

# nocoug

JOURNAL

Official Publication of the Northern California Oracle Users Group

VOL. 16, No. 4 · NOVEMBER, 2002

\$15



## NoCOUG Fall Conference in San Francisco

The NoCOUG Board is pleased to announce that our Fall Conference will take place on Thursday, November 21 at PG&E in downtown San Francisco. We are thrilled to have PG&E as a new meeting sponsor! Check out the back page for details on the great conference we have planned.

We recommend that you take public transit to the meeting if you can. There is limited parking, and the site is very close to BART and MUNI stops. ▲

### Mark Your Calendars for Upcoming Conferences!

We've already started planning some great NoCOUG Conferences for next year. Our first Conference of 2003 will take place at Oracle Corporation on Thursday, February 20. We have lined up some outstanding speakers, including Craig Shallahamer of OraPub, Inc. Shallahamer is a nationally recognized Oracle expert and author who is also a very popular speaker.

Our Spring Conference will take place on Thursday, May 15, 2003, at Lockheed Martin in Sunnyvale. Many thanks to the great NoCOUG supporters we have at Oracle Corporation and at Lockheed Martin! ▲

### TABLE OF CONTENTS

NoCOUG Fall Conference in SF.....	1
Mark Your Calendars.....	1
From the Editor.....	2
NoCOUG Board.....	2
User Groups.....	2
Publication and Submission Format.....	2
Advertising Rates.....	2
President's Message.....	3
Tech Tips.....	3
Board Member Spotlight.....	5
Resource Corner.....	6
The Expensive Parse Phase and the Cost of Using Synonyms.....	7
Treasurer's Report.....	13
Sponsorship Appreciation.....	13
All About Oracle's Touch-Count Data Block Buffer Algorithm.....	14
Tech Tips.....	17
Stand By Your Standby.....	19
Tech Tips.....	21
Uncommon Leaders: Managing the Blur of IT!!.....	23
NoCOUG Fall Conference Session Descriptions.....	27
NoCOUG Fall Conference Schedule.....	28
-ADVERTISERS-	
Quest Software.....	4
PerYourRequest.....	5
Informatica.....	6
Embarcadero Technologies.....	6
LECCOTECH.....	6
BMC Software.....	12
Database Specialists, Inc. ....	25

# Wrapping Up 2002

This edition of the *NoCOUG Journal* is the last issue of the year. I've really enjoyed my involvement with the NoCOUG board this past year. The board consists of a genuinely nice group of people who have the common goal of contributing something positive to the membership community. There is always lively discussion about how we'll achieve our goals, and I think we work very well together as a team. We also share some laughs along the way, starting with some agendas that account for having "a bagel moment." Yes, our president, Joel Rosingana, keeps his sense of humor . . . bagel in one hand, gavel in the other.

This is the end of my second year on the board, and I hope to stick around for a while longer. See you all next year.

—Lisa Loper, *NoCOUG Journal* Editor

## Other User Groups

### Local

NorCalOAUG – Northern California Oracle Applications Users Group

- **Contact:** Michael Capelle  
(650) 562-1167
- **Email:** capelle@tru-course.com
- **Website:** [www.norcaloaug.org](http://www.norcaloaug.org)

### Sacramento

SacOUG – The Sacramento Oracle User Group

- **Contact:** Ravi Verma  
(916) 705-3261
- **Email:** ravi.verma@telcommand.com
- **Website:** [www.sacoug.org](http://www.sacoug.org)

### International

IOUG – International Oracle Users Group

- **Website:** [www.ioug.org](http://www.ioug.org)

### U.S. Domestic

OAUG – Oracle Applications Users Group

- **Website:** [www.oaug.org](http://www.oaug.org)

ODTUG – Oracle Development Tools User Group

- **Website:** [www.odtug.com](http://www.odtug.com)

Canvassing calls by employment recruiters to local chapter contacts is strongly discouraged.

## NOCOUG BOARD

### President

Joel Rosingana, Independent Consultant  
joelros@pacbell.net

### Vice President

Roger Schrag, Database Specialists, Inc.  
rschrag@dbspecialists.com

### Treasurer/Secretary

Judy Lyman, Contra Cost County  
Public Works  
gooma@california.com

### Membership

Vacant Position

### Webmaster

Vadim Barilko, Independent Consultant  
vabarus@onebox.com

### Journal Editor

Lisa Loper, Database Specialists, Inc.  
lloper@dbspecialists.com

### Vendor Relations

Ganesh Sankar, Providian Financial  
bgs2k2@yahoo.com

### IOUG-A Representative and Past President

Vilin Roufchaie, Cingular Wireless  
vilin.roufchaie@cingular.com

### Members, At Large

Mike DeVito, Independent Consultant  
mike@devito.com

Hamid Minoui, Fritz Companies  
hamid.minoui@ups-scs.com

Darrin Swan, LECCOTECH  
darrin@leccotech.com

## Publication and Submission Format

The *NoCOUG Journal* is published four times a year by the Northern California Oracle Users Group approximately two weeks prior to the quarterly regional meetings. Please send your questions, feedback, and submissions to: Lisa Loper, NoCOUG Journal Editor, at [journal@nocoug.org](mailto:journal@nocoug.org).

The submission deadline for the upcoming February issue is January 1, 2003. Article submissions should be made in electronic format via email if possible. Word documents are preferred.

*NoCOUG does not warrant the NoCOUG Journal to be error-free.*

Copyright © 2002 by the Northern California Oracle Users Group. Permission to reproduce articles from this publication, in whole or in part, is given to other computer user groups for nonprofit use, with appropriate credit to the original author and the *Northern California Oracle Users Group Journal*. All other reproduction is strictly prohibited without written permission of the editor. Two copies of each reprint should be sent to the editor.

## ADVERTISING RATES

Contact: Nora Rosingana

325 Camaritas Way  
Danville, CA 94526  
Ph: (925) 820-1589

*The NoCOUG Journal is published quarterly.*

The rates are:

Size	Per Issue	Per Year
Quarter Page	\$100	\$320
Half Page	\$200	\$640
Full Page	\$400	\$1,280

*Personnel recruitment ads are not accepted.*



# How Time Flies!

by Joel Rosingana, President, NoCOUG

**H**ere it is, the fourth *Journal* of 2002 already. How time flies when you're having fun! Looking back over the year, we have had three powerhouse meetings. The year started with the meeting in February sponsored by Oracle Corporation. The membership always looks forward to the meeting at the Corporate Headquarters. Lockheed-Martin hosted another great meeting in May. Our August meeting, held in a new facility at ChevronTexaco, had a record attendance. Thanks to all our Hosts! Without you, NoCOUG would be much diminished.

This leads to a welcome to a new host—PG&E. Ray Liou, Senior DBA at PG&E heard my call for help at the last meeting. He went to work and got the permission and space necessary to host us at PG&E Corporate Headquarters in San Francisco. This was no easy task, especially on such short notice. Thank you, Ray! I would also like to thank the following people in Application Development and Support at PG&E who helped make this possible: April Mason, Team Lead; Bob Glynn III, Manager; and Lance Maeda, Director.

I'm looking forward to a San Francisco meeting, as I'm sure many of you are. We haven't been to the City in quite awhile. Let's show our appreciation by setting a new attendance record! Check this *Journal* and our website for more meeting details. Some of the advance information crossing my desk shows a great agenda in the making.

Each quarterly meeting requires a great deal of thought and planning. This culminates in an agenda. But, there is a larger agenda that we deal with each year. That is the long-range planning and staffing. We will be electing our 2003 officers at the December board meeting. We also spend considerable time at this meeting discussing upcoming issues. This

would be a great time for membership input. Keep in mind . . . the board exists for the membership. Your thoughts are very important. Stop a board member at the November meeting and express your ideas on NoCOUG's future direction. A follow-up note to board@nocoug.org would be helpful. This way we could have a list to work with at the board meeting. If you would like to attend that board meeting, let us know. The December meeting is held on a Saturday morning in Pleasanton.

I would like to take time in this



Joel Rosingana

last *Journal* of the year to say, "Thanks." I want to thank the dedicated board of directors for making my time as president a breeze. It's their commitment to a multitude of tasks that makes this all possible. I want to thank the membership for allowing me to serve as president. As I said at the beginning, it was fun! It was great to meet many of you throughout the year. As mentioned before, there will be a new slate of officers for 2003. A strong board of directors is one that gives everyone a chance to try new tasks. This adds depth of experience to the board. So, I expect you will see many of the same faces next year, but they may be taking on new and equally challenging positions. I hope we will also see some new faces. ▲

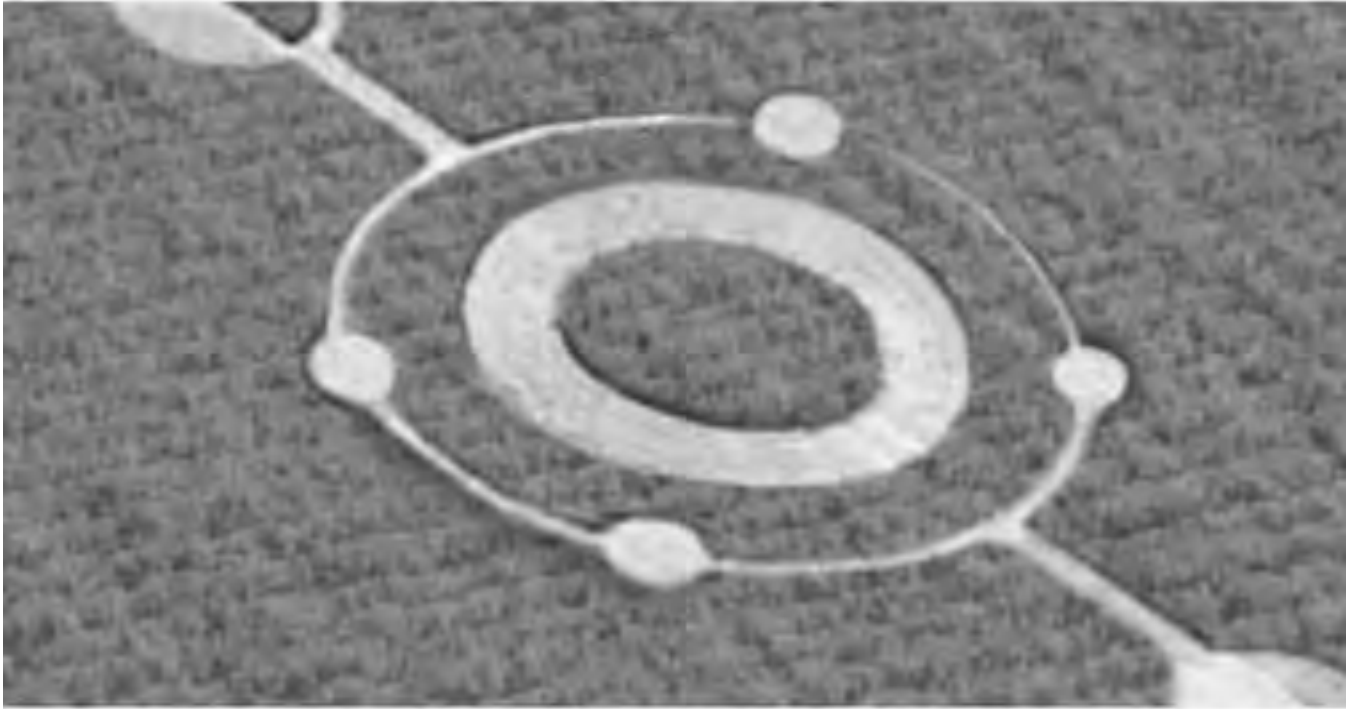


## TECH TIPS

### Oracle on Linux Resources

If you are using, or plan to use, Oracle on Linux, don't miss Mark Wagner's Oracle on Linux website at <http://www.oraclinux.org>. It's full of information and links on the subject, ranging from install documents and scripts to perl, PL/SQL, tuning, and backups. In addition, you can find a detailed guide to installing and configuring Oracle on the Linux platform by Roger Schrag of Database Specialists. You can view Roger's white paper at <http://www.dbspecialists.com/presentations.html#817onlinux>.

# With the Right Tools, Anything is Possible



## Quest Software Brings Higher Intelligence to Database Development

Smart people with good tools can mystify the masses and achieve the impossible. With Quest Software's market-leading tools for Oracle development, so can you. Quest Software offers the complete solution to design, develop, test and deploy code quickly and accurately, leaving you plenty of time to consider the possibilities.

The evidence is clear: more than 200,000 users worldwide believe Quest Software's development tools are the intelligent choice. Download your free trial versions from [www.quest.com](http://www.quest.com) and see for yourself.



[www.quest.com](http://www.quest.com) • 1.800.306.9329 (U.S.) • 1.949.754.8000 (outside U.S.)



# Meet the NoCOUG Webmaster!

Many of you may have noticed some great improvements to our NoCOUG website this year. That's because of our dedicated webmaster, Vadim Barilko. Vadim has been on the board for about a year. In addition to implementing small changes throughout the year, Vadim managed a site redesign that was launched in August. There's been lots of positive feedback about the site, including the new capability of renewing memberships online.

## Taking on a Challenge

"I enjoy a challenge," says Vadim. "I came to the board meeting and suggested some improvements to the website. They bounced the ball back to me and suggested that I do it. I decided that I had time to do something nice and useful." When Vadim took on the task and joined the board of directors as webmaster, he had a few things to learn. "I had designed databases and Internet applications, but I had never been a webmaster," he says. "I enjoy being part of an organization that shares knowledge and creates value for the local Oracle community," says Vadim, when asked why he volunteered so much of his time.

## Technology Background

Vadim has been working in the technology arena for many years. He started working with PDP-11 and VAX in the '80s. One of Vadim's earliest experiences was designing a database and applications at a biological research center outside Moscow after studying at Kalinin University in Russia. With broad interests in computer technologies, he had chances to work with Novell and UNIX networks, Lotus Notes and ISAM databases, SQL server and Oracle. He is currently doing Oracle database administration and application development using PL/SQL, Java, and XML technologies.



Vadim Barilko

## Family Life

Vadim lives with his big family in Walnut Creek. Every member of his family deals with computers: his wife doing statistical programming for medical research, oldest daughter drawing "Business Objects" reports, teenagers studying and playing computer games, and the youngest, a nine-year-old son, writing his Halloween story.

During his vacation time, Vadim

enjoys interesting travel adventures. In Russia he spent months river rafting in Siberia and did a lot of backpacking in the mountains. In California he's climbed Mount Shasta's north face (14,162 feet) and hiked to Clouds Rest (9,926 feet) in Yosemite. "I like mountains and enjoy hiking, downhill, and cross country skiing," says Vadim.

## Teamwork

Vadim especially likes the challenge of climbing with a team. "Success is only possible together," he says, continuing, "When you've reached the top, it's exciting, but the entire success is when the whole team comes home safely."

Vadim compares this type of teamwork with working on a software development team. "If you design an application and build a database and it works, it's like reaching the top. But, you only have full success if, after some time, the application still helps the business and the database is still reliable." Vadim's trait of looking at long-term results has been a great asset to the NoCOUG board! ▲



## Want to publish your data offline?

We can turn your Oracle data into a dynamic application running on an auto-start CD.

Catalogs, programmed training, reference data, maintenance support, and conference proceedings are some excellent application examples for our easy-to-use unplugged technology.

Use your favorite web browser to access data-driven menus and fast text searching. You have complete design freedom for your data display. Password-protected and encrypted content can be used. On-the-fly calculations can customize data based on user input.

PerYourRequest - Integrated Web and CD Application Development.

Need an answer? Call (408) 777-9949 or [info@PerYourRequest.com](mailto:info@PerYourRequest.com)

# LECCO SQL Expert

SQL with a Higher IQ

LECCO SQL Expert automates the task of performance tuning and optimization of database applications by providing developers and DBAs with expert knowledge of SQL.

Through its proprietary Feedback Searching engine, it provides optimum SQL statements—guaranteed to provide the correct results. No guesswork or hopeful suggestions, statements are actually rewritten.

- Oracle
- Sybase
- MS SQL Server

Join the ranks of the experts and download a FREE evaluation copy today!

Where the experts go for performance.

**LECCOTECH**  
www.leccotech.com



## RESOURCE CORNER

### Online Resources Courtesy of LA Oracle Users Group

**T**here's a wealth of information for Oracle DBAs and developers on the website of the Los Angeles Oracle Users Group (LAOUG). For a number of years, they have been posting white papers and presentations from speakers who present at their quarterly conferences. A sampling of topics you can download from their August 2002 conference include:

- Exploring the Mysteries of StatsPack
- Oracle9i Partitioning
- Oracle9i Business Intelligence Architecture

You'll find these presentations and a lot more at the LAOUG website at <http://www.laoug.org>. ▲

Dive into any database administration challenge. Without the risk.

DBArtisan. The fastest, easiest—and safest—way to manage multiple databases.

[www.embarcadero.com/download/artisan.htm](http://www.embarcadero.com/download/artisan.htm)



do more now

©2000 Embarcadero Technologies, Inc. All rights reserved. Embarcadero Technologies and DBArtisan are registered trademarks of Embarcadero Technologies, Inc. All other products are the trademarks of their respective companies.

**INFORMATICA®**  
NOW YOU KNOW™

Visit [informatica.com](http://informatica.com) to see how you can use your enterprise data for smarter decisions.

# The Expensive Parse Phase and the Cost of Using Synonyms

by Murali Vallath, Elogex Inc.

## Abstract

Parsing of SQL statements is the most expensive part of a transaction, especially when the statement is executed for the first time and when there is no similar statement found in the SGA. When resolving object names, Oracle's parse algorithms take a different execution, or resolution path, when synonyms are used. The outcome of this is high utilization of resources. The following information session looks at the cost of using public and private synonyms. It also addresses the option of accessing the object directly as a schema owner. The following questions will be addressed and answered during the discussion: 1.) Why would one consider using a synonym? 2.) Why is there such a high level of cost in using synonyms? and 3.) How can this cost be reduced?

## Introduction

Statements executed by a user or a process have to pass through a major operation to validate the security and accessibility of the objects referenced in the statement. Each time a user executes a statement, the statement is validated for security and accessibility. Prior to 7.0, every statement, same or different, had to pass through this operation and was a major performance problem from the standpoints of execution time, memory conception, etc. Measures taken to avoid this situation were to add more and more memory to the system and to allocate enough shared pool. In fact, there was a period when people measured performance with the amount of shared pool that was allocated. The major reason for this was that the statements executed were validated and, after validation, stayed in memory but were flushed out when other statements (same or different) needed space in memory for execution.

The introduction of the concept of shared SQL with version 7.0 of Oracle in 1993 brought major performance improvements. The performance improvements noticed were due to sharing of SQL statements by processes/sessions that used the same SQL statement to access the data. However, even with the introduction of this feature of sharing SQL statements, certain operations, such as security and accessibility checks, continued to be performed for new users or new SQL statements. Operations that required validation of SQL statement against the data dictionary, as well as loading of the corresponding metadata into memory, were not required. However, in the case of users repeating their operation, the related security, accessibility, metadata, and the statements by themselves were found in memory and the parsing operations minimized, greatly improving performance.

This improvement was a great benefit for the transaction processing OLTP environments, where users accessed the data to retrieve a single row or a few numbers of rows. The repeated check on security, checking object accessibility; metadata validation, etc., hindered performance tremendously. Though a major issue in performance was resolved with version 7.0, when Library Cache was introduced, the parse operation still continues to be the most expensive phase in a transaction life cycle.

While Oracle continued to improve the performance of the parse interface, as with the introduction of the Cost Based Optimizer in Oracle 8.0, under this functionality the Oracle kernel decides to retrieve the rows using the cardinalities of the tables, the prior access patterns, etc. Prior to the introduction of the cost based optimizer, all SQL optimization was done using the rule-based optimizer. Oracle continued its enhancement with removing the effect of parsing the SQL statements twice in PL/SQL procedures in Oracle 9i. Parsing continues to be a critical area during the development and tuning of the application to validate usage of SQL statements in reducing cost, thus improving performance.

## Oracle's Parse Operation

The best approach to understanding the parse operation is to understand its internal operation. Figure 1.0 (on the following page) explains the internals of the parse operation:

### STEP 1: Create a Cursor

A cursor can be considered as an association between the data area in a client program and the Oracle server's data structures. When a program is required to process a SQL statement, it is required to open a cursor. The cursor contains a pointer to the current row, and, as the rows get fetched, the pointer moves to the next row until all the rows specified by the condition in the statement have been processed and none are left. Since the cursor contains the statement that needs to be executed, the cursor also contains the information required, or conditions to be followed, to retrieve rows. While the cursor contains the statement, it operates independently of the statement. The cursor is created by a program interface called OCI in expectation of a SQL statement. SQL statement processing in general is complex in nature. The various physical components that the SQL statement has to iterate through are:

- Client-side runtime memory
- Server-side runtime memory
- Server-side private SQL area
- Server side shared SQL area, or the Library Cache

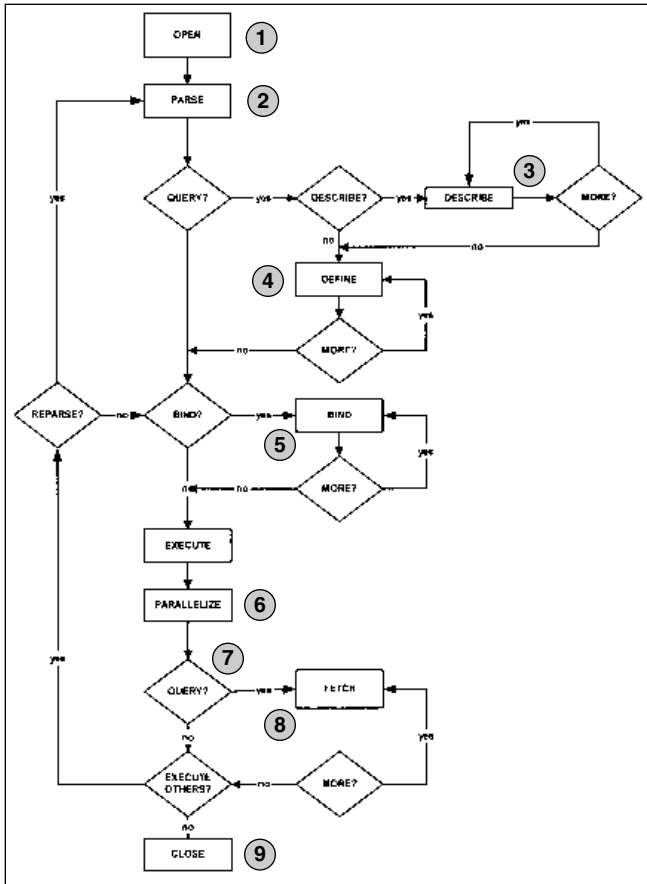


Figure 1.0 illustrates the various stages, or steps taken beginning with the time that the cursor is opened by the processes. A similar parse operation (not illustrated) is also followed by the PL/SQL procedure and the SQL statements embedded inside the PL/SQL code. The steps taken by a DML statement, beginning with the point that a query is executed and opened to the point that it is completed and the results returned could be broadly grouped into nine steps. It should be noted that only the DML statements may require performing all these steps. DDL statements like create, alter, and drop operations are performed in two steps; namely create and parse.

## STEP 2: Parse the Statement

This is the most complicated and expensive phase of the operation. During this phase the statement is passed from the users process to Oracle and is loaded into the shared SQL area. Before loading into the SQL shared area, the following steps have to be completed: the SQL statement is translated and verified, the table and columns checks are performed, and Oracle places a parse lock to prevent definitions from changing. While verifying the definitions, Oracle also checks the user privileges against the referenced objects. Oracle generates the query plan to determine the optimal execution path, followed by loading the statement into the shared SQL area.

During this phase of the operation, Oracle has to determine if the statement being executed is not identical to a previously executed statement. Based on the findings, Oracle will determine whether to use the parse information from a previous execution. This is the advantage of the Library

Cache feature introduced with version 7. Based on the repeated usage of the statement and Oracle finding the parse information in the Library Cache, it determines if this is a hard parse or a soft parse.

### Hard Parse

When a statement is executed for the first time and Oracle is not able to find any information pertaining to this statement in the library cache, it has to do a complete parse operation, which is also referred to as the *hard parse*. During this initial operation Oracle has to perform these several steps:

- Syntax checks on the statement
- Semantic checks on the statement
- Data Dictionary validation for the objects and columns
- Name translation of the objects (with reference to synonyms)
- User privileges for the objects referenced in the SQL Statement
- Generation of an execution plan, with the help of the Optimizer
- Loading of the statement into the Library Cache

Even after all of these steps have been completed and the statement has been loaded into the library cache/shareable area, it is actually not in a shareable state. This is due to the fact that Oracle has a lock on the cursor/statement header (cursors created by Oracle have two parts; the cursor header and the cursor body). The lock is placed on the header when the cursor is initially opened and released when the cursor is closed. Once the cursor is closed, the cursor obtains the shareable state (i.e., other sessions could see this statement in the library cache).

Before storing the statement in the library cache, Oracle computes a hash value and stores it on the hash table. This hash value is compared against a similar query from another user. It should be noted that Oracle, as a first attempt to process a SQL statement, generates a hash value and compares that with the ones in the library cache to determine if any matches exist before the parse operation. While normally there is only one copy of the header that could be found in the library cache, there could be many copies of the cursor body existing in the library cache. The reason for duplicate cursors, or many cursor bodies for the same cursor header, is due to variations on the SQL statements. After the initial header and body creation, subsequent bodies are created under the same header under the following conditions:

- The threshold of the bind variables changes because the string length of the contents of the bind variable



has changed. For example, if the employee table is initially checked with “TOM,” and the next query checked the employee table with “Vandenlangenburgh-Johnson,” Oracle notices that the length of the bind value has changed and cannot reuse the existing cursor body; therefore it creates another one.

- SQL statements using literals instead of bind variables.

To overcome the difficulties phased by most applications that use literals, Oracle has introduced a new parameter called `CURSOR_SHARING = FORCE`. This parameter creates bind variables for all literals, thus forcing cursor sharing.

### Soft Parse

As discussed above, before parsing the SQL statement, Oracle ensures generation of a hash value and compares the hash value with the statements already in the library cache. Now, when the match is found, Oracle may have to do a simpler parse, depending on certain conditions. This parse is a *soft parse* and could be one of these kinds:

1. If the user is accessing the SQL statement for the first time and finds it in memory, Oracle has to ensure that the statement is exactly the same as the user has requested and ensure that the user has authentication to use the objects referenced in the SQL statement.
2. If the user is accessing the SQL statement for the second time and finds it in the library cache, Oracle still has to validate to ensure that the grants for the objects referenced in the SQL statement have not changed.

The parse operation not only consumes memory and reduces the response time, but also consumes CPU cycles to complete the query.

### STEP 3: Describe the Results

In this step of the process, DESCRIBE provides information about the select list items. This step is more relevant when using dynamic queries using an OCI application.

### STEP 4: Define Query Output

While the statement is executed and data is retrieved, appropriate memory variables are required to hold the output information. This step takes care of defining this memory area for the variables.

### STEP 5: Bind Variables

If bind variables are used, Oracle is required to bind them. This gives the Oracle server the address where bind values will be stored in memory. Usage of bind variables is another important factor for efficient usage of memory and the overall performance of the system. Bind variables help in repeated execution of the same statement. Usage of literals (alternative to bind variables), causes statements to be unique and does not allow sharing of SQL statements. The

non-shared statements consume memory and cause excessive parse operation.

### STEP 6: Parallelize the Statement

This step is performed only when parallelism is required and configured. When parallelism is chosen, the work of a query is divided among a number of slave processors. If this is required for parsing or if the query is eligible for parallel execution, it is determined during the parse step and the appropriate execution plan has to be defined. The plan is executed during this step of the process.

### STEP 7: Execute the Statement

This step executes the SQL statement and retrieves rows from disk or memory and places the values into the bind variables. This ensures the completion of the statement to produce the desired results.

### STEP 8: Fetch Rows for a Query

From the bind variables defined in step 5, the values that are obtained as a result of the statement execution and placement of values in the bind variables are moved into the output variables of the process. These results are returned in a table format to the calling interface.

### STEP 9: Close the Cursor

Once the data is returned to the calling interface, the cursor is closed.

### Object Name Translation

It has been shown that the hard parsing of SQL statements is very cumbersome and probably the most complex phase in the entire execution phase of a SQL statement. One of the inner steps during this operation is the translation of object names, which can also be expensive, depending on the methods used to implement object name references. The object could be accessed either by fully qualifying the object owner with the object or by defining synonyms.

### Direct Access

When using the direct access method to access the objects, the schema owner normally qualifies the object names with the schema name. The application user will log directly as the schema owner or as a user who has direct reference to the objects by way of grants.

Therefore, to select rows from the employee table owned by Elogex, the application, or user executing a SQL statement, will have to qualify the object name referenced in the statement with Elogex.

```
SQL>Select * from elogex.employee;
```

Under this approach, while there is direct reference to the objects without any intermediate translation, the schema owner, or owner of the object, is exposed, causing a security risk. This is truer in the case of applications that are developed for sale to outside organizations than for those developed for internal end-user usage.

## Synonyms

A synonym is an alias for any table, view, snapshot, sequence, procedure, function, or package. Oracle has more than one method to access tables defined in a particular schema. Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary. A synonym is not actually a schema object itself, but instead, a direct reference to a schema object. Synonyms are often used for security and convenience. Synonyms are used to:

- Mask the real name and owner of a schema object
- Provide public access to a schema object
- Provide location transparency for tables, views, or program units of a remote database
- Simplify the SQL statements for database users

Synonyms are very useful in both distributed and non-distributed database environments because they hide the identity of the underlying object, including its location in a distributed system. This is advantageous because, if the underlying object must be renamed or moved, only the synonym needs to be redefined and applications based on the synonym continue to function without modification. Synonyms can also simplify SQL statements for users in a distributed database system. For example, if a table name is very long, it can be shortened by using a synonym and by referencing the synonym in the SQL statement. A synonym can be public or private. An individual user can create a *private synonym*, which is available only to that user. Database administrators most often create *public synonyms* that make the objects available for general, systemwide use by any database user.

## Public Synonyms

The special user group named PUBLIC owns a *public* synonym and every user in a database can access it. In order to create a public synonym the user should have CREATE PUBLIC SYNONYM privilege.

Example of creating a public synonym:

1. Start SQL\*Plus:  
SQL\*Plus: Release 8.1.7.0.0 - Production on Mon Jan 14 22:27:43 2002  
(c) Copyright 2000 Oracle Corporation. All rights reserved.  
Connected to:  
Oracle8i Enterprise Edition Release 8.1.7.0.1 - Production With the Partitioning option JServer Release 8.1.7.0.1 - Production
2. Connect as the schema owner.  
SQL> connect Elogex\_owner@saturn  
Enter password: \*\*\*\*\*
3. Ensure that the schema owner has the CREATE PUBLIC SYNONYM privilege assigned.
4. Create a public synonym for the EMPLOYEE table, as shown below.  
  
SQL>create public synonym emp for Elogex\_owner.employee  
Synonym created.

5. Use the public synonym, emp, to select all columns from the employee table.  
SQL>select \* from emp;

EMPLOYEE_ID	EMPLOYEE_NAME
1001	Jeff
1002	Ben
1003	Wayne
1004	Murali

## Private Synonyms

A *private* synonym is in the schema of a specific user who has control over its availability to others. In order to create a private synonym the user should have CREATE SYNONYM or CREATE ANY SYNONYM privilege.

1. Start SQL\*Plus:  
SQL\*Plus: Release 8.1.7.0.0 - Production on Mon Jan 14 22:27:43 2002  
(c) Copyright 2000 Oracle Corporation. All rights reserved.  
Connected to:  
Oracle8i Enterprise Edition Release 8.1.7.0.1 - Production With the Partitioning option JServer Release 8.1.7.0.1 - Production
2. Connect as the schema owner.  
SQL> connect Elogex\_owner@saturn  
Enter password: \*\*\*\*\*
3. Ensure that the schema owner has the CREATE ANY SYNONYM privilege assigned
4. Create a private synonym for the EMPLOYEE table, as shown below.  
SQL>create synonym emp for Elogex\_owner.employee  
Synonym created.
5. Use the private synonym, emp, to select all columns from the employee table.  
SQL>select \* from emp;

EMPLOYEE_ID	EMPLOYEE_NAME
1001	Jeff
1002	Ben
1003	Wayne
1004	Murali

## Dependency Resolution

When translating object names to the actual object, Oracle has to take a few steps through the architecture to obtain or reference the object. When an object is referenced via a SQL statement or is referenced directly, Oracle attempts to resolve all references to the object in the following order:

1. Searching in the current schema. If a referenced object is not found in the current schema, the following occurs:
2. Oracle attempts to resolve the reference by searching for a private synonym in the same schema. If a private synonym is not found, the following occurs:
3. Oracle moves on, looking for a public synonym. If a public synonym is not found, the following occurs:
4. Oracle searches for a schema name that matches the first portion of the object name.

5. If a matching schema name is found, Oracle attempts to find the object in that schema. If no schema is found, an error is returned.

Due to the various steps that Oracle has to validate through to determine the existence of the object, synonym resolution is time consuming in the name translation phase of the parse operation. To understand the complexities of the name resolution process, the method by which Oracle translates the public synonym to the actual object must be examined.

1. The SQL statement with the object reference is loaded into the library cache.
2. Since the object referenced by the cursor does not exist in the users schema, the dependency is a negative dependency.
3. The SQL statement also has a positive dependency on the public synonym itself.
4. The library cache lookup establishes a lock on the object and this requires memory to be allocated.
5. The library cache lookup also establishes a pin structure and this also requires memory to be allocated.
6. Loading of metadata for an object involves the dictionary cache for information not present in the library cache and this in turn requires memory as well.
7. Because more memory is utilized for the above steps, there is a potential for aging out data in the library cache.
8. Since memory is involved there is potential of latch contentions.

The above steps all indicate that resolving a public synonym is not a simple process but is a complex one towards name resolution. While the processing required to resolve public synonyms is huge, in the case of private synonyms, this complexity is half of that encountered by a public synonym. In the case of a direct reference, when the user makes a connection to the database and executes the SQL statement, the object referenced in the SQL statement is fully qualified, and the name resolution is direct due to the direct reference qualification to the object name.

### Cost in Using Synonyms

If a test is performed using the scenarios of direct reference, private synonyms, and public synonyms, the cost of CPU usage doubles with each of the scenarios respectively. For a statement that refers to a single table only, the CPU usage during the parse phase is approximately 1:2:4 for a fully qualified reference, a private synonym, and a public synonym respectively. For complex SQL state-

ments, the cost of synonym usage is even greater. The synonym-based solutions involve potentially large library cache dependency tables and are sensitive to the aging out of the library cache objects representing the table name in the users' own schemas. These library cache objects cannot be "kept." In an instance with high parse rates, the use of synonyms increases CPU usage dramatically, and often causes contention on the latches that control access to the library cache, dictionary cache, and shared pool.

### Conclusion

The parse operation inside the Oracle architecture is a very complex operation. Every step should be taken towards developing and tuning SQL statements to obtain best performance benefits. Usage of synonyms is another critical area where considerable thought should be given when defining and using public vs. private synonyms. While the fear of testing the code, migrating applications to run on different environments, and greater maintenance activities for the DBA are all of concern to most people, they could be overcome through other mechanisms like adding code (`ALTER SESSION SET CURRENT_SCHEMA`) into the application and would accomplish the desired behavior. At a minimum, the developers should consider using private synonyms in place of public synonyms, because the ultimate goal is to provide higher performance to customers.

### ABOUT THE AUTHOR

Murali Vallath has over 16 years of experience designing and developing databases. He is an Oracle Certified Database Administrator. He currently works as a Senior Database Architect with a primary focus in designing, performance, and tuning of Oracle databases for Elogex in Charlotte, North Carolina. His specialty area is with Oracle Parallel Server/Oracle Real Application Clusters. He has presented at various conferences and has published papers both externally and internally to his employers. Murali is the President of the Charlotte Oracle User's Group. He can be reached at [mvallath@elogex.com](mailto:mvallath@elogex.com) or [muralivallath@hotmail.com](mailto:muralivallath@hotmail.com).

### References

1. Scaling Oracle 8i by...
2. <http://www.ixora.com>
3. Oracle documentation
4. Private Vs. Public Synonyms – Internal paper – Murali Vallath ▲

*Reprinted with permission from the IOUG Live! Conference, April 2002.*



[ SPACE EXPERT™ GIVES YOU A SMARTER WAY TO  
*visualize* YOUR DATABASE'S SPACE PROBLEMS. ]

It's not always easy to see where pesky space problems are lurking, ready to degrade the performance of the database that your business depends on. And it's not always easy to keep your database operating at peak levels without lots of highly trained DBAs.

Until now, that is.

Introducing Space Expert™ for Oracle from BMC Software. It intelligently and automatically visualizes, isolates, analyzes and corrects space-related problems. Enabling even a novice DBA to perform the work of many. Which means the IT department's SLAs are good as gold. IT managers can concentrate on managing. Your database's performance is enhanced. And your company is more competitive than ever.

See for yourself. **Register for a 30-day trial copy of Space Expert for Oracle today at [www.bmc.com/distdata/spaceexpert](http://www.bmc.com/distdata/spaceexpert).** Then you'll discover just how easy we make it for you to help your database, and your business, really perform. Because it takes intelligence, not hocus pocus.





# Many Thanks to Our Sponsors

NoCOUG would like to acknowledge and thank our generous sponsors for their contributions. Without this sponsorship, it would not be possible to present regular events while offering low-cost membership dues. If your company is able to offer sponsorship at any level, please contact NoCOUG President Joel Rosingana at joelros@pacbell.net.

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRON TEXACO

ORACLE CORP.

## Thank you! Year 2002 Gold Level Support Vendors:

- BMC Software
- Database Specialists, Inc.
- Embarcadero Technologies
- Informatica
- LECCOTECH
- Quest Software, Inc.

For information about our Gold Level Vendor Program, contact the NoCOUG vendor coordinator via email at: vendor\_coordinator@nocoug.org



## TREASURER'S REPORT

Judy Lyman, Treasurer

Beginning Balance  
July 31, 2002

\$ 57,375.39

### Revenue

Advertising	-
Sponsorships	-
Membership Dues	\$ 3,514.27
Meeting Fees	1,180.00
Vendor Receipts	5,500.00
Interest	67.48
Vendor Mailing	-

Total Revenue

\$ 10,261.75

### Expenses

Regional Meeting	\$ 4,686.91
Journal	5,654.41
Membership	-
Administration	1,408.90
Website	1,775.00
Board Meeting	490.13
Miscellaneous	-
IRS	430.00
FTB Tax	404.00

Total Expenses

\$ 14,849.35

Ending Balance  
September 30, 2002

\$ 52,787.79

# All About Oracle's Touch-Count Data Block Buffer Algorithm

Craig A. Shallahamer, OraPub, Inc.

## Introduction

Oracle introduced a *touch-count* based data buffer management algorithm to address the growing size, performance requirements, and complexities of current-day systems. Data block buffers are no longer managed using a modified least recently used (LRU) algorithm. The touch-based algorithm introduced in Oracle 8i significantly reduces latch contention, forces buffers to earn their way higher up the main list, and allows for increased granular buffer performance optimization opportunities. This paper details Oracle's touch-count algorithm, how to monitor its performance, and how to manage it for optimal performance.

## New Algorithm Needed

Things change. In the Oracle world, things change fast and with significant impact. Because of change and need to remain competitive, Oracle has essentially been forced to re-evaluate many of its central algorithms. For example, the redo buffer algorithm has changed over the past few years. But why a new data block buffer algorithm? Answer: Bigger caches, performance requirements, and control. Over the past few years the possible size of an Oracle data block buffer cache has dramatically increased. I have seen data block caches over a gigabyte. While I am not recommending caches of this size, it does demonstrate people's desire to cache the entire database. There is, however, a problem. Oracle's data block buffer algorithm was designed and built with the expectation of significant physical I/O. DBAs who have experimented with very large data block buffers have personally experienced a performance *decrease*. That is correct, decrease. Why? Oracle was designed for I/O and so when a buffer cache nature shift occurs, a different algorithm must be employed.

Simply being able to manage a very large buffer cache is not enough. In fact, maintaining performance with a large cache is not enough. To remain competitive, Oracle systems must not only support large caches, but also provide significant performance gains. Again, this required a new algorithm.

In addition to larger caches and increased performance requirements, Oracle has been moving its marketing position and technical position from distributed databases to a massive central Oracle database position.

This requires increased control. Oracle is slowly providing the DBA with ways to segment a single database system into smaller sub-databases. As you will see, Oracle's touch-count implementation is highly tunable, which will provide some of the required control to manage the buffer cache in a highly dynamic and very large cache.

When put together, bigger caches, increased performance requirements, and control, it should not be surprising that

Oracle has chosen a new data block buffer cache algorithm. As the next section discusses, what is surprising to many is that Oracle has been changing the database block buffer algorithms for many years.

## History of Oracle Buffer Cache Management

If you have been working with Oracle for quite a few years, you may never have thought about it, but Oracle has been modifying its data block buffer algorithm. The center of attention is the least recently used list—or LRU for short. A LRU is simply a list of pointers to the cached database blocks. In general, toward one end of this list are the popular blocks and on the other end the not-popular blocks. The popular end of the LRU chain is called the *most recently used* or the MRU end of the list. The not-popular end of the LRU chain is called the least recently used or the LRU end. And yes, it is sometimes confusing talking about the LRU end of the LRU chain. But after a while you get used to it. Regardless of the naming conventions, the general idea is to keep popular blocks cached to minimize physical I/O. In the beginning, Oracle implemented what is known as a *standard* least recently used algorithm. Whenever a buffer was brought into the cache or touched again while in the cache, the buffer's pointer (remember, the blocks are never physically moved, just a pointer to the physical memory block) was moved to the MRU end of the LRU chain. The thinking here is that the not-popular blocks would naturally migrate towards the LRU end of the LRU list and perhaps their contents filled with a more popular block.

However, there is a cache killer called the *full-table scan* where each and every table block is placed into the buffer. If the buffer cache is 500 blocks and the table contains 600 blocks, all the popular blocks would be replaced with this full-table scanned table's blocks. This is extremely disruptive to consistent database performance because it forces excessive computer system usage and basically destroys a well-developed cache.

To combat this destructive force, the famous *modified least recently used*, more frequently referred to as the *modified LRU* algorithm, was implemented. While Oracle has changed this algorithm and our control over it over the years, its basic operation has remained the same. The modified LRU algorithm places full-table scanned blocks read into the buffer cache at the LRU end of the LRU chain and only permits a limited number of these blocks to exist in the cache at once. This prevents a full-table scan from flushing the entire buffer cache. While this might seem like all our buffer cache problems are solved, think again. How about a large index range scan? Picture hundreds of index leaf blocks flowing into the buffer cache. The modified LRU algorithm only addresses full-table scan issues, not index

block issues. As an interim solution and to remain competitive, Oracle implemented multiple buffer pools. While most DBAs do not use multiple buffer pools and the touch-count algorithm makes them less attractive, they can be very effective when used with precision. The *keep pool* is designed for small objects that, for application response time issues, should *always* be cached. The *recycle pool* is designed for larger objects that can disrupt normal and smooth buffer cache operations. And finally, the *default pool* is for everything else. To maintain smooth and effective buffer cache operation, a very fast algorithm must be implemented that essentially forces every buffer to seriously earn its way into the buffer cache and to seriously earn the right to remain in the buffer cache. A touch-count algorithm is all about forcing each and every block to earn its way not only into the buffer cache but to remain in the buffer cache. As you will read below, the words *seriously earn* is an understatement. Even the default touch-count related instance parameters make it very difficult for a block to simply remain in the buffer cache. It's almost cruel how many hurdles a buffer must *continually* jump to simply remain in the buffer cache. How this works in general and Oracle's specific implementation is presented in the next section.

### Touch-Count Buffer Management

Since a touch-count based algorithm, sometimes referred to as a *count frequency scheme*, is more complicated than a simple LRU algorithm, there must be some motivating factor before the benefits of using a touch-count algorithm surpass its cost and also provide increased performance when compared to Oracle's past LRU algorithms. We have already discussed these motivating factors. They are bigger caches, performance requirements, more control, full-table scan access, and index access. These factors create an intense and very strenuous cache environment. It is only then that the touch-count algorithm becomes economical. At the core of any touch-count based algorithm are the harsh requirements placed on each buffer to not only remain in the cache, but to remain in the MRU end of the LRU list. The general touch-count algorithm does this by assigning each buffer a *counter*. Each time the block is touched, its counter is incremented. The value of this touch counter is used to assign a kind of *value* or *popularity* to each buffered block. This touch-count is then referenced and manipulated during buffer cache activity. The reference and manipulation specifics are touch-count algorithm implementation specific. This means, Oracle's implementation may be different than another vendor's implementation.

### Oracle's Touch-Count Implementation

Oracle does a number of nasty things to make it very difficult for a buffer to remain in the buffer cache. Only the chosen few can remain in the MRU end of the LRU chain. What does Oracle do that makes the life of a buffer so difficult? The next few sections specifically address this topic.

#### Midpoint Insertion

The LRU is divided up into the two basic areas or regions: a *hot region* and a *cold region*. All buffers in the hot

region are called *hot buffers* and all buffers in the cold region are called *cold buffers*. There is a midpoint marker between the hot region and the cold region. The midpoint pointer moves to ensure the appropriate number of buffers are in the hot region and in the cold region. That is, the midpoint pointer is not associated with a specific block.

By default Oracle divides the LRU evenly. That is, the hot region is composed of 50% of the buffers and the cold region is composed of 50% of the buffers. This can be controlled by manipulating the instance parameter, `_db_percent_hot_default`. Increasing this instance parameter will increase the percentage of buffers in the hot region; that is, buffers above the midpoint. When a server process reads a block from disk and places into the buffer cache, it places the buffer in the middle of the LRU chain; that is, between the hot and cold regions. This is known as *midpoint insertion* and is a fundamental concept in Oracle's touch-count algorithm implementation. Actually, the buffer is placed just into the hot region. It is important to grasp that the buffer is *not* placed at the MRU end of the LRU chain and that the buffer must work or earn its way to the MRU end of the LRU chain.

#### Touch-Count Incrementation

In concept, each time a buffer is touched, for whatever reason, its touch count is incremented. In practice this is not the case, which makes it more difficult for a buffer to increase its touch count. In a buffer's life, there are times when it will be of intense interest to server processes and then drop out of favor very quickly.

This "bursty" kind of buffer activity would wreak havoc in a touch-count based algorithm. So to reduce this problem, Oracle only allows a buffer's touch count to be increased, at most, once every three seconds by default. This does not mean a buffer's touch count can only be incremented once within any three-second period. If you graph out the time sequence, you will see a touch count could be incremented twice within three seconds. This "three-second period" can be modified by the instance parameter, `_db_aging_touch_time`.

Another important aspect of touch-count incrementation is that the buffer is not moved; that is, its pointer is not moved. Touch-count incrementation and buffer pointer movements are *independent*. Buffer pointer movement will be discussed below.

A latch is simply a token that one must possess to perform some specific action. *Anytime* there is an Oracle buffer, cache, list, node, etc., activity, a latch or latches are involved to ensure cache correctness. Well...now there is an exception. Oracle updates a buffer's touch count without a latch. This means a buffer may be manipulated while the touch count is being incremented. If this occurs, Oracle assures us the worst that could happen is the touch count is not actually incremented and no cache corruption will result. So once again, it can be difficult to get that touch count increased. While shocking at first, this makes a lot of sense.

#### Buffer Movement

As mentioned previously, when a buffer is brought into the buffer cache, it is placed just above the midpoint and into

the hot region. Unlike least recently used algorithms, Oracle's touch-count algorithm will not move a buffer just because it is touched. Yes, its touch count will probably be incremented, but it may not be moved. When a server process is looking for a free buffer to place a physical Oracle block into the buffer cache, it must first find a *free* buffer. A *free* buffer is simply a buffer whose contents match the copy on disk. That is, there is no difference. If there is a difference, the block is tagged as *dirty*. When a server process is searching for a free buffer or a database writer (DBWR) is looking for dirty blocks (sometimes called *victims*), and if a block's touch count is observed to be greater than two, it is moved to the MRU end of the LRU chain. The default block movement threshold of two is controlled by the instance parameter, `_db_aging_hot_criteria`. Oracle's touch-count implementation is tough on buffers! When a buffer is moved to the MRU end of the LRU chain, its touch count is usually reset to zero! (Experiments and people I trust indicate that when a buffer is moved to the MRU end of the LRU chain, its touch count is reset to 0 unless `_db_aging_stay_count` (default: 0)  $\geq$  `_db_aging_hot_criteria` (default: 2), in which case the touch count is set to the current touch count divided by two.)

As I'll describe below in more detail, the now popular buffer, which just had its touch count chopped to zero will need its touch count sufficiently incremented or it could be paged out to disk.

### Hot-to-Cold Movement

If a buffer is moved from the cold region to the hot region (i.e., to the MRU end of the LRU chain), the midpoint marker must also move one position to ensure the proper number of hot and cold buffers are present. When this occurs and with no fault of its own, a buffer will be forced to cross over the threshold from the hot region into the cold region. When this occurs, Oracle once again resets the buffer's touch count to one. Even if the buffer's touch count is 250, if it crosses the threshold, then its touch count will be set to 1. The threshold crossing touch-count reset value is controlled by the instance parameter `_db_aging_cool_count`.

### Touch-Count Performance Tuning

Performance tuning can consist of only a few fundamental steps. First the system under investigation must be monitored. Second, if there is contention or a problem, hopefully it will be identified. Next, an analysis must be performed and finally a solution implemented. Discussed below is how to monitor Oracle's buffer cache and ways to utilize our control over the touch-count algorithm by appropriately manipulating the available instance parameters.

### Touch-Count Performance Monitoring

Monitoring Oracle buffer cache performance is algorithm independent. However, if a buffer cache related problem is identified, your solution options are algorithm dependent. The best way to accurately identify a buffer cache problem, or any Oracle server based performance problem for that matter, is to monitor response time and its components. While this topic is discussed in detail in my response time paper [12], I will quickly summarize. Response time analysis

considers both service time and queue time. Service time is the CPU time used during a database operation (e.g., buffer cache management) and queue time is time the CPU is waiting on something to continue processing (e.g., latching). Ratio based analysis does not consider service time or queue time, but focuses on Oracle system activity, and therefore it *leads* one into where the problem *might* reside. Session wait analysis effectively identifies queue time and its components, but does not consider service time. This can sometimes lead to incorrect problem identification. Response time analysis considers both service time and queue time in its analysis, thereby *always* correctly identifying where the real problem resides. When queue time components are identified and categorized, the performance bottleneck is quickly and confidently identified. If the buffer cache is the main problem issue, queue time will be greater than service time and one of the following Oracle wait events will consume the majority of the queue time. The buffer cache associated wait events are *LRU chain latch*, *free buffer wait*, *cache buffer chains latch*, *buffer busy wait*, *db file sequential read* (index based physical reads), and *db file scattered read* (full-table scan based physical reads). Once buffer cache contention is discovered and if the touch-count algorithm is being used, then one of your options is to manipulate the touch-count related instance parameters. This is discussed in a following section.

### Classic Buffer Cache Performance Tuning

While manipulating the touch-count related instance parameters, which is discussed in the next section, can be effective in resolving buffer cache performance issues, directly addressing the wait events is probably your best solution. Below is a brief list of what to do to address each of the key buffer cache related wait events. I am sorry, but this paper is not the appropriate place to explain *why* I make the below suggestions or *how* to actually perform what I suggest. If you have specific issues or questions, please feel free to email me.

**LRU chain latch wait.** Reduce the number cache blocks being touched, reduce the number of processes requiring an LRU chain latch, increase the number of LRU latches, reduce latch hold duration, add more CPUs, and use faster CPUs.

**Free buffer wait.** Increase I/O subsystem throughput capacity, alter the application to produce fewer dirty blocks, give the DBWRs more power, and increase the number of data block buffers to better absorb activity bursts.

**Buffer busy wait.** Determine if the same buffer is busy or if there are many, many buffers that have been busy. If the same buffer is busy, then fix the application or reduce the block's popularity. If many buffers are busy, then increase I/O subsystem throughput capacity, or reduce the number of busy buffers.

**Cache buffer chains latch wait.** Determine if there are many copies of the same block in the buffer cache. If there are many copies, then fix the application or reduce the block's popularity. Otherwise, increase the number of block buffers, increase the number of hash buckets, and use faster CPUs.

**Db file sequential or scattered read waits.** Change the



application to reduce the physical I/O requirements (e.g., tune the SQL) and increase I/O subsystem throughput capacity.

### Touch-Count Parameter Performance Tuning

Recommendations must be based upon solid experience. When solid experience is not available, a relevant and trustworthy experiment must be performed. In this section I will first list and briefly describe the relevant touch-count instance parameters, then I will detail my research experiment, and finally I will summarize the results.

### Touch-Count Instance Parameters

There are five relevant touch-count related instance parameters that we can use to our advantage. Each of these parameters has been discussed above, but for reference purposes, I listed each below.

- **\_db\_percent\_hot\_default.** The percentage of block buffers that reside in the hot region. Default 50 (percent).
- **\_db\_aging\_touch\_time.** The window of time where a buffer's touch count can only be incremented by one. Default 3 (seconds).
- **\_db\_aging\_hot\_criteria.** The threshold when a buffer is being considered to be moved to the MRU end of the LRU chain. Default 2 (touch count).
- **\_db\_aging\_stay\_count.** Involved with resetting the touch count when a buffer is moved to the MRU end of the LRU chain. Default 0 (touch count).
- **\_db\_aging\_cool\_count.** The re-assigned touch-count value when a block moves from the hot region into the cold region. Default 1 (touch count).

### Research Setup

While the general effect for most instance parameters can be for the most part predicted, altering touch count parameters is not quite that easy. To remove any bogus thinking and to remove any personal bias, I performed a series of rigorous experiments and will simply report to you what I discovered. The experiment was set up to place four different stresses on the database system. They were cache buffer chain latch contention, LRU latch contention, free buffer wait contention, and full table scan and index physical read wait contention. I was able to create each of these basic stresses by running specific scripts and purposefully setting various instance parameters. However, I was not able to force significant LRU latch contention, which is an indication of how well the LRU algorithm is now working!

For the latch waits, I recorded what is called the *latch impact*. Latch impact is a simple calculation based upon `v$latch`, as follows: `sleeps*(sleeps/decode(gets,0,1,gets))`. To determine if the touch-count instance parameters affect latch activity, I recorded the latch impact. For non-latching waits, that is, free buffer waits and physical read waits from a full-table or index scan, I recorded the actual wait time. To determine if the touch-count instance parameters affect latch activity, I recorded the latch impact. Due to OOW publication deadlines, this paper only reports my research regarding the touch-count instance parameter `_db_per-`

`cent_hot_default`. However, by OOW 2001, I should have completed my research, which will record the impact of changing all five touch-count parameters. You can download the latest paper version at OraPub's website at <http://www.orapub.com>.

Before each unique stress and instance parameter change, I recycled the database system. I then began stressing the system and gave it a 60-second warm-up period. Then thirteen times, at one minute intervals, I gathered complete Oracle and operating system statistics. What I show in this paper are just a few of the statistics gathered. With the statistics in hand, I was then able to perform statistical analysis to determine what, if any, changes the touch-count instance parameters had on the various systems.

### Research and Thought Analysis

First off and of utmost importance, if response time analysis does not point to a queue time problem predominately composed of buffer cache contention (the wait events have previously been discussed), then messing with the touch-count parameters is of no real value. If response time analysis does point to a queue time problem predominately composed of buffer cache contention, then messing with the touch-count parameters *can* be of real value. At the end of this section are selected experimental statistics for each experimental run. Below are some thoughts regarding changing the five key touch-count instance parameters. My hope is that this will provide direction and a catalyst towards your own specific situation analysis. Remember our objectives are to minimize buffer movement and keep the cache full of the most popular blocks.



## TECH TIPS

### Secure Configuration Guide for Oracle9iR2



Oracle Corporation has published a new database security checklist on the Oracle Technology Network. This ten-page white paper is a quick read and offers lots of good suggestions for keeping the data in your database secure and less susceptible to unauthorized access. Although specifically written for Oracle9i Release 2, most of the recommendations apply to earlier versions as well. Download the Secure Configuration Guide for Oracle9iR2 at [http://otn.oracle.com/deploy/security/oracle9i/pdf/9iR2\\_checklist.pdf](http://otn.oracle.com/deploy/security/oracle9i/pdf/9iR2_checklist.pdf). ▲

***\_db\_percent\_hot\_default.*** The experimental data (see figure below) clearly shows latch activity (cache buffer chains latch and LRU latch) and free buffer wait time is significantly affected by changes to this instance parameter. Based upon the experimental data, the optimal setting is contention dependent. The data clearly demonstrates that reserving 75% or more of the buffer cache as the hot region is detrimental to performance. Therefore, unless you really understand what this parameter is doing, you understand what your application is doing to the buffer cache, and you feel you should increase this parameter above 75%, don't! It is very interesting to note that changes in this parameter had no statistically significant effect on scattered read waits (physical FTS read waits). If you want more blocks to be considered hot, increase this parameter. If you want to push more blocks out into the cold, decrease this parameter.

***\_db\_aging\_touch\_time.*** If there is too much block movement or popular block burst activity, increasing this parameter value will reduce blocks reaching the movement threshold, therefore reducing buffer movement. Severe cache buffer chains latch or LRU latch contention is an indication of too much block movement. Increase this parameter to reduce sudden activity disruptions and to reduce buffer movement. However, be careful. If you increase this parameter too much, you are essentially devaluing popular blocks.

***\_db\_aging\_hot\_criteria.*** If you need to slow down block movement, increase this parameter's value. The need to

slow down block movement can be detected by significant cache buffer chains latch or LRU latch contention. If you want to make it more difficult for a buffer to be moved (to the MRU end of the list), increase this parameter. The result will be that only the really popular blocks will remain in the cache.

***\_db\_aging\_stay\_count.*** If popular blocks are being pushed out of the cache too often (evidenced by a low buffer cache hit ratio), then increasing this parameter value will help popular blocks stay in the cache longer. But increasing the parameter too much will allow nearly-popular blocks to initially receive a higher touch count without really earning it. This will essentially reduce the meaning or value of a high touch count.

***\_db\_aging\_cool\_count.*** Increasing the value will make it easier for blocks to stay in the cache and will also slow down and smooth the cooling affect and increase the likelihood of blocks' promotion. If popular blocks are being prematurely pushed out of the cache (evidenced by a low buffer cache hit ratio), increasing the parameter value will help these blocks remain in the cache longer. However, if this value is too high, unpopular blocks will also remain in the cache, resulting in block thrashing evidenced by significant cache buffer chains latch or LRU latch contention.

However, it does not significantly (90% confidence level) impact scattered read wait activity. The "latch impact" value is a powerful yet simple calculation of the impact of latch contention. The latch impact is calculated as sleeps\*(sleeps/decode(gets,0,1,gets)).

### Concluding Thoughts

Based upon my experimental research and thinking about this for quite a while, I've come away with three key points:

1. Oracle is constantly changing their internal algorithms to adapt to change,
2. Manipulating touch-count instance parameters can significantly affect buffer cache activity and performance, and
3. More rigorous and comprehensive research needs to be performed. This will take days of computer time. However, by the time you receive this paper at IOUG 2002, this research may have been completed, documented, and this paper updated. The most recent version of this paper will always be available on OraPub's website at <http://www.orapub.com>.

### Acknowledgements

A special thanks to my email acquaintances, clients, and students who have brought forth a plethora of stimulating discussions and challenging dilemmas. These situations coupled with my unusual enthusiasm for Oracle performance analysis has evolved into this technical paper. A special thanks is in order for Mr. John Beresiewicz. His thorough review, expertise in the Oracle 8i buffer cache, and his willingness to share his knowledge has always been deeply appreciated.

(continued on page 26)

Oracle Touch-Count Experimental Results											
Instance	_db_percent_hot_default	_db_aging_stay_count	_db_aging_cool_count	_db_aging_touch_time	_db_aging_hot_criteria	LATCH IMPACT - FREE BUFFER WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))		SCATTERED READ WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))	LATCH IMPACT - FREE BUFFER WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))	LATCH IMPACT - FREE BUFFER WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))	LATCH IMPACT - FREE BUFFER WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))
						FREE BUFFER WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))	SCATTERED READ WAIT TIME (SLEEPS/DECODE(GETS,0,1,GETS))				
ORA81601	75	1	2	1	1	10.41	2.37	14.02	10.41	2.37	14.02
ORA81602	25	1	2	1	1	16.58	1.90	14.02	16.58	1.90	14.02
ORA81603	75	1	2	1	1	16.75	1.93	14.02	16.75	1.93	14.02
ORA81604	25	1	2	1	1	10.21	1.89	14.02	10.21	1.89	14.02
ORA81605	95	1	2	1	1	11.15	1.71	14.02	11.15	1.71	14.02
ORA81606	75	1	2	1	1	10.15	1.84	14.02	10.15	1.84	14.02
ORA81607	25	1	2	1	1	10.10	1.82	14.02	10.10	1.82	14.02
ORA81608	50	1	2	1	1	10.15	1.82	14.02	10.15	1.82	14.02
ORA81609	75	1	2	1	1	10.45	1.72	14.02	10.45	1.72	14.02
ORA81610	25	1	2	1	1	10.44	1.72	14.02	10.44	1.72	14.02
ORA81611	75	1	2	1	1	126.27	24.75	14.02	126.27	24.75	14.02
ORA81612	25	1	2	1	1	141.75	51.70	14.02	141.75	51.70	14.02
ORA81613	50	1	2	1	1	129.25	32.90	14.02	129.25	32.90	14.02
ORA81614	75	1	2	1	1	184.50	42.90	14.02	184.50	42.90	14.02
ORA81615	25	1	2	1	1	148.75	32.70	14.02	148.75	32.70	14.02
ORA81616	75	1	2	1	1	176.17	33.50	14.02	176.17	33.50	14.02
ORA81617	25	1	2	1	1	168.43	34.97	14.02	168.43	34.97	14.02
ORA81618	50	1	2	1	1	168.17	34.67	14.02	168.17	34.67	14.02
ORA81619	75	1	2	1	1	171.56	34.93	14.02	171.56	34.93	14.02
ORA81620	25	1	2	1	1	171.77	34.78	14.02	171.77	34.78	14.02

Figure 1. This is the experimental results. Data was gathered 13 times every minute to yield a more statistically correct value, although I realize it would have been much better to gather more samples. Performing statistical significance testing clearly shows that changing the *\_db\_percent\_hot\_default* parameter does significantly (90% confidence level) impact latch and free buffer wait activity.

# Stand By Your Standby

by Michael S. Abbey, The Pythian Group, Inc.

## Preamble

The initials TLC—in some circles they stand for The Learning Channel, in some the acronym stands for The Lifetime Commitment, or even Taxi and Limousine Commission (NYC). The most familiar expansion of these initials, Tender Loving Care, is what this paper is all about. Oracle's standby database technology has been around since the early '90s—and needs lots of TLC. To compare this piece of Oracle's e-solution to the Rock of Gibraltar would be an understatement; the standby database technology is more solid than even that famous landmark. With that said, let's move into the meat (or cauliflower for all you vegans) of this paper.

Standby databases are just that. They sit in recovery mode waiting patiently for their turn to be the master. So many standby databases pay their dues, put in years of being called *the standby* and never get their turn for prime time. Interestingly enough, that's a good thing. If you never use your standby for production purposes, that could be a testimony to the stability of your production site. In a nutshell, this is what we are going to cover in this paper:

- Standby database setup
- Ensuring the standby is up-to-date
- Nologging impact on the standby

The live presentation at the conference will delve into a few issues that we have not covered here; specifically, testing the usability of the standby, creating a sort segment on a read-only standby, and helping bring a standby up-to-date when its recovery is hopelessly behind the master. We have developed a few tips that will make your standby management life cycle easier. They are highlighted in this text using the following convention:

*Some of these tips will make your life easier from day one—they have served us well in the field for years.*

## Standby Setup

This is a big topic, a high-level summary of which we will cover in this section. We're cutting corners on some of the technical details, but this section will arm you with the basics.

## Getting Database Files to the Standby Site

The seed for your standby setup is your most recent successful hot backup. We stress the word successful in this context since you will end up with an unusable standby if the backup that seeds the standby did not complete successfully. Let's get a list of database files to send to the standby site. We mostly use the UNIX *scp* or *scp2* command, pulling the files from a *gzip*'ed hot backup location. In the code below, the hot backups have been written to */u01/hot*.

```
SQL> set pages 0 lines 999 trimsp on feed off echo off
SQL> spool standby_push.sh
SQL> select 'scp /u01/hot'||file_name||'.gz' oracle@st1:
2 file_name||'.gz'
3 from sys.dba_data_files;
scp /u01/hot/u01/data/herc/tdata01.dbf.gz
oracle@st1:/u01/data/herc/tdata01.dbf.gz
scp /u01/hot/u02/data/herc/rbs01.dbf.gz
oracle@st1:/u01/data/herc/rbs01.dbf.gz
scp /u01/hot/u04/data/herc/system01.dbf.gz
oracle@st1:/u01/data/herc/system01.dbf.gz
. . .
SQL> spool off
```

Take the output from that SQL command, *standby\_push.sh*, and run it on the master site. For this script to run error free, you must have the matching directory structure on the standby site as on the master to receive the *gzip*'ed backup files.

## Un-Compressing on the Standby Site

Let's now create a script to un-compress the files sent to the standby site.

```
SQL> set pages 0 lines 999 trimsp on feed off echo off
SQL> spool gzipminusd.sh
SQL> select 'gzip -d '||file_name||'.gz'
2 from sys.dba_data_files;
gzip -d /u01/data/herc/tdata01.dbf.gz
gzip -d /u02/data/herc/rbs01.dbf.gz
gzip -d /u04/data/herc/system01.dbf.gz
. . .
SQL> spool off
```

You then proceed to the standby site and run *gzipminusd.sh* to prepare the files recently delivered for the standby. Just about done—there are two more setup related issues we cover next.

## Getting INIT.ORA to the Standby Site

We need to get a copy of the master's initialization parameter file to the standby site, which usually is a simple *scp* or a wonderful facility we call *mousetp*. Let's *mousetp* the init file to the standby site as follows—this is no big deal, but *mousetp* may be something new for many readers.

- Proceed to the master's \$ORACLE\_HOME/dbs.
- cat the init file.
- Place its contents in the copy/paste buffer with the mouse and your scroll back buffer.
- Proceed to the standby's \$ORACLE\_HOME/dbs.
- vi (you can use another editor if you are so inclined) the properly named init file.
- Paste the contents of your buffer into the new init file.
- Exit and save!

Voilà—we like *mousetp*; it's similar to its sister transfer program called *ftp*, but there are no firewall or password issues to deal with.

## The Standby Control File

Now the heavy-duty stuff starts . . . let's go get a control file to use for our standby database. The control file is created on the master site and propagated to the standby site as follows:

- Enter SQL\*Plus as SYS or another user with SYSDBA.
- Enter the command **alter database create standby control file as '{scf\_name}' reuse;** (the name is your choice, just make it meaningful—meaningful is good as it embraces an approach less prone to human error)—we use the name *stdbyctl.ctl* and will refer to it in further bullets in this list.

*Get used to using the **reuse** keyword, which will allow you to overwrite a copy of the standby control file that may be left over from a previous creation.*

- Proceed to the \$ORACLE\_HOME/dbs directory, the place Oracle drops all un-qualified filenames it creates in UNIX.
- Push the standby control file to your standby site.
- Inspect the *control\_files=* line in *init* on your standby to find out where its control files reside, with a sample shown next:

```
control_files = (/u01/data/herc/control01.ctl,
                /u02/data/herc/control02.ctl,
                /u04/data/herc/control03.ctl)
```

- Copy *stdbyctl.ctl* to each unique location, */u01/data/herc*, */u02/data/herc*, and */u04/data/herc* in our case.
- Proceed to each unique directory location, and create soft links pointing at *stdbyctl.ctl*.

*Use the soft link approach as illustrated next—this removes the need to rename the standby control file as it's copied to its correct location(s).*

```
cd /u01/data/herc
ln -s stdbyctl.ctl control01.ctl
cd /u02/data/herc
ln -s stdbyctl.ctl control02.ctl
cd /u04/data/herc
ln -s stdbyctl.ctl control03.ctl
```

## Getting the Standby Ready for Recovery

Everything is now in place. There are essentially two modes within which a standby database operates. In managed recovery mode, it sits in anticipation of the arrival of a new archived redo log, and applies it when transfer from the master site completes. The other mode is manual, where it is your responsibility to trigger the recovery operation at a pre-determined time interval.

*We have had much more success with the manual recovery triggering approach, not relying on Net8 to transfer the archived redo logs to the standby site. Your experiences may*

*differ, but in this context we have been much happier with doing the push of log files to the standby.*

*If you are running Oracle Enterprise Edition, you can choose which approach to use. Net8-based automatic propagation of archived redo logs is not available if you are using Oracle Standard Edition.*

Using the manual recovery approach, opening the standby is accomplished as follows:

- Proceed to standby site as oracle, and enter SQL\*Plus again as a user with SYSDBA.
- Issue the command **startup nomount;** followed by **alter database mount standby database;**
- You then find yourself in a dialogue with the standby database recovery activity similar to that shown next. After this listing, we will get started on the activity we call standing by your standby.

```
SQL> recover standby database;
ORA-00279: change 94029277 generated at 12/22/2001 13:00:00 needed
for thread 1
ORA-00289: suggestion : /u01/oraarch/herc/herc_118603.arc
ORA-00280: change 94029277 for thread 1 is in sequence #118603
Specify log: (<RET>=suggested | filename | AUTO | CANCEL)
```

With the manual recovery approach, we use an assortment of UNIX scripts to enter SQL\*Plus and feed the recover command followed by **auto**, until the recovery terminates when an archived redo log file cannot be found.

*Be careful how you do error handling with this manual approach as an Oracle error will be returned when the standby recovery runs out of archived redo logs to apply. That oracle error is not fatal and must be ignored when monitoring success of the recovery.*

## Monitoring the Standby

There are a number of issues that arise when managing a standby environment, the least of which answer the following questions:

1. Is the standby mounted?
2. Is the standby up-to-date?
3. Has there been any *nologging* activity against the master?
4. LIVE PRESENTATION—Has a datafile been added to an existing tablespace or a new tablespace added?
5. LIVE PRESENTATION—Is the standby usable?
6. LIVE PRESENTATION—Are all the required archived redo logs being saved long enough on the master?

The answers to the first three of these questions serve as the subject material for the balance of this work. The answers to the last three will be covered at the presentation during Live 2002. Failure to attend to some or all of the issues may render your standby database useless. There is nothing worse than painstakingly setting up a standby and then, when it is needed most, finding out it is corrupt or

unusable for an assortment of other reasons. Three words will help you ensure the standby is OK—test, test, and, did we mention, test!

### Ensuring the Standby Is Mounted

You always want to ensure your standby is mounted; if it happens to be down for some reason, recovery cannot proceed. Keeping in mind the standby is mounted but not open; we have no access to any traditional data dictionary views (such as DBA\_ series), only being able to work with the V\$ views. Of interest in assessing the standby database status is V\$LOG\_HISTORY, using its RECID column in particular. Since we can describe the V\$ views when the standby is mounted, the following code will display the makeup of V\$LOG\_HISTORY:

```
echo "stsys/shsk90 as sysdba
desc v$log_history;" | $ORACLE_HOME/bin/sqlplus -s >
$HOME/logs/vdollar
```

We then, as shown in the next listing, grep the output of this output file, looking for the text RECID; finding this text means the standby is mounted. If the standby were down, about all that would appear in the log file is the test “ORA-01034: ORACLE not available”. Notice how we do an “exit 1” if the RECID text is not found.

```
FOUND=`grep RECID $HOME/logs/vdollar`
if [ "$FOUND" = "" ]
then
echo NOT THERE
exit 1
fi
exit 0
```

### When the Last Archived Redo Was Applied

You need to know the status of the standby and the timestamp of the most recent archived redo log that has been applied. This is an interesting concept. One could easily say, “Who cares about the timestamp of the recovery exercise on the standby? We have verified it’s mounted, so it

will keep itself current without the required intervention.” Neglecting to perform this check can have disastrous results, as illustrated in the next scenario. Suppose a master (size ~250Gb), normally generates between four and six gigabytes of redo per 24-hour period. During a newly crafted data load, all of a sudden that quantity of redo swells to 43 gigabytes, and the standby gets hopelessly behind the master. We check the system time on the standby site, and find Mon Dec 31 13:10:00 EST 2001, and a check on the standby database yields results similar to the following:

```
SQL> select to_char(max(first_time), 'DD-MON-YYYY hh24:mi:ss')
2 from v$log_history;
TO_CHAR(MAX(FIRST_TI
-----
31-DEC-2001 03:05:00
```

In this hypothetical situation, the standby recovery exercise is over 10 hours behind the master (10h05m to be exact). As a smart DBA, it’s wise to monitor the recovery on the standby on a regular basis. Suppose a bit later, you check the system time and find Mon Dec 31 13:40:00 EST 2001, a mere 30 minutes after the last check. The same SQL query returns the value 31-DEC-2001 03:36:00 from V\$LOG\_HISTORY. This means that in this 30m elapsed at the O/S level, the standby has gone from being 10h05m behind to 10h04m behind current time. Thus the standby has gained one minute for an elapsed clock time of 30 minutes! This is not good—you do the math—if the standby gains 1 minute every 30 minutes, the 604 minutes it is behind could theoretically take as long as 604\*.5 hours or 302 hours for the standby to catch up. That’s a mere 12 or so days!

*There are a number of ways to deal with this unfortunate situation; consult the PowerPoint handouts from my presentation for a discussion of solving this quandary.*

Before calling this a wrap, we are going to look at the mess that can be caused by nologging transactions on the master.



## TECH TIPS

# Data Modeling for the Data Warehouse

**W**. H. Inmon’s latest article in DBAzone.com delves into data modeling for the data warehouse. Inmon writes, “In order to be effective, data warehouse developers need to show tangible results quickly. At the same time, in order to build a data warehouse properly, you need a data model. And everyone knows that data models take huge amounts of time to build. How then can you say in the same breath that a data model is needed in order to build a data warehouse and that a data warehouse should be built quickly? Aren’t those two statements completely contradictory?”

Inmon goes on to provide some very interesting answers and explanations on this subject. For the full article, and other articles by Inmon, check out <http://www.dbazine.com/inmon14.html>.

## Detection of Nologging Activity on the Master

Making yourself aware of nologging activities on the master is a primary piece of information you need at your fingertips. We often use the *nologging* keyword when interacting with data previously stored in the database. We use the keyword most often in SQL\*Plus with the *create index*, *alter index*, and *create table . . . as select* constructs. As well, SQL\*Plus insert statements with the */\*+ append \*/* hint and SQL\*Loader direct path loads can render your standby next to useless. Suppose there have been a series of nologging transactions against a POSTOFFICE table on your master site. Down the road, for some reason, the standby is activated and a query against that same table delivers the following error:

```
ORA-01578: ORACLE data block corrupted (file # 45, block # 2534)
ORA-01110: data file 1: '/u01/oradata/herc/mydata38.dbf'
ORA-26040: Data block was loaded using the NOLOGGING option
```

It's our job to track nologging activity against the master, using yet another dynamic performance view. The following query can be used to detect unrecoverable activity against the master. The query expects a parameter, accepted into the code and used for the date comparison. The UNRECOVERABLE\_TIME column is set when a nologging activity is executed upon the data that resided in any datafile.

```
select a.file_name, a.tablespace_name,
to_char(b.unrecoverable_time,'dd-mon-yyyy hh24:mi:ss')
from sys.dba_data_files a, v$datafile b
where a.file_name = b.name
and nvl(unrecoverable_time,to_date('&1')-1) > to_date('&1');
```

Setting the date for comparing in this query is the only outstanding piece. Suppose you built your standby on January 23, 2002, knowing there was no outstanding unrecoverable activity as of that date. Keep track of that date somewhere and pass it into the query as that command line parameter. Suppose it were Thursday February 14, and the log file tracking nologging on the master showed the following results:

```
/u01/oradata/herc/many_indx_01.dbf MANY_INDEX 13-feb-2002 02:14:38
/u04/oradata/herc/many_data_11.dbf MANY_DATA 13-feb-2002 04:51:54
/u04/oradata/herc/many_indx_12.dbf MANY_INDEX 13-feb-2002 02:19:48

3 rows selected.
```

Guess what? Your standby's in trouble. You are susceptible to the error stack shown earlier 01578, 01110, and 00600 indicating an unusable portion of the standby. Here's how to fix it.

- Shutdown the standby database.
- Proceed to the hot backup containing these three files. Ensure the timestamp for the copies of these three files is earlier than the time registered in V\$DATAFILE for a nologging insert. In our example this would be Feb 13 02:15, Feb 13 04:52, and Feb 13 02:20 respectively.

*If any file's most recent completed hot backup is dated later than its unrecoverable insert time, simply make a new copy of that file and use it on the standby.*

- Push a new copy of the offending file(s) over to the standby site, and uncompress if necessary.
- Re-create the standby control file on the master as discussed in "The Standby Control File" section earlier in this paper.
- Push that new control file to the standby site and position it in the proper locations as dictated by the standby's initialization parameter file.
- Get the standby ready for further recovery with a **startup nomount** followed by an **alter database mount standby database**.
- Change the value of the date parameter passed to the unrecoverable insert check from 23-JAN-2002 to 14-FEB-2002 in preparation for tomorrow's check.

## Wrap Up

I love that word—it reminds me of writing COBOL programs in the 16th century, and naming the closing housekeeping module T5400-WRAPUP! We've bitten off a lot in this dissertation on standing by your standby. There's nothing more embarrassing, not to mention career-limiting, than planning an intricate network of fault tolerance, entertaining a suite of Oracle's technologies, and not being thorough enough to ensure it all works. The standby setup itself, though not always as straightforward and error-free as what we discussed in this paper, is pretty easy. What gets complicated down the road is the management and TLC of that standby. With all due respect to the late Tammy Wynette, let's call this a wrap—sung to the tune of Stand By Your Man (preferably in the key of "C" for country):

*Stand by your standby  
Give it the care it de-serves  
Do track nologging inserts  
Or you'll be up all night.*

## About the Author

Michael S. Abbey is the lead in a very successful offering of remote DBA services with the Pythian Group in Ottawa, Canada. Michael has co-authored 10 works in the Oracle Press series, the most successful being the Beginner's Guide series. Michael is an instructor at the IOUG's University Master Class, delivering guidance in the area of Advanced DBA Best Practices. He has been working with the Oracle Server for the past 16 years, having seen versions 3 through 9i. ▲

*This article is reprinted with permission from the IOUG Live! Conference, April 2002.*

# Uncommon Leaders: Managing the Blur of IT!!

By Rich Niemiec, TUSC

*Editor's Note:* Following is an excerpt about leadership from IOUG president, Rich Niemiec.

## Abstract

Where have all the heroes gone? The answer is that they are all around you waiting for you to pull the best out of them. This presentation is a sabbatical from the technical presentations. It is a time to look at the non-technical traits that allow a professional to succeed and propel those around them to succeed. While technical proficiency is paramount to success, character traits are equally important to the success of a team and a business. This presentation will be a reminder of the importance of the character traits as an ingredient to success. Books on leaders and leadership have been around for a long time, yet never has the need for leaders in the information systems world been so great. The World Wide Web is becoming the main medium for world transactions. Members of the IS community must now become leaders and visionaries within their companies as the landscape of technology changes beneath us daily. The goal of this talk will be to discuss general leadership traits that make the uncommon leaders that will build the Fords of the future.

The Laws of Leadership haven't changed. You still need to make the correct decision at the correct time to successfully move a business forward. Exercising good timing as well as good character is imperative for a great leader. A leader who exercises good character traits will find that he or she will always be aware of timing concerns since employees and customers will communicate them. Maxwell's results of timing are listed below:

## Results of Timing

Wrong action/wrong time = Disaster  
Right action/wrong time = Resistance  
Wrong action/right time = Mistake  
Right action/right time = Success

Remember that leaders make sacrifices for everyone else. As the quote goes, "One of the costs of leadership is your own rights. As responsibility grows, you lose the right to think about yourself." As the leader, you are a contributor to humanity.

## Are You a Boss or a Leader?

"Where have all the Heroes Gone?" —John Mariotti

Are you a boss or a leader? If you are a boss, you are depriving yourself of the time that you desperately need to succeed. Leaders succeed because the team they've built doesn't need or want someone breathing down their neck. If you find

that you constantly need to "ride" a person to get work out of them, then you are spending your efforts in the wrong place. Take the time to correctly position them in the organization and then teach them to take charge of that area. You will be surprised when they start helping you retrieve time instead of soaking it from you. Below are some traits of bosses and leaders. Investigate the changes that you can make in positioning your staff to move yourself from the left to the right.

## Bosses

Creates fear  
Says "I"  
Job should be done  
Rely on authority  
Drives  
Fixes blame  
Rules/10% problem  
Causes resentment  
Work is drudgery  
Problems are disasters

## Leaders

Builds confidence  
Says "we"  
Career should be forged  
Rely on cooperation  
Leads  
Solves problems/fixes mistakes  
Works along 90% cooperative  
Fosters growing enthusiasm  
Work is interesting  
Problems are opportunities

Which Covey box you find yourself in may also help determine whether you have been acting more like a boss or a leader. Stress is probably the most frequent cause for poor leadership. Investigate the chart below to see causes for missing leadership.

## Bosses

Stress  
Burnout  
Crisis management  
Putting out fires

## Leaders

Vision and perspective  
Balance  
Manage with discipline and control  
Few crises

Focusing on the causes for missing leadership and changing the situation will lead to success. No change means that things in motion will continue in the same direction. Understanding the top issues of the CEO and/or CIO points the leader in the correct direction. As of late 2001, the top issues for 2002 are as follows:

## CEO Top Issues

- Increasing profitability
- Business growth
- eCommerce
- Improve IT performance
- Strategic/cultural
- Mergers and acquisitions

## CIO Top Issues

- Improve IT performance
- Web enablement
- Increase productivity through automation

- Enterprise storage/database architecture
- Knowledge management

### Building a Learning Organization

The companies that will succeed in the current blur of technology are the ones that have built an environment that is a learning organization. The traits of this type of organization are:

- People feel that they're doing something that matters.
- Every individual is growing/stretching.
- Team is more intelligent together than apart.
- Organization aware of its underlying knowledge base.
- Visions emerge from all levels—leaders manage.
- Employees know what's going on.
- Free to inquire about assumptions/biases.
- Employees are colleagues with mutual respect.
- Employees experiment/risk ... not killed for a mistake.

If you don't have a learning environment, you are not prepared for the pace that is upon us. You must make changes that allow your organization to move in this direction. Without change, you will continue in the same direction at a time in history where doing so is not an option that will bring success.

### Integrity

*"Oil and truth are bound to come to the surface at any time."*

Integrity is honesty that stands the test of time with character. It takes a lifetime to gain, yet only a moment to compromise. As we head from the information age into the knowledge age, people and relationships become more important. The uncommon leader who deals with co-workers with integrity will be the ones that rise to the top. People often say that your character is your destiny. Many people judge character by weighing integrity much greater than the other character attributes. The companies that have employees and leaders with integrity will be the ones that become the Fords of the future as the knowledge age matures. Integrity is instilled in employees by praising integrity when it is displayed and not allowing character that does not display integrity.

### Moral Courage

*"A well beaten path may not lead in the right direction."*

Moral courage is standing up for what is right instead of standing where it is convenient or comfortable. Many say that moral courage is no longer a necessity for our leaders. I beg to differ. I believe that the leader lacking in moral courage may get results in the short term, but will be surpassed in the long term by leaders and companies that exercise moral courage with co-workers and customers. Moral courage is instilled in employees by setting an example and correcting them tactfully when they fail to understand the complex feelings of another individual.

### Physical Courage

*"It's easy to look on the bright side as long as it is turned your way."*

Physical courage is the tenacity of mind and of body in

adversity. Physical courage (mostly of the mind) is beginning to determine the fine line between success and failure in today's business world. Technology continues to set a blistering pace, leaving those who must learn and adapt to change exhausted both mentally and physically. A lack in the supply of IS personnel continues to drain the overworked few who can stay with the pace of technology. The road continues to bend and add stress and time to the job of the IT professional. Those who tenaciously adapt to change will be our uncommon leaders at the companies of the future. Physical courage is instilled in employees by sharing a vision that they desperately would like to achieve and then providing them the tools to get there.

### Self-Control

*"When a person burns with anger or frustration, it is their cue to be silent."*

Self-control is maintaining your composure when you have every physical and mental right to lose it. As the workload increases and the days become longer, how easy it becomes to lose our self-control at least to some minor level. As companies fight for control of the Web and the consumer's attention, the time allowed to get things done will be greatly compressed and stress will be at an all-time high. The business teams rich with self control will be the ones that adapt best to the bend in the road. Self-control is instilled in employees by tactful communication and setting an example.

### Enthusiasm

*"One person with enthusiasm makes the majority."*

Enthusiasm is the ability to bring a team to a level of performance by painting a picture of a vision that they desperately want to achieve. With the stresses of change, continual education and overworked IT professionals among us, leaders need to employ enthusiasm among teams to relieve stress and awaken purpose and vision. Leaders who understand what motivates the individuals of their respective teams will be the ones that succeed. Enthusiasm is instilled in teams by sharing vision.

### Tact

*"If you haven't any enemies to forgive, pardon a few of your friends."*

Tact is being able to communicate with others in a friendly and professional manner. There is no longer a supply of IT professionals who meet the demands of a technology that rules all segments of business and profit. No longer will our leaders be able to get away with treating employees poorly in any way. The tactful leader will succeed; an IT-starved market unwilling to put up with any type of prejudice will bury the untactful leaders. Tact is instilled in employees through tactful communication and setting an example.

### Knowledge

*"Opportunity has a way of calling on those who are already successful."*

Knowledge will be king in the next century. Adapting to changing technology continues to determine success. Those who educate their teams effectively will succeed in their busi-



ness, but more importantly, they will succeed in keeping the employees who will determine the future success of a business. Knowledge is cultivated in employees and co-workers by setting up appropriate learning goals for them and implementing plans that create time for education.

### Initiative

*“When is the last time you tried to punch a hole in the sky?”*

Initiative is taking on a task or carrying out a plan without being told what to do every step of the way. Great leaders have teams of employees that are full of initiative. Their teams rarely need more than the vision or the direction; they will complete the plan for success. Leaders with teams that lack initiative will be buried with an endless task of micro management. Initiative is instilled in employees and co-workers by allowing them to fail in their efforts and then showing them the path for success.

### Respect

*“Talk is cheap but like other cheap things it is liable to prove expensive in the end.”*

Respect is something should be given automatically and yet it is something that is earned over time. The uncommon leaders gain the respect of others by displaying the traits of an uncommon leader, yet they are also the first to show respect for others, no matter how different or complex another individual can be. With the infinite number of personalities that exist in this world, only the true leaders understand and respect the individuals around them. Instill respect in employees by showing respect to them.

### Loyalty

*“Loyalty is working toward something you believe in.”*

Loyalty is standing true to something or someone you believe in. Loyalty continues to be a fleeting trait with employees as “Show me the money” replaces it. America will achieve a lower standard of success if employees continue to shorten their duration of employment. The cost and stress of changing jobs for the both employer and the employee is great. Loyalty must be re-awakened by instilling it in employees through rewards and gratitude.

### Unselfishness

*“You will never become successful without the help of others.”*

Unselfishness is loving and caring for those around you enough to share that which makes up success. More and more blue chip companies are implementing stock option plans in an effort to gain employee loyalty. The leader who unselfishly shares success with others will be pushed to greater heights. Instill unselfishness in employees by noting the times it is displayed and unselfishly sharing success with them.

### Fortitude

*“Your character will be your destiny.”*

Fortitude is strength of mind that allows one to endure adversity or pain with courage and character. Fortitude is probably the true measure of how well we have progressed in life as individuals. It is often a painful reminder of how far we have to go in our development of fortitude when we

are presented with an extremely difficult situation. If we can instill fortitude in our leaders and teams, we are guaranteed success. The best way to instill fortitude in an individual is to notice those times when they exude it.

*“I desire so to conduct the affairs of this administration that if at the end, when I come to lay down the reins of power, I have lost every other friend on earth, I shall at least have one friend left, and that friend shall be down inside of me.”*

—Abraham Lincoln.

### Setting Your Goals Now

Set your goals NOW (record your goals for the list below):

- Goals for success and happiness
- Steps to achieve those goals
- Qualities you will need to achieve those goals
- Qualities that you are deficient in

### Other Sources

There are a variety of sources that will help you gain the character that will drive success. Find a leader of the past and you will see the same leadership traits over and over again. Benjamin Franklin and Abraham Lincoln lived and lead during two tremendous periods of freedom’s history, the American Revolution and the Civil War. Some of the traits that each of them lived by are mentioned below. Benjamin Franklin’s list includes the following: temperance, silence, order, resolution, frugality, industry, sincerity, justice, moderation, cleanliness, tranquility, chastity, and humility.



## Need Part-Time DBA Support?

When you need no-nonsense, straightforward Oracle expertise, call on Database Specialists. We'll roll up our sleeves and help you get things done. With our DBA Pro service, we can configure a part-time or remote DBA program that best suits your needs. You receive:

- A cost-effective and flexible extension of your IT team
- Proactive database maintenance and quick resolution of problems
- Increased database performance and minimized database downtime
- Constant database monitoring with Database Rx™
- Onsite and offsite flexibility
- Reliable support from a stable team of DBAs familiar with your databases

**Call Us Today!**

415-344-0500 • 888-648-0500  
www.dbspecialists.com

ORACLE | CERTIFIED SOLUTION PARTNER

  
**Database Specialists**

In *The Words Lincoln Lived By*, author Gene Grissman points to the following Lincoln traits: determination, courage, honesty, morality, patience, magnanimity, time management, work, diligence, curiosity, vision, assertiveness, tenacity, self-preservation, justice, influence, responsibility, communication, focus, compromise, flexibility, simplicity, energy, tact, conciliation, forgiveness, ambition, study, resourcefulness, ethics, altruism, compassion, trust, achievement, citizenship, democracy, patriotism, tolerance, idealism, self-reliance, piety, timing, adversity, deliberation, research, conviction, freedom, self-discipline, humor, friendship, charity, and life's brevity.

If you don't have a mentor, read a book on one of the above leaders. It will be just like having a mentor. I look at each of these as mentors. Past leaders influence future leaders.

### How They Became Leaders

- Natural gifting 10%
- Crisis 5%
- Influenced by a leader 85%

### Summary

People write books on success and making more money all of the time. What they forget is that the equation for success is not one that can be chased down. The equation is to simply live the traits of the uncommon leader. Only then will success find you! Freedom brought innovation, innovation brought advances rarely seen in the world. If we can couple technological advances with character advances, we will build a future that will make current visions of utopia archaic.

### References

*God's little devotional book for leaders*  
*The Words Lincoln Lived By*, Gene Grissman  
*The Art of Virtue*, Benjamin Franklin  
*The traits of the uncommon leader*, U.S. Marine Corps Manual  
*Uncommon Leaders*, Rich Niemiec, 1982  
*Poor Richard's Almanack*, Benjamin Franklin  
*Uncommon Leaders*, TUSC, 1989  
*The Fifth Discipline Fieldbook*, Peter Senge  
*7 Habits of Highly Effective People*, Steven Covey

### About the Author

Richard J. Niemiec (rich@tusc.com) is the CEO of The Ultimate Software Consultants (TUSC), a Lombard, Illinois based database consulting company. TUSC specializes in the full cycle of database development including Business Modeling, Design, Development, Implementation and Support. Richard has been giving lectures and presentations on Oracle for the past 12 years and is the current President of the International Oracle Users Group Americas (IOUGA). Rich can be reached at TUSC at (630) 960-2909 (www.tusc.com). Please report errors in this article to TUSC. Neither TUSC nor the author warrant that this document is error-free. ▲

*This article is excerpted and reprinted with permission from the IOUG Live! Conference, April 2002.*

## Touch-Count Data Block Buffer Algorithm (continued from page 18)

### References

1. "Advanced Performance Management For Oracle Based Systems" Class Notes (2001). OraPub, Inc., <http://www.orapub.com>
2. "Capacity Planning – Performance Modeling & Prediction" Class Notes (2001). OraPub, Inc., <http://www.orapub.com>
3. Jain, R. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991. ISBN 0-471 50336-3
4. Michalko, M. *Thinkertoys*. Ten Speed Press, 1991. ISBN 0-89815-408-1
5. "OraPub System Monitor (OSM)" tool kit (2001). OraPub, Inc., <http://www.orapub.com>
6. Shallahamer, C. *Avoiding A Database Reorganization*. Oracle Corporation White Paper, 1995. <http://www.orapub.com>
7. Shallahamer, Craig A. (1999). *Direct Contention Identification Using Oracle's Session Wait Views*. Published and presented at various Oracle related conferences worldwide. <http://www.orapub.com>
8. Shallahamer, Craig A. (1999). *Direct Contention Identification Using Oracle's Session Wait Views*. OraPub Internet Video Seminar. <http://www.orapub.com>
9. Shallahamer, Craig A. (2000). *Holistic Problem Isolation Method*. OraPub Internet Video Seminar. <http://www.orapub.com>
10. Shallahamer, C. *Optimizing Oracle Server Performance in a Web/Three-Tier Environment*. OraPub White Paper, 1999. <http://www.orapub.com>
11. Shallahamer, C. *Oracle Performance Triage: Stop the Bleeding!* OraPub White Paper, 2001. <http://www.orapub.com>
12. Shallahamer, C. *Oracle Response Time Analysis: The Next Performance Management Frontier*. OraPub White Paper, 2001. <http://www.orapub.com>
13. Shallahamer, C. *The Effectiveness of Global Temporary Tables*. OraPub White Paper, 2001. <http://www.orapub.com>
14. Shallahamer, Craig A. (1995). *Total Performance Management*. Published and presented at various Oracle related conferences worldwide. <http://www.orapub.com>

### About the Author

Mr. Shallahamer's 17-plus years of experience in the IT marketplace bring a unique balance of controlled creativity to any person, team, or classroom. As the President of OraPub, Inc., his objective is to empower Oracle performance specialists and capacity planners. His specializations include doing and teaching others to do whole system performance optimization and capacity planning for Oracle based systems.

Mr. Shallahamer authors and teaches both of OraPub's key courses: *Advanced Performance Management for Oracle Based Systems* and *Capacity Planning - Performance Modeling & Prediction*. In addition to course development and delivery, Mr. Shallahamer consults, is helping to develop landmark performance management products, and technically reviews Oracle books. ▲

*This article is reprinted with permission from the IOUG Live! Conference, April 2002.*

# NoCOUG Fall Conference

Thursday, November 21, 2002  
Session Descriptions

For up-to-date information, see [www.nocoug.org](http://www.nocoug.org)

## DBA TRACK

11:45–12:45

### **Applying Patches for Oracle Applications 11.x Dreyer's Grand Ice Cream Case Study**

Ramkumar Srinivasan, DBA, *Dreyer's Grand Ice Cream Inc.*

In this presentation, we will detail the techniques used and offer real-life experiences while applying the interoperability patch for Oracle Applications 11.0.3. We will also review our experience while installing patches in Oracle 8i on HP-UX running Oracle application 11.0.3 with server partitioning (Oracle 8.0.6.3). We will also offer a detailed explanation of installing Oracle 8i, adpatch and how to use adpatch, adctrl and adadmin Oracle application utilities.

*(Intermediate Level)*

2:30–3:30

### **Oracle Database Consolidation**

Noel Yuhanna, Exodus, a Cable & Wireless Company

Currently the trend has been to reduce cost, so a lot of corporations are looking into consolidation. The advantages of consolidation go beyond reducing cost—consolidation helps provide better availability and scalability, better ensuring that SLAs are being met. We will look at how to consolidate Oracle databases and what approach to follow. We will also look at the Oracle Database Resource Manager and steps involved in using it to manage resource allocation. Finally, we will look at some of the guidelines for performing a successful database consolidation. *(Intermediate Level)*

4:00–5:00

### **Using Statspack in Oracle8i and 9i to Identify Problems**

Ian Jones, Senior Consultant, Database Specialists, Inc.

This presentation will cover the Oracle-provided Statspack utility. Statspack provides an easy way to identify performance problems and understand performance bottlenecks. We will explain how to install Statspack, collect and analyze data, and describe some real-world situations where use of Statspack led to timely solutions. *(Advanced Level)*

## APPLICATION DEVELOPMENT

11:45–12:45

### **Rapid Application Assessment, Application Mining in an Oracle Environment**

Mark Brooks, Product Manager, CAST Application Mining

What is Application Mining? Why should we do Application Mining? How does it compare to alternative techniques? What do the analysts (Gartner, etc.) think about this area? We will talk a little about the history and have a

discussion of the special challenges presented in distributed applications. Specific examples will address the issues that arise within Oracle projects, Oracle migrations, and integration with Oracle apps. *(All Levels)*

2:30–3:30

### **Oracle9iAS Portal Release 2: Advanced Topics**

Mohamed El-Mallah, Advisory System Engineer, Oracle Corporation

This session discusses some of the Oracle9iAS Portal advanced topics that will help both Portal Developers and Administrators to build and deploy more sophisticated and reliable Portals. Topics included in the session are: Introduction to Oracle9i Application Server and Portal, New Features of O9iAS Release 2, Integrating Websites into O9iAS Portal, Building JSP and Perl Portlets. *(Intermediate Level)*

4:00–5:00

### **Index Strategies for Optimal Performance**

Claudia Fernandez, Technical Services Manager, LECCO Technologies, Inc.

Learn how to take maximum advantage of Oracle 8i and 9i indexing options to achieve optimal performance. This presentation will demonstrate which index strategies are appropriate for particular requirements. Understand how indexes impact performance of a SQL statement and the overall database system. *(Intermediate Level)*

## DATA WAREHOUSING

11:45–12:45

### **Assessing Your Data Asset for Quality and Meaning**

Michael Scofield, Consultant and Author

Most enterprises have more data than they realize. We will explore techniques of surveying the data asset. We will explore the fundamentals of data architecture, data profiling, data quality assessment, and how this new knowledge of data behavior becomes a form of metadata, and should be captured and managed as such. This presentation will include vivid examples of garbage discovered in production databases. *(All Levels)*

4:00–5:00

### **Customer Experience: Building an Oracle Data Warehouse**

Saqib Mausoo and Raimundo Reyes, Argonaut Insurance

Argonaut Insurance has built a data warehouse using Oracle 9i technologies. We have designed the warehouse in Designer 6i, extracted mainframe data through Oracle Pure Extract MVS Gateway, and performed transformation and mapping through Warehouse Builder. The target schema is in Oracle 9i, where we have used some of the new warehouse and OLAP features, and the reporting is through Discoverer 4i (desktop and web version using Oracle 9iAS technology). All in all, an end-to-end Oracle warehouse solution. In this session we describe our experience and the pros and cons of using the above-mentioned technologies. *(Intermediate Level)*

# NoCOUG Fall Conference

Thursday, November 21, 2002

**Location:** The NoCOUG Fall Conference will be held on Thursday, November 21, 2002, at Pacific Gas and Electric headquarters at 77 Beale Street in downtown San Francisco. See complete directions to the meeting location at [www.nocoug.org](http://www.nocoug.org).

- 8:00            **Registration and welcome. Refreshments served.**
- 8:00–9:30     **Continental breakfast and vendor exhibits**
- 9:30–10:00   **General Session**
- 10:00–10:30 **Welcome from our host with Roger Gray, VP and CIO ISTS, and Bob Glynn III, Manager, ADS ISTS, Pacific Gas and Electric**
- 10:30–11:15 **Keynote: “The New Knowledge Era—Making Knowledge Management Work for the Knowledge Economy in Asia” with Dr. Kuan-Tsae Huang, Chairman, Taskco**
- 11:15–11:45 **Morning Break**
- 11:45–12:45 **Parallel Sessions**  
*DBA Track: Applying Patches for Oracle Applications 11.x Dreyer’s Grand Ice Cream Case Study*  
*Application Development: Rapid Application Assessment, Application Mining in an Oracle Environment*  
*Data Warehousing: Assessing Your Data Asset for Quality and Meaning. (Part 1)*
- 12:45–1:45   **Lunch Break**
- 1:45–2:25     **Roundtable Discussions for DBA and Developer Tracks**
- 1:15–1:30     **Break**
- 2:30–3:30     **Parallel Sessions**  
*DBA Track: Oracle Database Consolidation*  
*Application Development: Oracle9iAS Portal Release 2: Advanced Topics*  
*Data Warehousing: Assessing Your Data Asset for Quality and Meaning (Part 2)*
- 3:30–4:00     **Afternoon Break**
- 4:00–5:00     **Parallel Sessions**  
*DBA Track: Using Statspack in Oracle8i and 9i to Identify Problems*  
*Application Development: Index Strategies for Optimal Performance*  
*Data Warehousing: Customer Experience: Building an Oracle Data Warehouse*
- 5:15–??      **NoCOUG Networking and Happy Hour at Beale Street Bar & Grill, 133 Beale Street**
- Cost:         **\$40 admission fee for non-members. Members free. Includes lunch voucher.**



**Thank you to PG&E,  
our meeting sponsor.**

**View session  
descriptions at  
[www.nocoug.org](http://www.nocoug.org).**

**RSVP online at <http://www.nocoug.org/rsvp.htm>**

**NoCOUG**  
P.O. Box 3282  
Danville, CA 94526

FIRST-CLASS MAIL  
U.S. POSTAGE  
PAID  
SAN FRANCISCO, CA  
PERMIT NO. 11882