movies, so that Pixar's brand name will also appear on advertising, toys, videos, and anything else that might be produced. Disney also has an option to acquire up to five percent of Pixar's stock. With the announcement, Pixar stock rose almost 50 percent.

AT&T Goes Wireless → → → → → → → → → → → → → → →

AT&T released details of a new "revolutionary" fixed wireless technology that would enable it to bypass the regional Bell operating companies' (RBOCs) phone networks and offer local service direct to its business and home customers. The new technology is the result of a secret, three-year research effort. It uses the 10-GHz radio band to provide connectivity between the user and the local exchange, eliminating the need for AT&T to install its own local loop network or to interconnect with the RBOCs' networks.

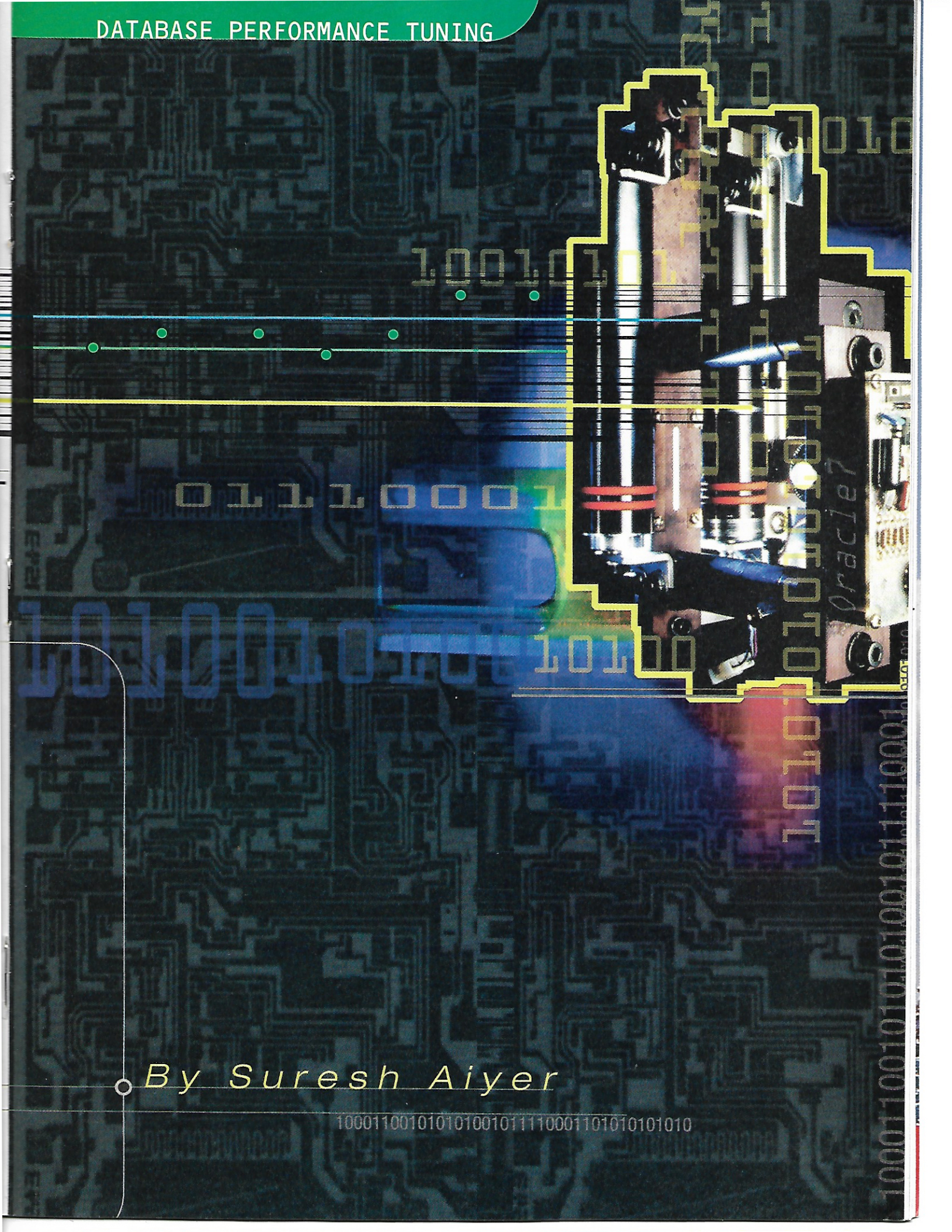
Although AT&T is initially positioning the technology to provide access for residential customers, the technology is equally as suitable for businesses, especially remote offices. Over the past several years, AT&T has been acquiring licenses that give it rights to this part of the radio spectrum in 93 percent of all U.S. households. The general assumption was that it wanted to provide cellular service; in reality, AT&T had plans to take over the home and business market.

The technology will be tested in Chicago later this year and, if it works well, will be rolled out to other parts of the country next year. The initial release will provide each customer with two phone lines and 128Kbps Internet access, but the company has already tested Internet access at 256Kbps and 384Kbps. With AT&T's wireless technology, customers won't have to scrap their existing phones or house wiring. Instead, AT&T will install a small radio unit, the size of a medium pizza box, on the outside of their houses. It will communicate via an antenna with a neighborhood base station that connects to AT&T's local and long-distance network. Each antenna will be able to support as many as 2000 users, and although there is currently a maximum distance of about 3000 feet between the antenna and the transceiver, AT&T is working to extend this distance to about 9000 feet. Some analysts were cautious about the announcement, however. The big questions are what will it cost and what's its capacity. Bob Kelly, a spokesman for Sprint PCS, a wireless cellular service, said his company has no plans to follow AT&T's lead.

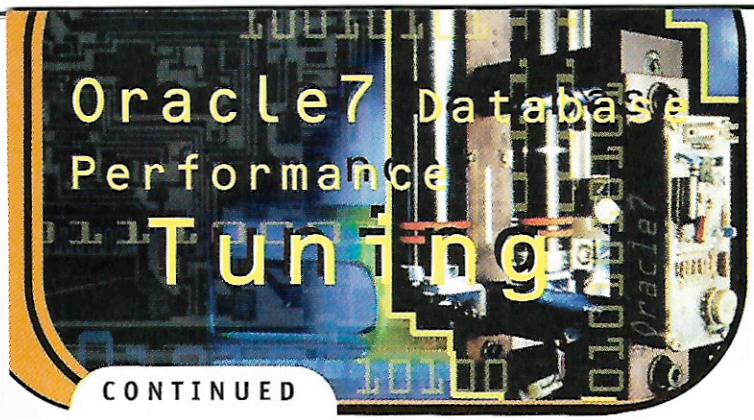
Oracle7 Database Performance Tuning:

Using the ULTBSTAT & UTLESTAT Scripts

This article shows you how to use the statistics in the report generated by the UTLBSTAT and UTLESTAT scripts to fine-tune your database for optimal performance.



By Suresh Aiyer



Successful Oracle7 database performance tuning involves the monitoring of numerous database parameters. The UTLBSTAT and UTLESTAT scripts provided by Oracle can assist in this effort by summarizing the operating state of your database in a single report. You can then use this report for fine-tuning your database performance and for preventive maintenance of your Oracle7 databases. These two scripts provide information about database memory objects, including library cache, dictionary cache, latch usage, file I/O, and rollback statistics. In this article, I show you how to interpret the report generated by UTLBSTAT and UTLESTAT scripts, and I reveal ways in which you can use these statistics to improve your database performance.

Set Up

Oracle Server version 7.3.2 was used for the purpose of the article. The UTLBSTAT and UTLESTAT scripts are located in the ORACLE_HOME/rdbms/admin directory. The first script, UTLBSTAT.SQL, creates a set of tables and views and populates them with the statistics in the database at that time. The object names contain the word "begin" — for example, STATS$BEGIN_FILE. The second script, UTLESTAT.SQL, which is run at a later time, creates objects whose name contains the word "end" — for example, STATS$END_STATS. It populates these objects with ending database performance statistics and generates a report that contains the changes in the statistics during the interval between the run times for the UTLBSTAT.SQL and UTLESTAT.SQL scripts. Both of the scripts should be run under USERID SYS from SQL*Plus, Server Manager, or an Enterprise Manager SQL Worksheet. You can also run these scripts after connecting as INTERNAL in Server Manager.

You should set the Oracle7 initialization parameter TIMED_STATISTICS to TRUE before you run these scripts, and you should run the scripts only after the database has been running for a period of time. If you run them immediately after the database startup, the buffer cache will not be loaded with representative application data, and the statistics generated will likely not be valid for database performance analysis. Also, if the database is shut down in the middle of the execution of these scripts, the statistics are no longer valid. Statistics having negative values indicate that the database has been shut down and restarted.

When the scripts should be run depends on the type of application being run. For example, if you run several batch jobs every evening, execute UTLBSTAT before the batch jobs start, and execute UTLESTAT after the batch jobs complete. In another scenario, if users of your system use an application from 8 a.m. to 8 p.m., you may run the UTLBSTAT at 8 a.m. and UTLESTAT at 8 p.m. You can automate the UTLBSTAT and UTLESTAT runs using operating system schedulers. The scripts generate the report in a file named report.txt. This file is located in the $ORACLE_HOME/rdbms/admin directory.

Both UTLBSTAT and UTLESTAT scripts create temporary objects in the SYSTEM tablespace. Frequent runs of these scripts may fragment the SYSTEM tablespace. To avoid such fragmentation, you can copy these scripts and modify them so that they create their temporary objects in a tablespace other than the SYSTEM tablespace. Another useful change to the scripts would be to include database links in the FROM clause of the SELECT statements to enable the scripts to run in remote databases and help gather remote database performance statistics.

Interpreting the Report Output

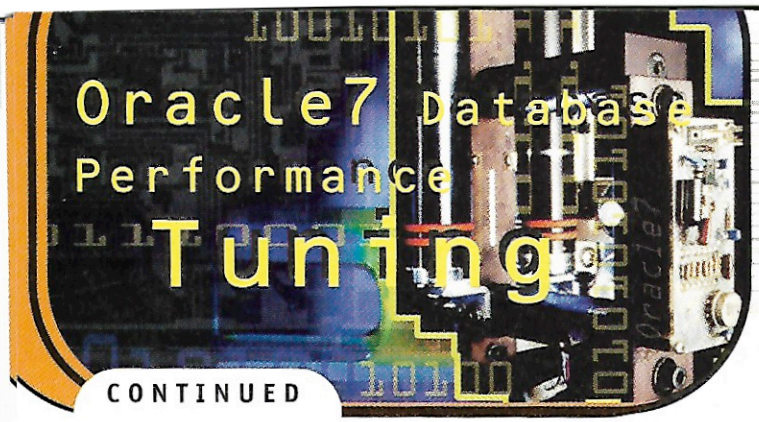
The statistics generated by these scripts are classified in the following sections: library cache statistics, overall system statistics, system-wide wait events, latch statistics, rollback statistics, instance initialization parameters (for reference), dictionary cache statistics, and file and tablespace I/O statistics. The date and time when the scripts are run are also included in the report. Some of the statistics are grouped at a per-transaction level, per-logon level, and per-second level. At the end, the report shows the Oracle7 Server version information. The following sections show you how to interpret these statistics to achieve better database performance. The statistic itself is shown in italics; the report generated by the scripts is shown in Listing 1 (located on the Web at www.oreview.com).

Library Cache Statistics

The library cache consists of shared SQL and PL/SQL areas. These statistics indicate if shared SQL statements are being reparsed because of insufficient memory allocated to the library cache. The *pins* column indicates the number of times an item was executed. The *reloads* column shows the number of misses. The ratio of *reloads* to *pins* is the percentage of executions that resulted in reparsing. If the *get hit ratio* or *pin hit ratio* is less than 90 percent, or if the ratio of *reloads* to *pins* is more than one percent, you should increase the memory allocated to the library cache by increasing the initialization parameter SHARED_POOL_SIZE.

Overall Statistics

The *DBWR checkpoints* indicates the number of checkpoint messages that were sent to database writer during the course of the report. This does not mean that so many checkpoints were performed, because if a second checkpoint is issued while the first is active, the first one stops and the second checkpoint starts. If checkpoints occur frequently, they can degrade the database's per-



CONTINUED

formance. During the checkpoints, data blocks are written to disk, causing additional I/O. You can lower the frequency of the checkpoints by making the initialization parameter LOG_CHECKPOINT_INTERVAL larger than the redo log file size. Thus checkpoints are performed only during log switch. If the number of checkpoints is reduced, the performance of normal database operations improves, although recovery after an Oracle7 instance failure takes longer. If the database initialization parameter CHECKPOINT_PROCESS is set to TRUE, then a checkpoint process would update the headers of data files during checkpoints. This would relieve the log writer of the responsibility to update data file headers during checkpoints and can thus improve the performance of transaction logging in an OLTP environment.

Cluster key scan block gets and *cluster key scans* provide information about clusters in the database. *Cluster key scan block gets* is the number of cluster blocks accessed; *cluster key scans* is the number of scans processed on cluster blocks. If the ratio of *cluster key scan block gets* to *cluster key scans* is greater than one, the rows for one cluster key are stored in multiple data blocks, and the cluster needs to be analyzed for row chaining. When a cluster is created, the SIZE parameter determines the number of cluster keys per block; the default is one. If the SIZE parameter is not specified correctly, the rows for one cluster key may not fit properly in the data block. Therefore, it is important that you check the SIZE parameter of the cluster.

A dirty buffer is a data block buffer that has been modified but not written permanently to the database. *Dirty buffer inspected* is the number of times a foreground process encounters a dirty buffer that has aged out via the least recently used (LRU) queue while the process is looking for the buffer for reuse. This number should be zero if the database writer is keeping up with the foreground processes. If this number is high, check the number of database writers. Normally the number of database writers should be twice the number of disk drives on the system. If asynchronous I/O is available at your operating system level, it is better to use that rather than multiple database writers, because it requires less overhead and less processes. When using asynchronous I/O, Oracle performs parallel writes without using multiple database writers. Asynchronous I/O can be set by setting initialization parameter ASYNC_WRITE. This parameter is operating-system-dependent.

Free buffer requested is the total number of free buffers needed in order to create or load a block. *Free buffer inspected* is the number of buffers skipped in the buffer cache by a user process in order to find a free buffer. If this value is high, it means that there are too many modified blocks, and you should increase the buffer cache. *Free buffer waits* (in the system-wide wait event section) is the number of times that processes needed a free buffer and one was not available. *Write complete waits* (in the system-wide wait event section) is the number of times a process waited for the database writer to write a current block before making a change to a buffer. If the value of *free buffer waits* or *write complete waits* is high or showing growth, consider increasing the initialization parameter DB_BLOCK_WRITE_BATCH, which specifies the number of blocks that should be written to the disk at one time.

Consistent changes indicates the number of times a database block has rollback entries applied to perform a consistent read on the block. Applications that produce a great deal of consistent changes can consume a lot of resources. *Consistent gets* indicates the number of times a consistent read was requested for a database block. *DB block changes* indicates the total number of changes made to all of the database blocks in the SGA that were a part of an UPDATE or DELETE operation. This statistic shows, on a per-transaction basis, the rate at which database buffers are being modified. It is a rough indication of the total database work being performed. *DB block gets* is the number of blocks accessed in the buffer by INSERT, UPDATE, DELETE, or SELECT FOR UPDATE statements. The sum of the values of *consistent gets* and *db block gets* is known as the number of logical reads.

The *physical reads* statistic stores the number of I/O requests to the operating system to retrieve a database block from the disk subsystem. It is the number of requests for a block that resulted in physical I/O. This is a buffer cache miss. *Physical writes* is the number of I/O requests to the operating system to write a database block to the disk subsystem. The bulk of the writes are performed by either the DBWR or the LGWR server background process.

The ratio between logical reads and *physical reads* should be greater than 90 percent:

Ratio = logical reads\(logical reads + physical reads)

If this ratio is less than 90 percent, too many buffer cache misses are occuring. To correct the problem, consider increasing the buffer cache by increasing the value of the initialization parameter DB_BLOCK_BUFFERS.

Opened cursors cumulative is the total number of cursors that were opened during the execution of the UTLBSTAT and UTLESTAT scripts. A cursor is opened for each SQL statement parsed into a context area. If the cursors are reused, the performance improves, because there is no need to reparse SQL statements. If *opened cursors cumulative* is high, check for applications opening many cursors. An application should close the cursor after the

SQL statement completes. A high *opened cursors cumulative ratio* can also indicate poorly designed application statements — for example, statements that do not use bind variables. Consequently, shared SQL is not possible because the server can only reuse shared SQL statements that are identical in syntax. For more details about application tuning, refer to Roger Snowden's article "Application Tuning, Part 1" in the January/February 1997 issue (page 10).

Recursive calls indicate data dictionary cache misses and segment extensions. The Oracle database maintains tables used for internal processing. When the database needs to make a change to these tables, the database internally generates a SQL statement. These internal SQL statements also generate recursive calls. In general, *recursive calls* should be less than four per process, and the ratio of *recursive calls* to *user calls* should be less than 10 percent. If the value of this statistic is high, you should tune the data dictionary cache and re-create segments such as tables, indexes, temporary segments, and rollback segments with a storage clause that has a few large extents. Data dictionary cache misses can be reduced by increasing shared pool area. If it is not possible to recreate database objects, you should identify the objects that are extending frequently and increase the value of their `NEXT` storage parameters to ensure that the next extent is created large, which will prevent subsequent extensions. It is also possible that for rollback segments, `OPTIMAL` parameter might be set wrong such that the rollback segment is constantly growing and shrinking, thus causing the extensions. Set the `OPTIMAL` parameter for the rollback segment so that it does not shrink often.

The *redo entries* statistic increments each time an Oracle user process copies redo entries into the redo log buffer. The server creates space by performing a log switch. If the server's log files are small compared to the size of the SGA or the commit rate of the workload, the server may have problems logging ongoing transactions. When the server performs a log switch, Oracle must ensure that it writes to disk all committed dirty buffers before switching to a new log file. If the server has a large SGA full of dirty buffers and small redo log files, a log switch must wait for the database writer to write dirty buffers to the disk before continuing. *Redo sync writes* is the number of times that the server writes redo log changes from the log buffer to disk.

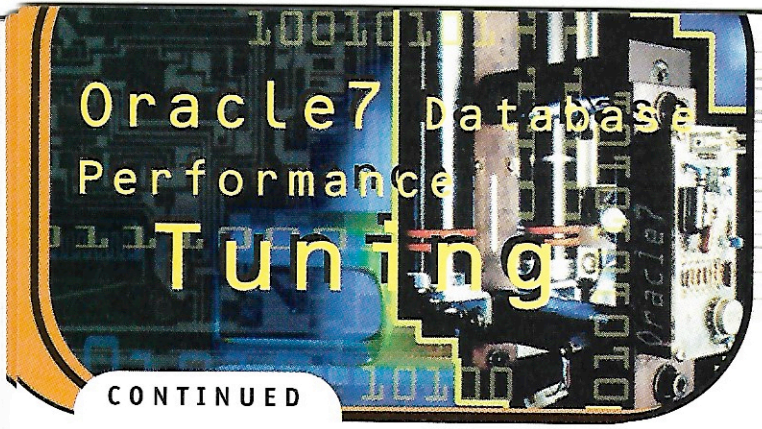
Redo small copies is the number of redo entries that are smaller than the initialization parameter `LOG_SMALL_ENTRY_MAX_SIZE`. Oracle writes redo entries smaller than `LOG_SMALL_ENTRY_MAX_SIZE` using the redo allocation latch; the server writes larger entries using copy latches. On single CPU computers, there is no copy latch because only one process can be active at one time. In this case, all redo entries are copied on the redo allocation latch, regardless of size. On multiple CPU computers, the redo log buffer can have multiple redo copy latches. In this case, if the redo log size is more than `LOG_SMALL_ENTRY_MAX_SIZE`, a copy latch is obtained and redo log data is copied into the redo log buffer using the copy latch. Multiple redo log copy latches enable multiple processes to copy entries to the redo log buffer concurrently and hence result in better performance. To use more copy latches, decrease `LOG_SMALL_ENTRY_MAX_SIZE`. Set the initialization parameter `LOG_SIMULTANEOUS_COPIES` equal to the number of CPUs to achieve better copy latch concurrence. There is only one redo allocation latch per instance, and the number of redo copy latches is the same as the number of CPUs on the machine. Multiple processes can hold a redo copy latch. The hit ratio for *redo copy* and *redo allocation* latches should be more than 90 percent (refer to the Latch Statistics section of this article for these ratios).

Sorts (disks) is the number of times that the database created a temporary segment to perform sorting on disk. Oracle creates temporary segments when there is not enough room in memory to complete the sort. *Sorts (memory)* is the number of times a sort was executed in the memory. *Sort (rows)* is the total number of rows sorted. If *sorts (disk)* is high, increase the initialization parameter `SORT_AREA_SIZE`. Also, if possible, modify the application to perform fewer sorts. SQL statements with a `GROUP BY` or `ORDER BY` clause, `DISTINCT` operator, `UNION/INTERSECT/MINUS` statements, sort-merge joins, and index-creation statements result in sorting. By default, a tablespace is permanent in nature. The objects created in such a tablespace remain in the tablespace unless deleted by a user. Oracle7.3 introduced a tablespace called `TEMPORARY`, which can only have temporary objects in it. Objects in the `TEMPORARY` tablespace are periodically removed by the Oracle Server.

Table fetch by rowid should be high. This statistic includes rows that were accessed using an index and rows that were accessed using the statement `WHERE ROWID = "XXXXXXXX.XXXX.XXXX".ROWID` is the fastest way to access data and should be used wherever possible. *Table fetch continued row* is the number of rows that are chained to another block. It indicates that additional I/O must be performed to access the entire row. The *table fetch continued row* value should be lower than 10 percent of the *table fetch by rowid* statistic. If the *table fetch continued row* value is high, reduce the chaining of rows by re-creating the table with proper storage parameters or reinserting the chained rows. You cannot avoid row-chaining in a table with `LONG` columns. Chaining can also occur in update-intensive applications in which the storage parameter `PCTFREE` is not set correctly for a table's or index's data blocks. In such cases, you should modify the `PCTFREE` parameter for objects containing chained rows.

Table scan blocks gotten indicates the number of database blocks Oracle scanned in order to get data. During scanning operations, each row is retrieved sequentially, and every block encountered during the scan increments this statistic. Compare this statistic to *consistent gets* to get a feeling for how much of the consistent read activity results from scanning. The *table scan rows gotten* statistic shows the number of database rows processed to get the data.

Table scans (long tables) is the number of full table scans performed on tables with more than four database blocks. If the



number of full table scans is much greater than zero on a per-transaction basis, you should tune the application to use Oracle indexes effectively. If more than 20 percent of the rows from a table are returned, indexes should be used on long tables. *Table scans (short tables)* is the number of full table scans performed on tables with fewer than four database blocks. On short tables, full table scans result in better performance than index scans. The total number of full table scans that occur during the execution of the `UTLBSTAT` and `UTLESTAT` scripts is the sum of *table scans (long tables)* and *table scans (short tables)*.

User calls is the number of times a call is made to the server. If possible, you should reduce the number of calls to the server. You can use array processing in applications to reduce the number of calls to the server. *Parse count* shows the number of times a SQL statement was parsed. The *parse count* statistic counts the number of times the user called "parse" function from either the OCI or the Oracle precompiler. You can find the number of calls to the server per parse by dividing *parse count* by *user calls*.

User commits is the number of times that users have performed a `COMMIT`. When a user commits a transaction, the redo information must be written to disk. This redo information reflects the changes made to database blocks. Transaction commits represent the closest thing to a user transaction rate. *User rollbacks* shows the number of times that users have explicitly performed a transaction rollback and when the server has implicitly performed a transaction rollback because of some type of an error. *Write requests* indicates the number of times the database writer takes a batch of dirty buffers and writes them to disk.

System-Wide Wait Events

This section of the report lists various wait events specific to the Oracle7 Server. For each statistic, the total time taken by waits and the average time per wait are shown. *Free buffer waits* and *buffer busy waits* statistics do not appear in the report generated by Oracle Server version 7.3. However, they do appear in the report generated by previous versions. *Free buffer waits* stores the number of times a free buffer was requested in the SGA but none were available. Free buffers are buffers that are not being used by current transactions. If the SGA is full of dirty buffers and the database writer cannot write them to disk, then the value of *free buffer waits* will increase. Update-intensive applications that use small indexes or hash clusters may run the risk of having an entire SGA full of dirty buffers that a database writer cannot keep up with. In such cases, consider increasing the server's `DB_BLOCK_BUFFERS` initialization parameter.

A high *buffer busy waits* value indicates a predominantly I/O-bound application. In a single-instance mode, a high value indicates contention for database blocks. In an Oracle parallel server environment, this indicates pinging (competition between nodes for database blocks). The *buffer busy wait* ratio is *buffer busy waits* divided by logical reads (logical reads equals the sum of the values of *consistent gets* and *db block gets*). If this ratio is greater than four percent, you can discover the type of buffer waits with the following SQL statement:

```
SELECT class, SUM(counts) waits FROM v$waitstat
  WHERE class IN ("undo header", "undo block",
   "data block")
  GROUP BY class;
```

If waits are high for "undo header" or "undo block," there are waits for rollback segments, so you should increase the number of rollback segments. If waits are high for "data block," increase the number of freelists for the tables that are involved in many `INSERT` operations.

This section of the report also lists SQL*Net messages received from clients, SQL*Net messages sent to clients, and SQL*Net activities using database links to remote databases.

Latch Statistics

The hit ratio for all latches should be more than 90 percent. If the contention for *cache buffer lru* is high, increase the initialization parameter `DB_BLOCK_WRITE_BATCH`. If the hit ratio for *enqueues* is low, increase the initialization parameter `ENQUEUE_RESOURCES`. If there is contention for the *redo allocation* latch, decrease the parameter `LOG_SMALL_ENTRY_SMALL_MAX_SIZE`. *Row cache object* latches occur when a process is updating an entry in the dictionary cache. *Session allocation* latches occur when a process allocates a new session. If the hit ratio is low for any of these parameters, increase the shared pool area. The *no wait hit ratio* column shows the percentage of *no wait* latch requests that were satisfied immediately. This ratio should be close to one; if it is not, increase `SHARED_POOL_AREA`.

Rollback Statistics

The *undo_segment* statistic identifies the rollback segment number to which the line refers. *Trans_tbl_gets* is the number of rollback segment header requests. *Trans_tbl_waits* is the number of rollback segment header requests that resulted in waits. *Undo_bytes_written* is the number of bytes written to the rollback segment. *Segment_size_bytes* is the size of the rollback segment in bytes; this column has only the ending value. *Xacts* is the current number of active transactions. *Shrinks* is the number of shrinks the rollback segment had to perform in order to stay within the `OPTIMAL` size. *Wraps* is

the number of times a rollback segment entry wrapped from one extent to another. Non-zero values of shrinks and wraps indicates that the rollback segment is expanding and shrinking to the optimal setting. If rollback segments are shrinking and wrapping too frequently, you should redesign the rollback segment to reflect the type of transactions being performed against the database. This would involve recreating the rollback segments with appropriate values for the `NEXT` and `OPTIMAL` storage parameters.

If the ratio of *trans_tbl_waits* to *trans_tbl_gets* is greater than five percent, add additional rollback segments in the database. In general, rollback segments should be equal in size and created with a large number of small extents. If you have large batch job transactions or large snapshot group refreshes, you might want to create a few larger rollback segments and assign them to these types of transactions.

Dictionary Cache

The dictionary cache needs to be tuned if the ratio of the number of *get_miss* to *get_req* is greater than 10 percent. You may need to increase the size of the shared pool with the parameter `SHARED_POOL_SIZE`. The *count* column shows the setting of that cache parameter in the database, and the *current_usage* column shows the number of current entries in that cache.

File I/O Statistics

This section of the report has I/O information at every database file and tablespace level. The information includes the number of physical reads from the data file (*reads*), the number of blocks read from the data file (*blks_read*), the time to read blocks (*read_time*), the number of physical writes to the data file (*writes*), the number of physical blocks written to the data file (*blks_wrt*), and the time to write blocks (*write_time*). File I/O should be spread evenly across multiple disk drives. In general, tables should be located on different disks from their associated indexes, large tables and indexes should be striped across several disks, active data files should be located on the highest throughput disks, and redo logs should be located on disks that do not contain database data files.

As seen from the report shown in Listing 1, the disk drive *data04* has the maximum physical I/O, which is much more than the other disk drives. If a particular disk drive has more physical I/O than other disk drives, you should spread data from that disk across other disk drives. Also, in the report, the tablespace *po_hdr* has substantially more physical reads than other tablespaces, which indicates that you should give more tuning attention to objects in this tablespace as well as to applications accessing these objects.

You can set the initialization parameter `DB_FILE_MULTI_BLOCK_READ_COUNT` to increase the number of blocks read during a single read. Increasing this parameter reduces I/O when full table scans are performed. The report also generates I/O distribution summed at the tablespace level.

Achieving Robust Performance

`UTLBSTAT` and `UTLESTAT` are a set of SQL scripts useful for capturing a snapshot of system-wide database performance statistics. The statistics generated from these scripts can help you optimize the performance of your Oracle7 databases. These two scripts can gather a snapshot of your database performance over a given period of time and help you fine-tune the numerous database parameters for optimal database performance. As DBAs become more overwhelmed with database performance gathering statistics, newer tools have automated this task to some extent. One such product is Oracle Expert, which is part of the Oracle Enterprise Manager Performance Pack. It automates collection, analysis, interpretation, and implementation of tuning data.

Suresh Aiyer *is a consultant based in Virginia. He specializes in Oracle development and database administration. You can reach him at 703.560.7005.*

APRIL

productnews

By Kathleen O'Connor

BRADMARK INC. announced **DBGeneral Server Manager for Oracle**, a GUI tool that displays the logical and physical structures of an Oracle tablespace and can predict space problems with trend analysis. Other features include user and role management, table reorganization, general database administration, and SQL script generation.

Server Manager version 2.0 is now shipping with native agents for Windows NT 3.51 or later, Windows 95, HP-UX, IBM AIX, and Sun Solaris. It will be available for DEC Unix and Open VMS platforms by the second quarter of 1997. A 30-day free evaluation is available upon request.

Bradmark Inc., 800.275.2723 or www.bradmark.com.

ORACLE CORP. has been busy these past few months. Following are just a few of the strategic partnerships, licensing agreements, and product endorsements and upgrades announced recently from Redwood Shores.

ORACLE reveals Oracle Applications Release 10.7. Release 10.7, announced for release in late February 1997, includes complete Year 2000 compliance, improved global features, and supply-chain management. With its planned announcement to **Web-enable all 30+ modules in its Applications product line by the second quarter of 1997**, Oracle will become the first major vendor to extend its modules to the Web. New modules include a Supply Chain Planning module that provides product configuration and improved scheduling capabilities. Oracle added a fully integrated Time and Attendance component to its human resource application, enabling users to automate time and attendance records and track hourly employees. It can also help eliminate overpayments, underpayments, or payments not in compliance with an organization's rules and policies. Version 10.7 will be the first Oracle suite to run on Windows NT.

ORACLE has partnered with IDENTIX INC. of Sunnyvale, Calif. to offer fingerprint recognition in its next version of its Oracle7 database. With the integration of Identix's TouchNetII, users will be able to place their finger on a glass plate embedded in a mouse-size unit that plugs into their PC. An optical scanning system reads the print and converts it to algorithms. The computer then matches the print to one on file before granting or denying access. On a gruesome note, the unit also verifies that the finger is alive by checking blood flow and other characteristics.

ORACLE is licensing object request broker (ORB) software from VISIGENIC SOFTWARE INC. in San Mateo, Calif. Oracle will resell the products as a part of its network computer architecture, and the license will also enable Oracle to integrate the Visigenic's ORB technology into its databases, tools, and applications, including its Web Application Server 3.0 (the Visigenic technology is scheduled to be offered with version 3.0 by mid-1997). The technology will be used to develop client software and application cartridges for NCA networks. Oracle licensed both the Java and C++ versions of VisiBroker, which will let developers write application cartridges in those languages, and its VisiBridge software, which links ActiveX clients to applications that support CORBA.

ORACLE is also licensing BORLAND INTERNATIONAL INC.'s C++Builder and Java-based JBuilder tools — including the source code, object code, and other general technology — for use in its Developer/2000 and Designer/2000 tools and Sedona object-oriented development environment. This licensing should boost Oracle's application tools line and ultimately its network computers. The Java-enabled version of Developer/2000 was scheduled to ship sometime in March 1997, and Borland's tools will be integrated into Designer/2000 and Sedona by the third quarter.

More from the When Pigs Fly department: **ORACLE is planning to introduce its Pentium-based network computers in Japan on April 15.** Japanese vendors were the first to sign up to build these machines, which will include 16-bit software and multilingual suppport.

Oracle Corp., 415.506.7000 or www.oracle.com.