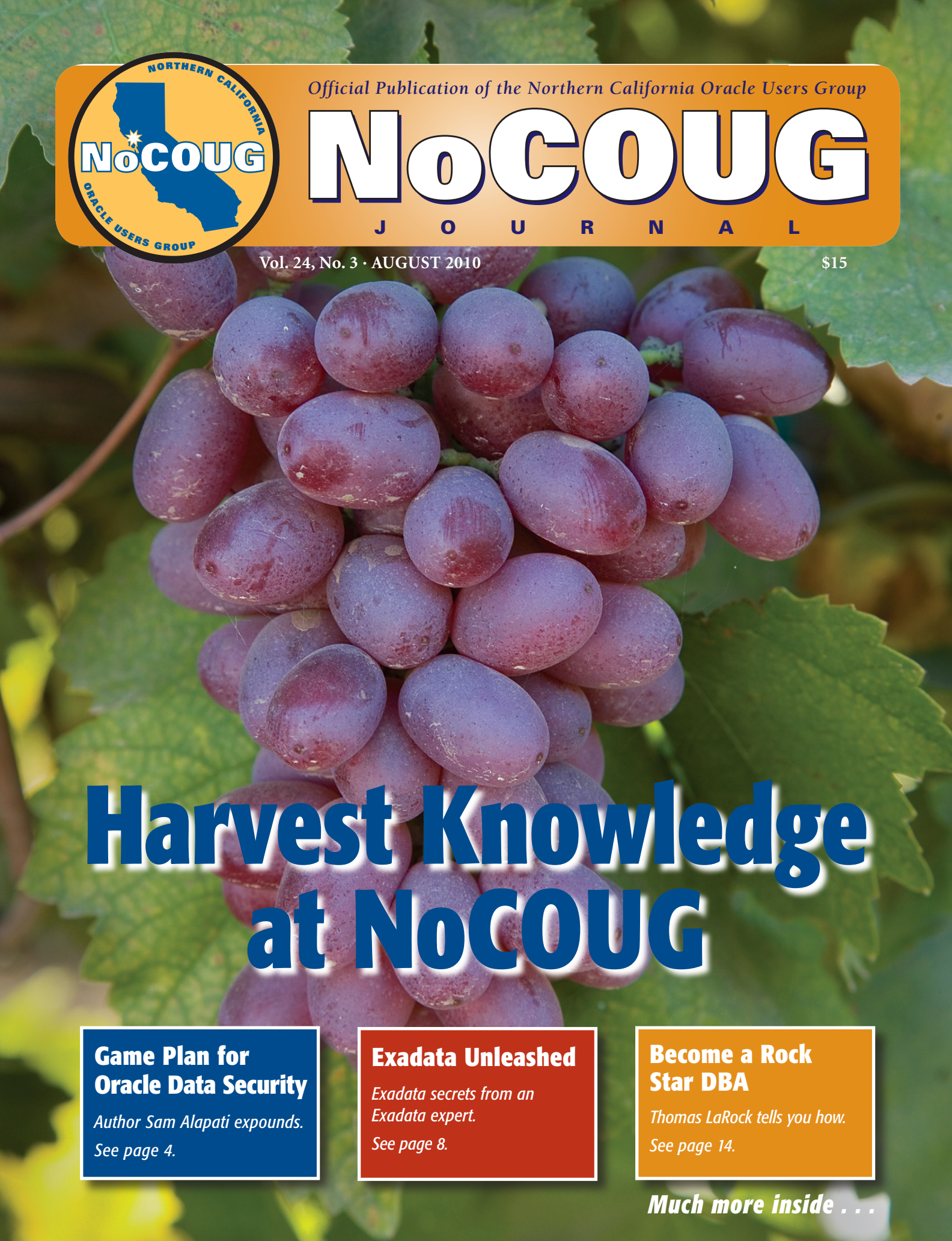
Official Publication of the Northern California Oracle Users Group

NoCOUG JOURNAL

Vol. 24, No. 3 · AUGUST 2010 $15



Harvest Knowledge at NoCOUG

Game Plan for Oracle Data Security

Author Sam Alapati expounds.

See page 4.

Exadata Unleashed

Exadata secrets from an Exadata expert.

See page 8.

Become a Rock Star DBA

Thomas LaRock tells you how.

See page 14.

Much more inside . . .

Sir Oracle?

Oracle guru Craig Shallahamer is a passionate educator with an engaging presentation style. You won't ever mistake him for one of those men of whom Shakespeare wrote:

"There are a sort of men whose visages

Do cream and mantle like a standing pond,

And do a wilful stillness entertain

With purpose to be dress'd in an opinion

Of wisdom, gravity, profound conceit;

As who should say, 'I am Sir Oracle,

And when I ope my lips let no dog bark!'

O, my Antonio, I do know of these,

That therefore only are reputed wise

For saying nothing. . . ."

I hope you'll join me to hear Craig deliver the keynote address at our summer conference on Thursday, August 19, at Chevron in San Ramon. ▲

—Iggy Fernandez, *NoCOUG Journal* Editor

Table of Contents

ADVERTISERS

Publication Notices and Submission Format

The *NoCOUG Journal* is published four times a year by the Northern California Oracle Users Group (NoCOUG) approximately two weeks prior to the quarterly educational conferences.

Please send your questions, feedback, and submissions to the *NoCOUG Journal* editor at **journal@nocoug.org**.

The submission deadline for the upcoming November 2010 issue is August 31, 2010. Article submissions should be made in Microsoft Word format via email.

2010 NoCOUG Board

ADVERTISING RATES

The *NoCOUG Journal* is published quarterly.

Size	Per Issue	Per Year
Quarter Page	$125	$400
Half Page	$250	$800
Full Page	$500	$1,600
Inside Cover	$750	$2,400

Personnel recruitment ads are not accepted.

journal@nocoug.org

Unforgettable Errors

by Hanan Hit

Hanan Hit

Unforgettable. That could describe the first soccer World Cup played on African soil. What happened on and off the field will remain in people's minds forever. It had everything: the good, the bad, and the ugly. New, well-deserved kings were crowned as the world champions—Los Reyes del Mundo. The bad part of the tournament were bad calls by the referees, like the one by Koman Coulibaly of Mali who annulled a legitimate goal by Maurice Edu that would have given the U.S. a 3–2 victory over Slovenia in the first round. The Uruguayan referee Jorge Larrionda called back the goal scored by Frank Lampard, which would have marked a temporary 2–2 tie of Britain with Germany, which eventually won it 4–1. The Italian referee Roberto Rosetti allowed Argentina's first goal in its win over Mexico, although Carlos Tevez was clearly offside.

The unforgettable referee errors led me to think about our own unforgettable professional errors. Bringing down a production database due to issuing a "shutdown immediate" command in the wrong window taught me to use the proper Xterm heading and to check the instance name and host name ten times prior to a shutdown. I'm sure we've all committed some sort of unforgettable error.

Human error is unavoidable in Oracle database management. But adequately preparing for an Oracle project greatly increases our chance of success and reflects positively on our organizations as well as on our careers. Preparing and planning will help, and NoCOUG can assist in this area. Use NoCOUG as a resource in your research and information-gathering stage. At NoCOUG conferences, you can network with others who may have already completed a similar task. The *NoCOUG Journal* and the NoCOUG website are also valuable sources of information.

This brings me to the NoCOUG Summer Conference, taking place on August 19 at Chevron in San Ramon. Craig Shallahamer, a world-renowned presenter, will kick off the day with a keynote presentation, "A Day in the Life of an Oracle Server Process," followed by a presentation titled "Optimizing Internal Serialization Control." The rest of the day will be filled with technical presentations by renowned Oracle experts. The day will be rounded out with the usual networking opportunities, book raffles, and plenty of food and drink. Get all of the conference details and submit your registration at the NoCOUG website.

I look forward to seeing you there! ▲

One-Day Performance Seminar with Craig Shallahamer

Wednesday, August 18, 2010—CarrAmerica Conference Center, Pleasanton, CA

OraPub's One-Day 2010 Performance Seminar was created specifically for experienced DBAs and is squarely focused on advanced Oracle performance analysis. To develop the seminar, OraPub surveyed the most experienced students in the firefighting and advanced analysis courses and specifically asked what were the most memorable, valuable, and relevant aspects of the courses.

Craig Shallahamer is planning a day of intensive learning, which includes multiple aspects of Oracle performance tuning. We will learn how to analyze the resources consumed by competing processes and how to avoid hitting resource capacity limits. We will map the different performance problems and the specific techniques to approach each problem type. We will explore the relations between different components that impact performance and how they influence one another. Then we will take all this knowledge and learn how it applies specifically to the problem of free buffer waits, taking a deep dive into Oracle architecture in the process.

If you want to learn how to quickly solve Oracle performance problems, come take advantage of a full day of training by Craig Shallahamer on August 18, at the CarrAmerica Conference Center in Pleasanton for $400. Contact **training_day@nocoug.org** *or register at* **www.nocoug.org**. *For more details, refer to page 26.*

Game Plan for Oracle Data Security

by Sam Alapati

Sam Alapati

This article is an abridged version of a white paper by Sam Alapati who has authored and co-authored numerous Oracle DBA books and Oracle Certification guides. Instructions for obtaining the full paper are provided at the end of the article.

Although most of us don't like to admit it, it's true that database security, while clearly understood by most database administrators as being a critical function, isn't something that's at the top of most CIO's and IT manager's lists. The truth is, managers are most concerned with database and application performance and uptime—they and their system and database admins don't spend sleepless nights worrying about security issues—rather, they agonize over the response times of their applications and the continuous availability of their systems. This prioritization of IT management makes sense, because application performance and availability are, indeed, the twin criteria on the basis of which management performance is measured. Are customers happy with the response time? Is the IT department able to get the reports out in time for management? Are we making sure failed batch jobs are fixed fast enough so they don't slow down work that depends on the output of the batched jobs? These are the questions swirling in all IT managers' minds when they walk into work in the morning.

In today's fast-changing technological environment, with virtualization, cloud computing, and service-oriented architecture (SOA) initiatives making enormous demands on their time and energy, managers are dimly aware of the need for security but manage to give short shrift to the security of customer data. By stating this fact, the author doesn't mean to impugn the motives of the IT administrators. On a day-to-day basis, it's your performance and availability that are deemed as the sine qua non by which the IT department is judged. Actually, the roots of this emphasis on performance and availability go deep, to the beginning of the application design and architecting of the systems. Security is always something that's considered important but accorded only a minimal importance. Most managers take the application of security firewalls and DMZs, password policies, SSH, and other secure protocols as an assurance that their systems are invulnerable to attacks.

The real world, of course, is far messier than any of us would like. Passive security measures such as firewalls, SSH, and similar protocols, while necessary, are certainly not sufficient to ensure that your data is safe. One of the key goals of this white paper is to help move IT managers, data security administrators, and database administrators (DBAs) to an action-oriented, proactive stance regarding data security and compliance. There are three distinct sections in this paper:

- **The 10 Security Questions Every Organization Must Ask:** Self-awareness is an essential precursor to change in a person's life. Similarly, your answers to the 10 key data security questions will enhance your awareness of your organization's exact security stance. At any given moment, your data is either secured or it isn't; you're either in compliance or you're not (compliance efforts offer compensatory controls—more about those later in this paper). The answers to the 10 questions show you where you stand in reality right at this moment.
- **The 7 Modern Data Security Mantras:** Once you get a good measure of your current security stance, the logical next step is to act on that understanding and strengthen your data security where it needs improvement. The 7 mantras or principles that the paper advocates are designed to create a security philosophy and maintain a strategic view of how you go about insuring security and compliance.

"Managers are most concerned with database and application performance and uptime—they and their system and database admins don't spend sleepless nights worrying about security issues—rather, they agonize over the response times of their applications and the continuous availability of their systems."

- **The Grocer's Dozen of Security Action Items:** Finally, we move on to the operational aspects of how to ensure a secure and compliant data environment. The items in this section are a set of action items—together, they'll put you on the road to really secure databases, applications, and data storage. Oh, by the way, did I say that you're also going to be beloved by both the internal and external auditors and regulatory enforcers?

The 10 security questions every organization must ask (or, First, know thy data)

With all the real-life data breaches occurring with clockwork-like regularity, it's amazing that most administrators don't really have a good handle on their data security barometer. Whistling in the dark as you walk past the graveyard is more like it. Did you ever lie awake at night asking yourself the question: "Holy mackerel: How do I know if I'm vulnerable?" To ensure that you really, really do understand your data vulnerability stance, and thus ensure that you get a healthy dose of sleep each night, I've devised a list of 10 questions that every organization and data administrator must ponder. It's probably hard for some of us to pose these questions, because some of the answers are bound to make us uncomfortable when we realize that various security-related stuff hasn't been properly addressed. But ask you must, because the answers to these questions, as I explain later, will give you a crystal clear idea as to exactly where you stand vis-à-vis data security and regulatory compliance.

Question 1: How vulnerable is my data?

At any given moment, vast amounts of corporate and customer data is floating around your organization, some in the form of active data that's being accessed by customers directly or summary data in warehouses being used by management to produce reports and to perform data mining. Do you know how vulnerable your data is? How do you know if your data is secure? Do you rely on a DBA's or system administrator's report that they're following all security best practices? What's the basis for your confidence (or lack of it)? Do you have any objective standards on which you're basing your assessment of data security?

Question 2: When was the last time we did a database vulnerability assessment?

A good database vulnerability assessment will provide a security health check of your system. The vulnerability assessment shows you your security weakness and where you need to tighten up security. The assessment will also unearth where you're falling behind regarding compliance with regulations such as Sarbanes-Oxley or PCIA. Most vulnerability assessments also perform a penetration test using a strong password-cracker tool to try to "crack" passwords, which are the first line of defense for any database.

Question 3: Do I have a security policy or a security plan?

If you don't have an explicit security policy or plan, it means that you're fighting the security and compliance battle with one hand (or both) tied behind your back. A strong plan will show your determination to win the security battle against the many foes, internal as well as external, that could breach your data security. The security plan will also force you to prioritize and pick solutions that provide the maximum return in higher security and compliance for your time and money.

"If you don't have an explicit security policy or plan, it means that you're fighting the security and compliance battle with one hand (or both) tied behind your back. A strong plan will show your determination to win the security battle against the many foes, internal as well as external, that could breach your data security."

Question 4: How soundly are my systems configured?

Most databases are installed and configured with performance and recoverability as the goal, not security. Installing a database, a web server, or an application well doesn't mean anything in regard to its security. Regular database vulnerability assessments ensure that the initial database, system, and application configuration, and the modifications that you make to your database and applications following the installation of those products, follow the recommended best practices, such as the removal of default passwords and other default settings.

The key thing to remember as far as configuration goes is that most of the settings are geared toward performance, not security; a good vulnerability assessment will unearth any configuration-related security holes.

Question 5: Am I following current security best practices?

You (or your company's security professionals) must know the current database security best practices. Oracle issues a set of security best practices for each of its database releases. Does your IT team use these security practices, many of which provide a first line of defense against attackers?

A thorough database vulnerability assessment will show you the extent to which you're following current database best practices. It will also uncover key things such as how well you've configured the database and app server installations, as well as whether your database has been sufficiently "hardened." The assessment shows whether your apps are using the default users and passwords, or users have excessive privileges, for example, as well as whether the operating system files have the correct protections applied to them. Probably the most basic step on your journey to a secure and compliant system is the hardening of your environment, including your databases—and only a formal database vulnerability assessment will show what you need to harden in your IT environment.

Question 6: Where is my data?

Production databases are such an important part of the daily life of IT managers and DBAs that almost invariably the development and test databases exist in a state of benign neglect. When folks are thinking about security, in most cases they are thinking about securing the production data, because that's deemed critical. While the production database performance and availability are indeed extremely critical to every organization, the accompanying neglect of key data in test and development databases is a big mistake—these databases contain the same customer data (including PII, SS #s, etc.) as the production databases.

As I explain later in this paper, use of features such as Oracle's Data Masking will protect your key data that is lying around in nonproduction databases. Considering that many firms now have their data stored in outsourced vendors' controlled databases, features such as Data Masking offer a quick and safe defensive approach. You can probably adopt a do-it-yourself approach to data masking, but not everyone has the programming time and resources to dedicate to this task.

Question 7: Who's using my data?

Sounds like a trivial question, correct? Quiz time! Do you know how many people are in your network right now, right at this minute, while you're reading this white paper? Do you know what they're doing in your system? Chances are, unless you're running a very small enterprise, you don't have a clue as to who all those users are that are logged into your system or what they're doing. Don't panic—an enterprise identity management solution can provide you with answers to these types of questions in no time.

Question 8: What data must I secure?

If you're like the majority of IT professionals, you most likely consider data security to be the security of the entire database. While backup strategies and availability strategies (think Oracle Data Guard) are designed for keeping data access available to your customers, the way you must approach data security is really different. In any database, the really important data that you must secure, both for preventing data breaches and to satisfy the compliance police, is but a small percentage of data—data such as personally identifiable information (PII), Social Security numbers, and so on.

Question 9: Who can change my data and who can approve the changes?

You must know which users have been formally granted privileges to change different types of data. One of the key requirements of most compliance regulations is the principle of segregation of duties (SOD), which is designed to protect the integrity of financial operations. By segregating duties according to job responsibilities and by not enabling one individual to perform cross-functional data access, you ensure that organizational controls such as financial controls can't be easily violated.

The answer to this question will reveal if you have the proper SOD policies in place—and also if you have any mechanism to automatically prevent SOD violations and report to you any attempts to violate SOD policies, including the recording and reporting of failed attempts to access sensitive, confidential, and restricted types of data.

One very effective way to control who can change data and who can grant those powers is to adopt a role-based access control (RBAC) policy. RBAC-based access polices provide access to users by basing it not on the individual user accounts but rather on the roles that are granted to those users. An identity-based solution such as Oracle's Role Manager can help you avoid the granting of conflicting access privileges, simplify the access privileges and activities certification process, and provide useful access management reports for auditors.

Question 10: Who has access to what, how, and why?

The answers to this question probably provide you with more information than the answers to any of the previous 9 questions. The reason is simple: the answers reveal your stance regarding how you're handling the key issues of authentication (the process of verifying that users who are seeking access are who they claim to be, and that they have the appropriate permission to use the data they're seeking), access, and the granting of privileges. A good vulnerability assessment of your databases and applications will also include a comprehensive entitlement report, which shows who the users of your system are and identifies the roles and privileges that they've been granted over time.

The vulnerability report will provide a good assessment of your user management policies. For example, all best-practice recommendations suggest that you not grant users direct privileges on direct objects but, rather, that you do so through the grant of roles. Almost all major regulations, such as SOX, for example, look for the use of role-based access management in your system. The security and vulnerability assessment will tell you if you're really using an RBAC-based access management.

Where do we go from here?

Your answers to the 10 questions posed in the previous sections show your company's current security and compliance stance. Before you move on to what you need to do to enhance your security, the paper proposes the adoption of a critical set of principles, to serve as guidelines for the actual steps you take to address security and compliance issues in your organization. Adherence to a set of principles while overseeing the implementation of security measures ensures that you are not operating in a vacuum but rather in an optimal fashion. You can download the rest of this paper, including *The 7 Modern Data Security Mantras* and *The Grocer's Dozen of Security Action Items*, at **www.miroconsulting.com/signup_miro_consulting.html**. ▲

Sam Alapati of Miro Consulting has 20 years of database management experience and has authored and co-authored numerous Oracle DBA books and certification guides. Miro Consulting helps companies analyze and negotiate enterprise software contracts—specifically Microsoft and Oracle licensing. In addition, the company offers software asset management and Oracle database security consultation services. Since 2000, the New Jersey-based company has negotiated over $1 billion in software licensing transactions and helped 400+ clients worldwide to optimize their total cost of ownership (TCO).

Making the Most of Oracle Exadata

by Marc Fielding

Marc Fielding

At Oracle's third-quarter earnings call, Larry Ellison announced that Oracle Exadata is "well on its way to being the most successful product launch in Oracle's 30-year history," with a sales pipeline approaching $1 billion for 2011. He attributed these sales to the game-changing performance of the Oracle Exadata platform. But what is the "secret sauce" behind these performance numbers? Read on to learn about the major performance features of Oracle Exadata, and discover tips on how to maximize performance from those features based on the author's own experience implementing Exadata.

Key Features

Smart scans: Smart scans are Exadata's headline feature. They provide three main benefits: reduced data transfer volumes from storage servers to databases, CPU savings on database servers as workload is transferred to storage servers, and improved buffer cache efficiency thanks to column projection. Smart scans use helper processes that function much like parallel query processes but run directly on the storage servers. Operations off-loadable through smart scans include the following:

- **Predicate filtering**—processing WHERE clause comparisons to literals, including logical operators and most SQL functions.
- **Column projection**—by looking at a query's SELECT clause, storage servers return only the columns requested, which is a big win for wide tables.
- **Joins**—storage servers can improve join performance by using Bloom filters to recognize rows matching join criteria during the table scan phase, avoiding most of the I/O and temporary space overhead involved in the join processing.
- **Data mining model scoring**—for users of Oracle Data Mining, scoring functions like PREDICT() can be evaluated on storage servers.

Storage indexes: Storage indexes reduce disk I/O volumes by tracking high and low values in memory for each 1-megabyte storage region. They can be used to give partition pruning benefits without requiring the partition key in the WHERE clause, as long as one of these columns is correlated with the partition key. For example, if a table has order_date and processed_date columns, is partitioned on order_date, and if orders are processed within 5 days of receipt, the storage server can track which processed_date values are included in each order partition, giving partition pruning for queries referring to either order_date or processed_date. Other data sets that are physically ordered on disk, such as incrementing keys, can also benefit.

Columnar compression: Hybrid columnar compression (HCC) introduces a new physical storage concept, the compression unit. By grouping many rows together in a compression unit, and by storing only unique values within each column, HCC provides storage savings in the range of 80–90% based on the compression level selected. Since data from full table scans remains compressed through I/O and buffer cache layers, disk savings translate to reduced I/O and buffer cache work as well. HCC does, however, introduce CPU and data modification overhead that will be discussed in the next section.

"By grouping many rows together in a compression unit, and by storing only unique values within each column, HCC provides storage savings in the range of 80–90% based on the compression level selected."

Flash cache: Exadata's flash cache supplements the database servers' buffer caches by providing a large cache of 384 GB per storage server and up to 5 TB in a full Oracle Exadata Database Machine, considerably larger than the capacity of memory caches. Unlike generic caches in traditional SAN storage, the flash cache understands database-level operations, preventing large non-repeated operations such as backups and large table scans from polluting the cache. Since flash storage is nonvolatile, it can cache synchronous writes, providing performance benefits to commit-intensive applications.

Hot/cold storage: The inherent geometry of rotating disks means that data is stored more densely in the outer portion of disk platters, giving higher throughput for disk operations on outer tracks and reducing the amount of time spent on head movement. Some Oracle systems currently leave inner tracks completely unused for this reason. Exadata allows the creation of separate "hot" ASM diskgroups for performance-critical data in outer disk regions and "cold" diskgroups for fast recovery area use in the inner regions.

I/O resource manager: Exadata's I/O Resource Manager (IORM) permits disk I/O operations to be prioritized on the storage cell in the same fashion as CPU time, and parallel query processes are currently managed by the Database Resource Manager (DBRM) on the database server. IORM is particularly useful when consolidating multiple database workloads together, by allowing I/O capacity to be allocated between different workloads according to their importance and configured limits.

Balanced hardware: The Exadata-powered Oracle Database Machine includes a fixed ratio of database nodes, storage servers, and associated networking equipment designed to avoid performance bottlenecks in any single part of the infrastructure.

Optimizing Performance

Exadata's performance features have been designed to work out of the box and require no manual configuration to use. That being said, a few small optimizations can greatly improve their effectiveness, and below are some battle-tested performance tips based on real-world Exadata deployments.

Use parallel query: Each Exadata storage cell contains 12 disks, which adds up to 168 disks in a full rack configuration. A key to efficient use of disk resources is to spread workload over these disks and particularly to avoid sequential operations that must wait until one disk operation completes before starting another. The ASM storage layer uses striping to distribute data evenly across physical disks, but it is up to the database instances to send I/O requests in such a way that the disks stay busy, which is where parallel query comes in. By splitting database operations into small chunks, parallel query can keep multiple physical drives busy, thus improving query response time.

The challenge in using parallel query is that the same optimizations that improve response time at slow periods can actually reduce scalability during periods of high demand, due to contention for the hard drives' fixed I/O capacity. Additionally, too many parallel processes can overwhelm system resources on database servers, causing additional performance degradation. To combat this problem, the Oracle database sets a limit on parallel query processes, PARALLEL_MAX_SERVERS. Once the system reaches PARALLEL_MAX_SERVERS, new requests run without parallelism at all, creating high variability in response time. The best way to avoid this situation is to avoid having it happen in the first place, using the Database Resource Manager to control maximum parallelism. Based on an analysis of expected concurrency and parallel query capacity (which is typically 128 and 256 processes on Exadata database nodes, depending on the mix of full-scan and more-CPU-intensive index operations), a resource manager plan can be constructed involving limits on both the number of parallel query processes per session and the total number of concurrent sessions.

While adding concurrency is of great benefit to large table scans, the overhead of managing parallel query processes can actually slow down scans of small tables. As a rule of thumb, if table operations take less than half a second to complete, parallel query delivers diminishing benefits, although this threshold can vary based on workload patterns. It is relatively simple to benchmark by testing common queries with varying degrees of parallelism, taking into account the system's total process capacity.

"The challenge in using parallel query is that the same optimizations that improve response time at slow periods can actually reduce scalability during periods of high demand, due to contention for the hard drives' fixed I/O capacity. Additionally, too many parallel processes can overwhelm system resources on database servers, causing additional performance degradation."

Learn to love the full scan: Index-based access paths work by generating a list of unique row identifiers from an index and then looking up the rows one by one from the source table. While very efficient for small lookups, the overhead of such sequential, random disk accesses increases nearly linearly as data size increases, making full scans more efficient for the large retrievals common in data warehouse workloads.

Since index access paths perform filtering at the database server rather than the storage server, Exadata's smart scan row filtering and join filters offer little benefit. Index-based table lookup performance further degrades with columnar compression, since even a single-row lookup requires reading an entire compression unit.

Good full-scan performance hinges on a good partition layout. Careful thought needs to be given to permit full partitions to closely match the data that users typically request.

Oracle provides several tools to help:

- A wide variety of partition types, including range, hash, list, interval, reference, or even virtual columns
- Composite subpartitioning with different partition and subpartition key columns, providing further benefits for queries involving multiple columns

- Storage indexes, giving the same benefits as partition pruning when data is clustered on disk

Although matching partition layout with query data sets provides maximum benefit of Exadata's features, it's rarely possible to match a partition layout to every possible query. In such cases, indexes are still required. Referential integrity continues to require indexes as well, although primary key indexes can be set to INVISIBLE (and therefore not be considered by the optimizer) if they prevent legitimate full partition scans.

Don't stop tuning applications: The advent of Exadata does not replace the need for application-level tuning. Poorly scaling application code will not suddenly become scalable when run on Exadata. The same tuning methods used in regular Oracle RAC databases will continue to work, so take advantage of the volumes of application-level tuning resources already available for Oracle RAC platforms.

"Businesses today are faced with ever-increasing user demands and data volumes. The combination of Exadata's feature set with a well-focused performance tuning effort can help address these challenges while benefiting from the results of 20 years of Oracle product development."

Spread disk groups across all available cell disks: Exadata's storage architecture makes it easy to implement SAME (Stripe and Mirror Everything) striping across all available drives. Thanks to the I/O Resource Manager, the performance guarantees previously only available by dedicating disks to specific applications can be obtained on a SAME layout. A SAME disk layout not only allows more efficient use of space, but it also gives better utilization of I/O capacity as well. For example, production applications can be guaranteed a certain percentage of I/O capacity—or even absolute priority when they require it—leaving otherwise available capacity for non-production applications.

Manage allocation units and extents carefully: Exadata storage cells are divided up into individual allocation units managed by ASM, analogous to stripes in a RAID configuration. Increasing ASM allocation unit size from the default 1 MB to 4 MB or even 8 MB improves performance of large reads by reducing the amount of seeking that drives need to do, along with reducing the overhead for managing allocation units themselves. The table-scanning benefits of larger allocation units only happen when they can be filled with a single table or index extent. To ensure that allocation units are not shared between multiple extents, set initial extents in large tables to be at least as large as the ASM allocation unit size. This process can be automated by using the CELL_PARTITION_LARGE_EXTENTS initialization parameter, which sets initial extents to 8 MB automatically for table partitions. Large 8 MB initial extents can be wasteful when storing very small objects, however. So for objects not expected to grow, smaller initial extents are still appropriate.

Reserve the flash cache for caching: Objects can be stored on flash disks permanently by creating flash-based grid disks. However, this type of storage layout usually hurts performance, since the built-in caching logic can do a better job of identifying which exact data blocks are the most frequently used and make intelligent caching decisions based on this information. Additionally, since true cache data does not need normal redundancy, a cache can store twice as much data in the same amount of storage as a grid disk.

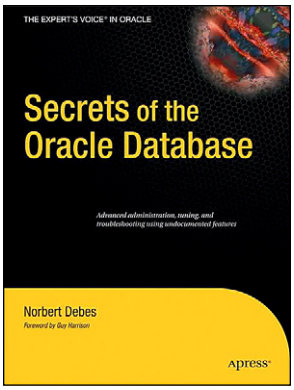
Consider Oracle Secure Backup: Oracle Secure Backup (OSB) runs on media servers directly connected to the Exadata InfiniBand fabric. In its current form, however, OSB has an important limitation: it can only back up to dedicated tape devices directly attached to OSB media servers. Neither disk-based backups nor third-party backup servers are supported. In environments that can accommodate this restriction, however, OSB eliminates network traffic on the database servers' Ethernet interfaces.

Use columnar compression judiciously: Hybrid columnar compression (HCC) in Exadata has the dual advantages of reducing storage usage and reducing I/O for large reads by storing data more densely. However, HCC works only when data is inserted using bulk operations. If non-compatible operations like single-row inserts or updates are attempted, Exadata reverts transparently to the less restrictive OLTP compression method, losing the compression benefits of HCC. When performing data modifications such as updates or deletes, the entire compression unit must be uncompressed and written in OLTP-compressed form, involving an additional disk I/O penalty as well. To avoid such overhead, consider compressing only data that infrequently changes, such as historical data. If partitioning data by date and data modification happens occasionally, a scripted automated process could periodically re-compress older partitions.

Conclusion

Businesses today are faced with ever-increasing user demands and data volumes. The combination of Exadata's feature set with a well-focused performance tuning effort can help address these challenges while benefiting from the results of 20 years of Oracle product development. ▲

Marc Fielding is a senior consultant with Pythian, a leading database infrastructure and application services company, where he specializes in high availability, scalability, and performance. Having worked with Oracle database products over the past nine years, his experience across the entire enterprise application stack allows him to provide reliable, scalable, fast, creative, and cost-effective solutions to Pythian's diverse client base. Read his blog posts at **www.pythian.com/news/tag/exadata** *or contact him at* **fielding@pythian.com**.



More Oracle Secrets

by Norbert Debes

Norbert Debes

A SERVERERROR Trigger is a feature that is very useful but rarely used. Oracle 8*i* included support for SERVERERROR Triggers. More functionality was added in Oracle 9*i*.

I frequently encounter applications that do not report Oracle ORA-nnnnn errors to the end user or at least provide only limited information on Oracle errors encountered. For example, an application might report no more than the Oracle error number. Many Oracle error message strings have placeholders for actual values that are filled in at the time of the error and provide additional information on the object that is related to an error (e.g., ORA-01918 user '%s' does not exist).

The Oracle DBMS also has support for retrieving a full error stack. Some applications report only the top-level error to the end user. Last but not least, users tend to have the annoying habit of reporting errors without remembering what the error was.

In all of these circumstances a SERVERERROR trigger is very handy and saves a lot of time tracking down issues.

Event Attribute Functions

More than two dozen so-called event attribute functions are available for retrieving all the details associated with a server error. Their names start with "ORA_." The manuals for the releases Oracle 9*i* and Oracle 10*g* document the functions in the "Application Developer's Guide – Fundamentals." Oracle 11*g* Release 2 documentation has them in the "PL/SQL Language Reference" manual. In my view the following two functions are among the most useful:

- ORA_IS_SERVER_ERROR
- ORA_SQL_TEXT

Note that all the "ORA_" functions work merely in the context of a server error trigger.

ORA_IS_SERVER_ERROR

The Boolean function ORA_IS_SERVER_ERROR checks whether a certain Oracle error number appears anywhere on the error stack. It returns true if the error is on the stack, false otherwise.

This function may be used to ignore certain errors such as syntax errors (e.g., ORA-00923: FROM keyword not found where expected).

ORA_SQL_TEXT

The function ORA_SQL_TEXT provides piecewise access to the SQL or PL/SQL statement that caused an error on the database server. The function returns the number of pieces that hold the entire statement text. The pieces are returned as a PL/SQL table in an OUT parameter of type ORA_NAME_LIST_T. The following code section contains a simple server error trigger that uses ORA_SQL_TXT to retrieve the text of the failed statement. It uses DBMS_SYSTEM to write to the process's trace file.

```
define schema=NDEBES
GRANT EXECUTE ON sys.dbms_system TO site_sys;
CREATE OR REPLACE TRIGGER site_sys.servererror_trigger AFTER
SERVERERROR
ON &schema..schema
DECLARE
   sql_text_tab ora_name_list_t;
   n binary_integer;
BEGIN
     sys.dbms_system.ksdwrt(1, 'SERVERERROR TRIGGER');
     sys.dbms_system.ksdwrt(1, sys.dbms_utility.format_error_stack);
     n := ora_sql_txt(sql_text_tab);
     for i IN 1..n loop
        sys.dbms_system.ksdwrt(1, sql_text_tab(i));
     end loop;
END;
/
```

DBMS_SYSTEM vs. UTL_FILE

I've been using a stored procedure that recorded server errors in the alert log with the undocumented package DBMS_SYSTEM for a number of years with good results. DBMS_SYSTEM.KSDWRT is a procedure that may be used to write to the alert log, a trace file or both. The downside of logging database clients' errors in the alert log is that the additional messages may clutter the alert log. It also makes finding errors for certain users tedious (at least with the output format that I used). Hence, I've recently rewritten the procedure based on UTL_FILE. This has the flexibility of writing to an arbitrary file and directory using a directory object. It's also possible to use separate files for each user or even for each session if need be. Obsolete files may be removed using find or rotated using logrotate (see **http://www.linuxconfig.org/Logrotate**, available for Linux, Solaris, AIX, and other platforms).

Architecture

A package called SERVERERROR_LOGGER is used to record information pertaining to server errors using UTL_FILE.

The database directory used with UTL_FILE is not hard coded. It is retrieved from a table called SERVERERROR_CONFIG. The package includes a thin wrapper around the package UTL_FILE.

A server error trigger does not record the error itself. Instead it calls the package SERVERERROR_LOGGER to perform the logging. As you will see in the next section, this reduces the trigger body to just three lines.

The procedure LOG_SERVERERROR retrieves information pertaining to a server error and saves it using the procedures FOPEN, PUT, and FCLOSE. These latter procedures serve the same purpose as their counterparts in the package UTL_FILE. They open, write to, and close a file respectively.

Installation

This section explains how to install the components that are required for recording server errors.

➤ Create a separate file system directory (optional)

I recommend using a separate directory for server error log files.

```
$ mkdir /oracle/ORCL/admin/servererror_log
```

➤ Create a database directory

Next, create a database directory object that maps to a file system directory for server error log files.

```
SYS@ORCL> CREATE DIRECTORY servererror_log AS '/oracle/ORCL/
admin/servererror_log';
```

➤ Run the installation script servererror_trigger_utl_file.sql

I put all my site-specific customizations into the schema SITE_SYS (inspired by Perl's site_perl directory). The script servererror_trigger_utl_file.sql uses two SQL*Plus define variables. The variable INSTALLATION_SCHEMA is for setting the target schema for database objects created by the script. The variable DIRECTORY_NAME is for setting the database directory name that will be used for creating server error log files. Both variables may be changed to suit your needs. The directory name is stored in a table called SERVERERROR_CONFIG and is retrieved once during package initialization.

```
define installation_schema=SITE_SYS
define directory_name=SERVERERROR_LOG
```

If the target schema does not exist, it is created with the following command in the script:

```
CREATE USER &installation_schema IDENTIFIED BY VALUES 'LOCKED';
```

This creates the user with the impossible password hash "LOCKED" containing non-hexadecimal digits. Hence, the schema is effectively locked.

The script must be run as SYS, since it includes several grants on objects owned by SYS.

```
SYS@ORCL> @servererror_trigger_utl_file.sql
```

➤ Create a server error trigger

The following code creates a server error trigger for an individual schema. The trigger is called whenever a server error occurs in a session of the database user by the same name as the schema. Server error triggers may also be created for the entire database (ON DATABASE).

```
SYS@ORCL> define a_schema_name=&schema_name
Enter value for schema_name: ndebes
SYS@ORCL> create or replace trigger &installation_schema..servererror_
&a_schema_name after servererror
on
&a_schema_name..schema
begin
   &installation_schema..servererror_logger.log_servererror(
      user || '_servererror.log'
   );
end;
/
```

➤ Test the server error trigger

To test the trigger, provoke an error, e.g., by using an incorrect column name in a SELECT statement as shown below:

```
SYS@ORCL> connect ndebes
Enter password:
Connected.
NDEBES@ORCL> SELECT wrong_column_name FROM dual;
SELECT wrong_column_name FROM dual
       *
ERROR at line 1:
ORA-00904: "WRONG_COLUMN_NAME": invalid identifier
```

The error caused the server error trigger to fire. The trigger called the package SERVERERROR_LOGGER, which in turn created a log file in the directory used earlier.

The contents of the log file are reproduced below:

```
$ cat /oracle/ORCL/admin/servererror_log/NDEBES_servererror.log
Mon Mar 22 17:45:00.497874000
ORA-20000: SERVERERROR Trigger fired in session with SID=501. See
below.
ORA-00904: "WRONG_COLUMN_NAME": invalid identifier
SQL statement for SID=501 broken up into 1 rows
SELECT wrong_column_name FROM dual
SID: 501
Serial#: 44
Audsid: 19204661
Username: NDEBES
OS User: oracle
Machine: dbserver
Client Process: 27086
Program: sqlplus@dbserver (TNS V1-V3)
Server: DEDICATED
Module: SQL*Plus
Action:
Client Info:
Logon Time: 22-Mar-2010 17:44:37
```

Some of the information shown above is retrieved from V$SESSION. The source code uses USERENV('SID') to retrieve the session ID. Thus the code will work with Oracle 10*g* Release 1 or newer versions but not with Oracle 9*i*. (Replace with "SELECT sid FROM v$mystat WHERE rownum=1" if you need an implementation that works with Oracle 9*i*.)

Configuring Log File Rotation

Logrotate is an open-source utility for log file rotation and compression. Log files may be rotated on a yearly, monthly, weekly, or daily basis. The frequency of log rotations and other settings is configured using a configuration file that is normally called "logrotate.conf." Logrotate uses a state file to keep track of files. The details are beyond the scope of this article. Please refer to the manual page that is available at the URL **http://www.linuxconfig.org/Logrotate**.

The following logrotate configuration file sets up weekly log rotation for server error log files:

```
# file: logrotate.conf
# set defaults
compress
missingok

# rotate server error log files
/oracle/ORCL/admin/servererror_log/*.log
{
    weekly
    rotate 3
    copytruncate
}
```

Use the following command to test log file rotation (you would normally use absolute path names instead of relative ones in the example below):

```
$ /usr/local/sbin/logrotate -f -s logrotate.state logrotate.conf
```

The switch "-f" forces log file rotation even though log files are not yet due for rotation based on the information in the state file (logrotate.state in the above example). Do not use "-f" in an actual job that rotates log files based on the interval defined in logrotate.conf.

When using the debug switch "-d," keep in mind that logrotate does not rotate any log files if the debug switch "-d" is used.

After running logrotate three times with "-f" you will see three compressed and rotated copies of server error log files:

```
$ ls -tr NDEBES_servererror.log*|less
NDEBES_servererror.log.3.gz
NDEBES_servererror.log.2.gz
NDEBES_servererror.log.1.gz
NDEBES_servererror.log
```

At this point you may delete the switch "-f" and integrate the logrotate command into a job scheduler such as cron.

Performance Implications

In order to assess the performance degradation due to a server error trigger, I ran a conventional import (imp) that imported 100,000 rows into a table and failed with ORA-00001: unique constraint (%s.%s) violated on each INSERT. Hence, the server error trigger fired 100,000 times. A single test run consisted of ten import runs and was repeated three times: once without a server error trigger, once with the trigger in place but disabled, and once with an enabled trigger. The results were measured with the UNIX utility time. The following table shows the results. The response time is reported in seconds.

	Import Response Time
No Trigger	118 s
Disabled Trigger	112 s
Enabled Trigger	256 s

As you can tell from the table, the performance degradation is severe in this extreme case where each INSERT causes the trigger to fire. The response time more than doubled. The enabled server error trigger prolonged the response time by 144 seconds versus the disabled trigger. Thus, a single execution of the trigger took about 1.4 ms. The performance of an application that causes a few errors per hour would not be affected by a server error trigger. The fact that the test without a trigger was 5% slower than with a disabled trigger is due to normal fluctuations in response time on a timesharing system.

Download

An SQL script for configuring server error triggers is available at **www.nocoug.org/Journal/servererror_trigger_utl_file.txt**.

Summary

In this edition, the benefits of using server error triggers for troubleshooting were discussed, and a working example of a server error trigger was presented. Rotating server error log files using logrotate ensures that the log files do not grow indefinitely and that past errors are retained for a configurable period.

The performance degradation of using a server error trigger is significant only if errors occur at a very high frequency in the order of multiple errors per second. Under normal circumstances, no performance degradation will be noticeable. ▲

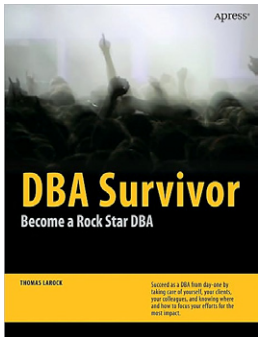
Norbert Debes has more than 13 years experience as an Oracle database administrator. For over 6 years, he held different positions and technical roles at Oracle Germany. He is the author of Secrets of the Oracle Database *(Apress, 2009).*

DBA Survivor Become a Rock Star DBA

A Book Review by Dave Abercrombie

Details

Author: Thomas LaRock
Pages: 250
Publication Date: March 2010
Price: $39.99 (list)
Publisher: Apress



Summary

Overall review: This book is about working effectively as a DBA. It does not address technical details and does not cover Oracle at all. The book focuses on communication and collaboration from the perspective of a newly hired DBA. This perspective is fresh and insightful, and provides general lessons that can benefit even well-seasoned DBAs.

Target audience: Junior DBAs.

Is this book platform specific? Any technical details are limited to Microsoft SQL server; Oracle is not mentioned. However, the book is mostly nontechnical.

Why did I obtain this book? I am convinced that the biggest issues that confront database professionals are not technical but instead are related to communication. If you cannot communicate your idea, then it doesn't really matter how clever it is. Also, if your own mind is closed to new ideas from others, then you will not be able to truly engage with your work. I was intrigued about a book for DBAs that shares my respect for the need for good communication.

Overall Review

Why should an Oracle DBA be interested in this book when its few technical examples are all for Microsoft SQL Server, not Oracle? Because LaRock offers sage advice for the significant nontechnical parts of our jobs. His focus on communication, business protocols, career growth, and selling yourself is unique and refreshing.

Although LaRock's target audience is someone just starting a DBA career, his advice provides general benefits, and it can be especially valuable for those who hire DBAs. LaRock's chapter on starting a new job was especially intriguing: I'll bet that many of us have yet to do all of the things it suggests for the first week on a job! I found that chapter so helpful that I shared it with friends who were starting non-DBA jobs!

Preparation

LaRock emphasizes training. Since training includes attending conferences and reading journals, you, dear reader, are already following some of his advice! LaRock explains that *"if you really want to learn a topic or subject, then try to teach it to someone else."* I strongly agree. He closes his chapter on preparation by advising, *"Keep learning, keep writing, keep speaking, keep meeting new people, and keep your eye on the prize."*

Starting a Job, Initial Checklist

As a new DBA, one of your first steps is to prepare a list of databases for which you are responsible. Obvious next steps include verifying that backups are running and that databases can be restored. Not so obvious is to prepare a list of "customers." Who within your company uses which databases, and which executive managers depend upon these databases? This knowledge helps you to communicate about plans and problems. More importantly, reaching out to these people builds your relationships with them. When you detect a problem, you can let these internal customers know that you are working on it before they find out about it through complaints. This proactive outreach goes a long way toward strengthening working relationships, benefiting external customers and improving the likelihood of company success.

LaRock suggests that you also make special efforts to work with your developers. *"Find out what projects they are working*

I am convinced that the biggest issues that confront database professionals are not technical but instead are related to communication. If you cannot communicate your idea, then it doesn't really matter how clever it is. Also, if your own mind is closed to new ideas from others, then you will not be able to truly engage with your work."

on, and what roadblocks they are currently facing, and see if there is an opportunity to offer assistance." The developers are often the ones pushing the limits of database technology. Unfortunately, this push is often caused by ignorance, pushing in the wrong way toward scalability limits. LaRock's response to this common problem is refreshing: *"The end result will be an abundance of little training opportunities for you to take advantage of."* Cultivate an attitude where you too can look at these not as annoying problems created by people who should know better, but instead as little training opportunities!

LaRock also suggests that you reach out to the operating system administrators. He points out that the DBA plays a vital role as "the universal translator" between developers and system administrators. Developers know the code, are close to business needs, and work daily with customer requirements. System administrators spend their time racking servers, installing routers, and managing storage. LaRock explains that *"unfortunately these two groups rarely interface unless there is a problem (or an opportunity to point fingers at each other). That's where you come in."* As a DBA you can speak both languages, so *"by meeting with both groups, you continue to brush up on your language skills and help to keep the peace, and ultimately you can help to reduce costs."*

LaRock reminds us of some common problems with monitoring and alerting systems. Alerts can be cryptic. Some alerts are very serious, yet some are likely false alarms. Seasoned employees often get a feel for alerting systems, usually doing a decent job of guessing the severity level. As a new DBA, you are in a unique position to question the alerting system and to encourage its improvement. LaRock shares his basic rule about alerts: *"If you do not know what they are for, then you need to find out why you are getting them."* Perhaps experienced and jaded DBAs can also adapt the perspective of a new DBA by critically examining all alerts.

LaRock concludes his chapter on starting a new DBA job with advice rarely seen in database books. He suggests that we join professional organizations, something already well known to the NoCOUG readers of this review! He returns to the theme of communication and collaboration. He advises us to enjoy lunch with our colleagues. He advises us to stay clear of office politics, which sounds easier than it is. *"Be open and straightforward." "Take care of yourself." "Develop your people skills." "Volunteer in a community or church organization."* Again, this is good advice for seasoned DBAs as well as new ones.

Basics

LaRock moves into a nice, high-level summary of several basic technologies related to databases. His brief summary of RAID and SAN technology is well illustrated and clear. His basic explanations can be used to bridge the knowledge gap between experts and those who have only a vague grasp of these concepts. He touches on, but does not attempt to resolve, controversies over RAID 5. His discussion of disk performance benchmarking will strike some readers as superficial. For example, he does not distinguish between metrics like maximum sustained sequential-read bandwidth and random-read I/O operations per second. LaRock does not dive into CPU issues at all.

"Why should an Oracle DBA be interested in this book when its few technical examples are all for Microsoft SQL Server, not Oracle? Because LaRock offers sage advice for the significant nontechnical parts of our jobs. His focus on communication, business protocols, career growth, and selling yourself is unique and refreshing."

LaRock summarizes High Availability (HA) options for databases. He warns that "as the DBA, you are going to be held responsible for making sure your databases are always available, even if you have nothing to do with the apparent outage." LaRock's HA summary is very brief, giving only an outline without any real technical detail. For example, his discussion of clustering (e.g., Oracle's RAC) is only about a page and a half long. LaRock clearly distinguishes HA from Disaster Recovery (DR). As with HA, his DR overview is very brief, conceptual without practical details. LaRock is walking a fine line here. His primary audience is the junior DBA, who needs to know DR and HA concepts but who is probably not responsible for developing and implementing DR and HA plans. This section is therefore a mix of superficial summary and specific advice.

Similar limitations are present in his discussion of networks. He humorously, but truthfully suggests that "as a DBA, you may sometimes find yourself blaming the network simply because you cannot blame bad code all the time." However, LaRock offers very little practical network advice. His discussion is based purely on a simple plumbing analogy. Unfortunately, this analogy provides very little insight here. For example, he writes, *"When it comes time to build a network, you want to research how pipes work and what kind of connectors to use, strike a balance between reliability and affordability, and make sure you never have to touch the equipment again."* LaRock concludes this discussion with some real-world warnings, no doubt learned the hard way. He points out how *"people tend to blame things they do not understand,"* which is often the database. He describes how you get the most recognition by solving emergencies, yet this can actually be counterproductive. It can diminish the effort devoted to preventing problems proactively, since such effort may go unrecognized. To counter this dynamic, LaRock gives the excellent advice to develop metrics that can demonstrate your progress to managers.

Development Environment

LaRock points out that *"a development server is a production server to a developer."* Perhaps this is obvious, but it is easy for a

(continued on page 21)

Automatic SGA Resizing: Is It Right For You?

by Chris Lawson

Chris Lawson

The Oracle database has become very sophisticated in recent years, as more and more new features are added. These new features, however, are occasionally accompanied by unanticipated (and unwanted) side effects. Here's a story of one new 10g feature run amok. For purposes of confidentiality, I will refer to the company as "Greg's Golf Supplies."

The Symptoms

For unknown reasons, users at Greg's Golf would see application response time unpredictably increase 10-fold, from about ½ second to 5 seconds. This usually happened during busy times, but not always. We were able to detect this response blip because the company had sophisticated response time tools that kept track of the longest response time for any user during a 5-minute period. So, even though the averages were fine, the firm's monitoring tools detected these rare response time spikes. These occasional spikes really were important for business and legal reasons. Remember, the company in this study isn't really a golf store.

What made this problem especially hard to troubleshoot was its rare appearance. Compared to the huge volume of transactions in the database, these spikes were rare—only about 1% of all transactions. Thus, the cumulative runtimes and other statistics, as reported by AWR reports, looked fine.

Since we had already detected a few improper queries in the application (e.g., scanning more data than necessary), we tended to assume that the application was the source of the response time blips.

"The Oracle database has become very sophisticated in recent years, as more and more new features are added. These new features, however, are occasionally accompanied by unanticipated (and unwanted) side effects."

Let the Guessing Begin

A few analysts began focusing on the application servers, believing them to be the cause of the mischief. I pointed out that that could indeed be the case, but without actual evidence pointing in that direction, it was just a guess. (I am always reluctant to go down paths based on hunches that are not supported by empirical facts.)

ASH to the Rescue

Using Active Session History (ASH), I was able to capture the "bad" SQL, but it looked the same as the other "good" SQL. In fact the "bad" SQL was a very trivial query—simply getting the next value for a sequence.

I spent hours analyzing the various sequences to no avail. Cache sizes were similar. Finally, I probed a bit deeper with my ASH scripts. I looked to see if the long-running SQL were waiting on another session. Then, I checked to see what was obstructing the other session. Here's an example of one script that I used to show the active sessions for a particular time:

```
set sqlblanklines on

WITH

p1 AS
(
  SELECT DISTINCT
    sample_time,
    event,
    time_waited,
    session_id,
    sql_text,
    p1text,
    p1,
    p2text,
    p2,
    p3text,
    p3,
    current_obj#,
    current_file#,
    current_block#,
    blocking_session

  FROM
    dba_hist_active_sess_history a,
    v$sqltext b
```

```
  WHERE a.sql_id = b.sql_id
  AND sample_time LIKE '17-Apr-08 09.56.3%'
  AND piece = 0
  AND time_waited / 1000000 > 2
)

SELECT
  sample_time,
  blocking_session,
  event,
  time_waited / 1000000 sec,
  p1text,
  p1,
  p2text,
  p2,
  p3text,
  p3

FROM
  p1

ORDER BY
  sample_time;
```

A Confusing Discovery

The results were confusing. I discovered that there was usually another session blocking my simple SQL (the sequence next-value code), but this other session was waiting on "SGA: allocation forcing component growth."

> ***"New features are usually helpful, but not always. It pays to be suspicious of hyped tools or algorithms that magically eliminate the need for DBA analysis."***

What in the world was that? Why should a session ever be waiting on SGA allocation? I examined SGA resizing activity by querying the view V$SGA_RESIZE_OPS.

```
set sqlblanklines on
col when format a25
col component format a25

SELECT
  TO_CHAR (start_time, 'Mon-Dd:Hh24:Mi:Ss') when,
  component,
  oper_type,
  initial_size,
  final_size

FROM
  v$sga_resize_ops;
```

I was surprised by what I found out. It turns out that we were constantly changing the SGA up and down—sometimes many times per second. In one example of the "thrashing" we saw in production, Oracle did a grow on the shared pool three times and a shrink all within 1 second! This is obviously not how this feature is supposed to work.

> ***"In 98% of performance fixes, the root cause is the application—not the database. When was the last time you solved a performance problem by changing an init.ora parameter?"***

I checked another production system and confirmed that the resizing should be infrequent. After a bit more investigation, I found other DBAs who experienced similar problems on high-transaction systems. Additionally, a Metalink note admitted that it may be wise to turn the feature off. (Editor's Note: Metalink indicates that the issue is fixed in Oracle Database 10.2.0.5 and 11.2.)

The production DBAs eagerly turned off the SGA automation by making a change in the init.ora file. We set SGA_Target = 0. With the automatic SGA feature restrained, the maximum application response time greatly improved—both in absolute value and in consistency.

A Nice Side Benefit

After the fix was implemented, we also rid ourselves of the odd wait event, "cursor: pin S wait on X." Originally, this wait event was often in the top-5 waits in the AWR reports. After the fix, the mystery wait event was no longer a major event. It was still there, but far, far down the list, with a total of just 2 seconds!

Conclusion

Note that this performance improvement had absolutely nothing to do with bad application code. In 98% of most performance fixes, the root cause is the application—not the database. When was the last time you solved a performance problem by changing an init.ora parameter?

New features are usually helpful, but not always. It pays to be suspicious of hyped tools or algorithms that magically eliminate the need for DBA analysis.

Special credit goes to Jonathan Lewis for his work exposing the problems with the SGA automatic resizing feature. He has written several articles on his experience with this feature. See **jonathanlewis.wordpress.com/2007/04/16/sga-resizing/**. ▲

Chris Lawson is an Oracle performance consultant and Oracle Ace who lives with his family in Dublin, California. He is the author of Snappy Interviews: 100 Questions to Ask Oracle DBAs *and* The Art & Science of Oracle Performance Tuning, *both available on Amazon. Chris' website is* **http://www.OracleMagician.com**. *In his spare time, Chris enjoys golf, choral singing (bass), and throwing frisbees to Morgan, the resident border collie.*

SQL 101: Which Query Is Better?

by Iggy Fernandez

Iggy Fernandez

I was asked the following question (paraphrased) during a job interview: In general, there are lots of ways of expressing a particular query requirement in SQL with implications for query performance. For example, which departments have employees with salaries greater than a certain cutoff? Here are two ways to express this query requirement in SQL. The first uses a conventional correlated subquery, while the second uses ANSI join syntax. Which is better? From a theoretical perspective? From a practical perspective? In certain situations? Since I can only pick one, which one should I pick?

```
VARIABLE salary_cutoff NUMBER

-- Correlated subquery

SELECT d.department_name
FROM departments d
WHERE EXISTS (
  SELECT *
  FROM employees e
  WHERE salary > :salary_cutoff
  AND department_id = d.department_id
);

-- ANSI join syntax

SELECT d.department_name
FROM (
  SELECT DISTINCT department_id
  FROM employees
  WHERE salary > :salary_cutoff
) e
JOIN departments d ON e.department_id = d.department_id;
```

My interviewer's opinion was that the best choice depends on the salary cutoff. He argued that if the salary cutoff is high, few employees will qualify and therefore the Employees table should be used as the driving table; that is, the second version of the query (the ANSI Join version) is the best choice when the salary cutoff is high. Conversely, if the salary cutoff is low, many employees will qualify and therefore it would be inadvisable to use the Employees table as the driving table; that is, the first version of the query (the Correlated Subquery version) is the best choice when the salary cutoff is low.

The above reasoning is incorrect because it assumes that there is only one choice of driving table for each version of the query. This assumption is based on an intuitive interpretation of each version. An intuitive interpretation of the Correlated Subquery version is that the Departments table is the driving table because the first words of the query are "SELECT d.department_name FROM departments." Similarly, an intuitive interpretation of the ANSI Join version is that the Employees table is the driving table because it is the first table encountered in the query. However—as the following demonstrations prove—the choice of driving table is not constrained. Hints are used in the demonstrations to alter the driving table at will. This effectively proves that the Oracle query optimizer can choose either table as the driving table. The demonstrations used Oracle Database 11*g* Release 2.

Demonstrations

The first demonstration uses the LEADING and HASH_SJ hints in order to make the Departments table the driving table for the Correlated Subquery version of the query. Note that Oracle uses a Semijoin method (Hash Join Semi) to join the Departments table and the Employees table.

```
exec :salary_cutoff := 0;

SELECT
  /*+ QB_NAME(main) LEADING(d@main) */
  d.department_name
FROM hr.departments d
WHERE EXISTS (
  SELECT /*+ QB_NAME(sub) HASH_SJ */ *
  FROM hr.employees e
  WHERE e.salary > :salary_cutoff
  AND e.department_id = d.department_id
);
```

Operation	Name	Rows
SELECT STATEMENT		
HASH JOIN SEMI		10
TABLE ACCESS FULL	DEPARTMENTS	27
TABLE ACCESS FULL	EMPLOYEES	107

The second demonstration uses the LEADING and USE_HASH hints in order to make the Employees table the driving table for the Correlated Subquery version of the query. Note that Oracle uses a regular Join method (Hash Join) in this case.

```
exec :salary_cutoff := 0;

SELECT
  /*+ QB_NAME(main) LEADING(e@sub) USE_HASH(d@main) */
  d.department_name
FROM hr.departments d
WHERE EXISTS (
  SELECT /*+ QB_NAME(sub) */ *
  FROM hr.employees e
  WHERE e.salary > :salary_cutoff
  AND e.department_id = d.department_id
);
```

Operation	Name	Rows
HASH JOIN		11
SORT UNIQUE		107
TABLE ACCESS FULL	EMPLOYEES	107
TABLE ACCESS FULL	DEPARTMENTS	27

The third demonstration also uses the LEADING and USE_HASH hints in order to make the Employees table the driving table for the ANSI Join version of the query. The query plan is almost identical to the previous query plan; the only difference is the use of the Hash Unique method instead of the Sort Unique method for sorting the contents of the Employees table. Note that the Oracle query optimizer's estimate (107) of the number of the rows remaining after the Sort Unique operation in the previous query plan is obviously incorrect because it does not differ from the estimate of the number of rows (107) in the Employees table. The optimizer makes a better estimate (11) of the number of rows remaining after the Hash Unique operation in the following query plan.

```
exec :salary_cutoff := 0;

SELECT /*+ LEADING(e d) USE_HASH(d) */ d.department_name
FROM (
  SELECT DISTINCT department_id
  FROM hr.employees
  WHERE salary > :salary_cutoff
) e
JOIN hr.departments d ON e.department_id = d.department_id;
```

Operation	Name	Rows
SELECT STATEMENT		
HASH JOIN		11
VIEW		11
HASH UNIQUE		11
TABLE ACCESS FULL	EMPLOYEES	107
TABLE ACCESS FULL	DEPARTMENTS	27

The fourth demonstration also uses the LEADING and USE_HASH hints in order to make the Departments table the driving table for the ANSI Join version of the query. This query plan differs significantly from the query plan in the first demonstration because it requires a sort operation and a regular Join instead of a Semijoin.

```
exec :salary_cutoff := 0;

SELECT /*+ LEADING(d e) USE_HASH(e) */ d.department_name
FROM (
  SELECT DISTINCT department_id
  FROM hr.employees
  WHERE salary > :salary_cutoff
) e
JOIN hr.departments d ON e.department_id = d.department_id;
```

Operation	Name	Rows
SELECT STATEMENT		
HASH JOIN		11
TABLE ACCESS FULL	DEPARTMENTS	27
VIEW		11
HASH UNIQUE		11
TABLE ACCESS FULL	EMPLOYEES	107

The above demonstrations prove that there is no constraint on the choice of driving table in either version of the query. However, the ANSI Join version requires an unnecessary sort operation if the Department table is chosen as the driving table. Query optimization is a very complex problem, and it appears that Oracle is not currently capable of utilizing the Semijoin method to process the ANSI Join version of the query. This significantly reduces the number of options considered by the query optimizer when processing the ANSI Join version. For example, the following query plan can be used with the Correlated Subquery version of the query but not with the ANSI Join version, because it uses another variant of the Semijoin method (Nested Loops Semi).

```
exec :salary_cutoff := 0;

SELECT
  /*+ QB_NAME(main) LEADING(d@main) USE_NL(e@sub) */
  d.department_name
FROM hr.departments d
WHERE EXISTS (
  SELECT /*+ QB_NAME(sub) */ *
  FROM hr.employees e
  WHERE e.salary > :salary_cutoff
  AND e.department_id = d.department_id
);
```

Operation	Name	Rows
SELECT STATEMENT		
NESTED LOOPS SEMI		10
TABLE ACCESS FULL	DEPARTMENTS	27
TABLE ACCESS BY INDEX	EMPLOYEES	41
INDEX RANGE SCAN	EMP_DEPARTMENT_IX	1

Conclusion

The above demonstrations indicate that the Correlated Subquery version of the query is the better choice.

Note that the Correlated Subquery version of the query is based on Relational Calculus, while the ANSI Join version is based on Relational Algebra. The founder of relational database theory, Edgar Codd, predicted that queries based on Relational Calculus would be easier to optimize. In his words: *"The relational calculus permits a user to request the data he desires by its properties. This is an ideal starting point for search optimization. The algebra, on the other hand, requires the user to formulate a sequence of algebraic operations that will generate the desired data from the data base relations. For queries other than very simple ones, the properties of the desired data tend to get hidden in the particular operation sequence (one of many possible ones) which the user selects. Therefore, starting from an algebraic source language, one has the choice of locally optimizing the execution of each operation (a very limited form of optimization) or tackling the difficult problem of analyzing sequences of such operations to discover the intended defining properties of the desired data."* (Relational Completeness of Database Sublanguages, 1972). ▲

Oracle on VM?

Ask the Oracles!

Bert Scalzo: Servers everywhere are being virtualized. Yet many DBAs are hesitant to embrace virtualized database servers. I've been mystified by this anomaly, so I've asked those opposed for their rationale. While there are numerous arguments against, basically two pervasive themes surface from among all the replies.

First and foremost, DBAs almost universally claim that their databases are "mission critical" and cannot suffer any performance hit that virtualization would necessarily impose. I hypothesize that these people must either consider shared resources as negative or they have read that virtualization overhead can be from 5% to 15%—and they cannot suffer that loss.

However those very same DBAs quickly allowed the single most important database performance factor (Disk I/O) to become shared well over a decade ago. We all quickly embraced new Storage Area Network (SAN) disk arrays in order to get large pools of storage. Yet very few of those SANs were dedicated to a single database or even a single DBA's multiple databases. SANs were generally shared resources—and often the DBA wasn't fully aware of who was sharing their spindles. We simply asked for "black box" amounts of space that were assigned for our use as LUNs.

Today we're simply permitting the three remaining key components (CPU, memory, and networking) to be shared like our storage. If we accepted it for Disk I/O back then, how can we now say that the much less important database performance factors cannot be shared? I believe it's just resistance to change.

As for the virtualization overhead, it's not a factor. If we were simply going to virtualize the database server and place it back on the same physical server, then sure—there would be a slight performance reduction. However DBAs generally order excess capacity for growth; thus, most servers are idle more than 50% of the time overall. But most virtualization efforts are to replace smaller servers with much larger shared ones. So losing my four quad-CPU and 16GB RAM physical server and then being allocated the same or more resources from a much larger shared server should be a nonissue. As long as there isn't overallocation of resources on the physical virtual servers (i.e., hosts), then the negative performance impact should range from minimal to nonexistent. Thus, if four-quad CPU and 16GB database servers were rehosted to a virtualized host that had 32 CPUs and 128GB of memory, the performance could actually be better (or, at worst, about the same).

The second pervasive but veiled theme is one regarding "loss of control." You're not going to like this observation nor be happy with me for making it. But in the good old days, the DBA was a god. We often had unfettered access to our hardware platform. It was not uncommon to have "root" access. We often performed numerous complex design and management tasks, including hardware platform research and ordering, operating system configuration and tuning, storage design and allocation, capacity monitoring and projections, and so on. Thus the DBA knowledge and responsibilities were Herculean—and we loved it that way.

But in a virtualized world, now the DBA simply treats everything as a "black box" that someone else both provides and manages. We cannot venture into the server room anymore and knowingly point to our static resources such as disks. Nor can we really know exactly where our virtual machine is being hosted, because it can move—sometimes dynamically. Plus we have to ask someone else for things we used to do for ourselves. It's a bit unnerving for those who remember the good old days.

Yes—there are some very valid performance issues that must be addressed when you virtualize your database, and those cannot be left to the defaults or chance. But most people seem to object more in the abstract. You're not going to stop virtualization—so you might as well learn to embrace and even like it. ▲

Bert Scalzo is a database expert for Quest Software and a member of the TOAD dev team. He has worked with Oracle databases for well over two decades. Mr. Scalzo's work history includes time at both Oracle Education and Oracle Consulting. He holds several Oracle Masters' certifications and has an extensive academic background—including a BS, MS, and PhD in computer science; an MBA; and insurance industry designations. Mr. Scalzo is also an Oracle ACE who has presented at numerous Oracle conferences and user groups.

"Today we're simply permitting the three remaining key components (CPU, memory, and networking) to be shared like our storage. If we accepted it for Disk I/O back then, how can we now say that the much less important database performance factors cannot be shared?"

(continued from page 15)

DBA to forget. He does an excellent job of describing the pressures that developers operate under. The demand for new code and enhancements is always present. In addition, developers are often on the front lines of production support, another fact that is easy to forget. LaRock provides many useful tips for increasing your empathy for developers and for working more effectively with them. He points out that Service Level Agreements (SLAs) are essential, providing a framework for expectations.

Good communication is key to managing expectations and working effectively with developers. His anecdotes about poor communication are insightful. Isn't it curious how communication about technical topics is often easier with nontechnical staff? Perhaps technical jargon interferes too much, making you think you have communicated something while hiding the fact that it was misinterpreted. Key things to consider include clarity about schedules, admitting mistakes and ignorance, and understanding your limits.

Troubleshooting

Oracle DBAs will skip this section on Microsoft SQL Server troubleshooting. However, if you choose to read it, see if you can avoid a smug sense of superiority about how much better Oracle diagnostics are compared to Microsoft's.

Eating, Training, Sharing

One of LaRock's last chapters is about eating well, exercising regularly, and relaxing. This is perhaps the first DBA book to address these topics, and in my view, this is indeed welcome and necessary. He concludes with helpful chapters that expand on his previous themes of training and sharing.

Conclusion

This book is about working effectively as a DBA. It does not address technical details and does not cover Oracle at all. The book focuses on communication and collaboration from the perspective of a newly hired DBA. This perspective is fresh and insightful, and provides general lessons that can benefit even well-seasoned DBAs. ▲

Dave Abercrombie has worked at Convio (with a "v," not an "f") for about ten years, having helped to found GetActive Software before its merger with Convio. This company's business model is rather like a distributed denial of service attack against itself. Its customers are nonprofit membership organizations that want to use the Web to engage and activate their members. So each day, Convio sends tens of millions of emails to these members and then tracks the ensuing member transactions and activities, such as donations, advocacy, and click-throughs. Dave has honed his troubleshooting and scalability skills by keeping these very busy databases happy. He has presented at Hotsos and is a regular presenter at NoCOUG. He can be reached at **dabercrombie@convio.com**.

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 memberships. If your company is able to offer sponsorship at any level, please contact NoCOUG's president, Hanan Hit, at **hithanan@gmail.com**. ▲

Long-term event sponsorship:

Chevron

Oracle Corp.

Thank you! Year 2010 Gold Vendors:

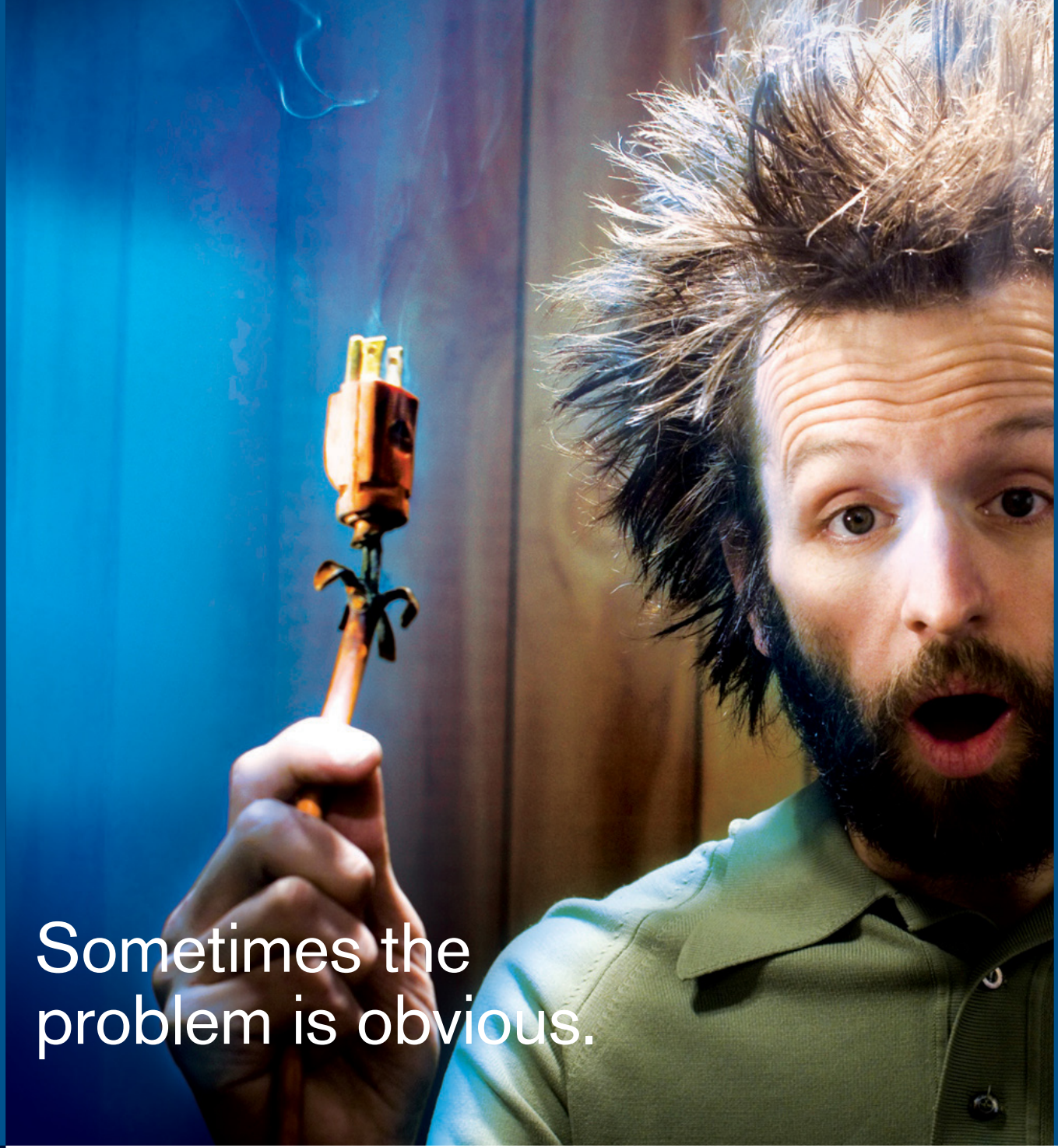
- 3PAR
- Confio Software
- Database Specialists, Inc.
- Delphix
- Rolta TUSC

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

TREASURER'S REPORT

Naren Nagtode, *Treasurer*

Beginning Balance		
April 1, 2010		**$ 43,171.32**
Revenue		
Membership Dues	2,735.00	
Meeting Fees	420.00	
Vendor Receipts	4,500.00	
Advertising Fee	–	
Training Day	700.00	
Sponsorship	–	
Interest	4.80	
Paypal balance	–	
Total Revenue		**$ 8,359.80**
Expenses		
Regional Meeting	2,648.67	
Journal	3,824.91	
Membership	34.61	
Administration	1,190.00	
Website	–	
Board Meeting	473.95	
Marketing	300.00	
Insurance	–	
Vendors	44.40	
Tax	–	
Training Day	20.90	
IOUG-rep	695.00	
Miscellaneous	20.00	
Total Expenses		**$ 9,252.44**
Ending Balance		
June 30, 2010		**$ 42,278.68**



Usually, it's harder to pinpoint.

Amazing what you can accomplish once you have the information you need.

When the source of a database-driven application slowdown isn't immediately obvious, try a tool that can get you up to speed. One that pinpoints database bottlenecks and calculates application wait time ***at each step***. Confio lets you unravel slowdowns at the database level with no installed agents. And solving problems where they exist costs a ***tenth*** of working around it by adding new server CPU's. Now that's a vision that can take you places.

A smarter solution makes everyone look brilliant.

CONFIO SOFTWARE

Download your FREE trial of Confio Ignite™ at www.confio.com/obvious

Download our FREE whitepaper by visiting www.oraclewhitepapers.com/listc/confio

NoCOUG Summer Conference

Session Descriptions

For the most up-to-date information, please visit **www.nocoug.org**.

Keynote

A Day in the Life of an Oracle Server Process
Craig Shallahamer, Orapub 9:30–10:30

Being an Oracle server process is no easy thing. While typically doing a whole lot of nothing, when asked to do something, expectations are extremely high. And all this happens while it's being subjected to all sorts of infuriating roadblocks. If you think teenage drama is exhausting, that's nothing compared to what an Oracle server process endures. This technical and entertaining keynote will bring to light some of the key performance and configuration issues we face today. With the aid of audience participation, impromptu demonstrations, and even a quick buffer and library cache dump, you'll come to respect, like never before, an Oracle server process and the code that breathes life into it.

Room 1220

Optimizing Internal Serialization Control
Craig Shallahamer, Orapub 11:00–12:00

Balancing high database performance and serialization control can be a daunting challenge with increasing complexity, transaction rates, and concurrency requirements. This presentation focuses on how Oracle meets this challenge by exploring the various internal serialization control schemes, such as the lock, the latch, and the mutex. This is a very practical yet deep internal presentation, filled with amazing discoveries about how Oracle works.

Maximize the Value of Your Oracle Stack with Integrated Application-to-Disk Management
Sandra Cheevers, Oracle Corp. 1:00–2:00

Oracle Enterprise Manager 11*g* has introduced industry-leading capabilities for managing the entire Oracle stack from packaged applications to Fusion Middleware, Database, and the underlying Sun systems. It also introduces groundbreaking innovations in end-to-end application performance management, configuration management and compliance, application quality management, and provisioning and patching. Join us for this unique opportunity to learn about the new Oracle Enterprise Manager 11*g*.

Workload Management for an Operational Data Warehouse
JP Dijcks, Oracle Corp. 2:30–3:30

Once upon a time the data warehouse was a peaceful land far, far away from the operational system craze. Alas, those times are long gone, and you as a data warehouse professional now need to manage very diverse workloads on your data warehouse. You deal with batch processes (both reading from and writing to your data warehouse), with continuous data loading, and with many different end-user workloads. In this session we will discuss how Oracle Database 11*g* Release 2 can address a mixed workload management problem. Rather than discussing a set of features, we will introduce a comprehensive solution that you can look at, understand, and implement at home to manage your data warehouse workload. You will see this all managed from Enterprise Manager, and you will see how you can handle exceptions, create policies, and make it all work in a real, live scenario that even includes parallel statement queuing. If you manage an operational warehouse, you should attend this session.

52 Weeks in the Life of a Database
Iggy Fernandez, Database Specialists 4:00–5:00

The most important tool in performance tuning is charts. As the old saying goes, one picture is worth ten thousand words. Oracle Database is richly instrumented but charts are surprisingly hard to come by. We demonstrate how to plot performance data in effective ways that allow us to visualize database performance. No special tools or licenses are necessary; the data can be obtained from Statspack, while Microsoft Excel can be used to produce stunning charts of database performance.

Room 1130

Solving SQL Injections That Exploit Zero-Day Vulnerabilities
Slavik Markovich, Sentrigo 11:00–12:00

Though many types of SQL injection can be prevented by secure coding practices, you can limit and even stop SQL injection attacks by deploying the correct tools to protect applications and databases. Certain classes of SQL injection, including those exploiting zero-day vulnerabilities, can be entirely blocked by deploying deep inspection tools, which will be demonstrated in the presentation. This course will present a new angle on a popular attack vector on the database layer of applications: SQL Injection. We will describe types and techniques of SQL Injection attacks on both web applications and built-in database-stored program units. We will show how simple SQL Injection can be used to own the database server through the means of privilege escalation. We will also list ways of preventing SQL Injection attacks—ranging from secure coding practices to various external tools that will alert and prevent SQL Injection attempts, and demonstrate how hacker techniques of evasion can be used to subvert them. Finally, we will introduce new deep inspection tools that can prevent SQL injection, even in zero-day scenarios.

Edition-Based Redefinition: The Key to Online Application Upgrade

Dan Morgan. 1:00–2:00

Large, mission-critical applications built on Oracle Database are often unavailable for tens of hours while the application's database objects are patched or upgraded. Oracle Database 11*g* Release 2 introduces revolutionary new capabilities that allow online application upgrades with uninterrupted availability of the application. Existing sessions can continue to use the pre-upgrade application until their users decide to finish; and, at the same time, new sessions can use the post-upgrade application. When sessions are no longer using the pre-upgrade application, it can be retired. The application as a whole therefore enjoys hot rollover from the pre-upgrade version to the post-upgrade version. The capability depends on these new kinds of objects: the edition, the editioning view, and the crossedition trigger. The capability as a whole is called "edition-based redefinition"—EBR for short.

New Features in Oracle Database 11*g* Release 2 You Won't Hear About from Oracle

Dan Morgan. 2:30–3:30

Oracle Database 11*g* Release 2 contains new functionality as well as substantive enhancements to existing functionality, such as Advanced Compression, PL/SQL Warnings, and even new ways to create tables and indexes. This presentation, by Oracle ACE Director Dan Morgan, will include live demonstrations in SQL*Plus that will show how to use the most valuable of these new features.

The Oracle ACE Program

Dan Morgan. 4:00–5:00

The Oracle ACE Program is designed to recognize and reward Oracle experts for advocating Oracle Technology and Applications. Oracle ACE recipients are chosen based on their significant contributions to, and activity in, their respective community. The program currently has two levels: Oracle ACE and Oracle ACE Director. This presentation will explain the ACE program and highlight the Oracle ACEs at the NoCOUG Conference. We will invite all Oracle ACEs attending the conference to join in talking about their experiences in the program.

Room 1140

Oracle and/or Hadoop—The Answers You Are Looking For!

JP Dijcks, Oracle Corp. 11:00–12:00

A lot has been blogged about MapReduce or its Hadoop implementation or Cascading, and sometimes it feels like this is the solution for everything from parallel processing to the common cold and beyond. In this session we will explain what MapReduce, Cascading, and Hadoop actually are; how you can apply the same rules and processes in Oracle; and what a good strategy for leveraging the best of both worlds could look like. This is not just a session with hundreds of buzzwords, we will actually show you an example of programming MapReduce within the database and discuss both pros and cons of this solution. We will also show you how to potentially leverage both processing models and explain what works in which situation.

BPM and SOA Alliance via Dynamic Distributed Data

Amir Bahmanyari, Franklin Templeton Investments. . 1:00–2:00

Business Process Management services are core providers of real-time data in a Service-Oriented Architecture implementation. A Process Instance is created as a result of a new transaction sent to BPM services. The dynamic changes in the business state of such transactions may be persisted in distributed databases. These databases include several instances, one of which persists BPM processes data. Another instance may store BPM business partners' related data. A separate database instance is utilized to persist global information about the transaction for reports generation. An application of such BPM database instances is to monitor the real-time changes in an in-flight transaction's business states. This presentation demonstrates how BPM services are architected to provide real-time business-state data in a SOA implementation context. Enterprise Context Services scopes such SOA database applications at the BPM layer.

The Modern Enterprise: Realizing the Benefits of Enterprise 2.0

Brian Dirking, Oracle Corp. 2:30–3:30

Enterprise 2.0 technologies such as portals, content management, social networks, and micro-blogs are transforming how people work and interact with users, content, business processes, and applications. With these technologies, users are empowered to find more relevant information when they need it, connect and collaborate with others in a business process context, and share knowledge more effectively with employees, customers, and partners. Join us in this session as we discuss how your organization can adopt and scale Enterprise 2.0 capabilities to address key business challenges such as improving employee productivity, increasing customer satisfaction, and accelerating time to market for new products and services.

Modernizing Oracle Forms Applications to Utilize Web Services

Jennifer McNeill, Unify.com 4:00–5:00

Many companies have millions of dollars invested in existing Oracle application systems, with most of these proven systems that run their business. It is generally not practical to spend countless dollars replacing existing functional systems with new technology or a new programming paradigm. It makes far more business sense to implement new technology (such as Java) where the impact will be the greatest (such as in the front end of the application) and the relative cost will be the most reasonable. While past versions of Oracle Forms have enabled companies to access the Web, 11*g* is providing one of the most robust environments thus far. However, the move to the Web is not always as easy as it seems. As in the past, we see many companies moving to Java because it seems the wave of the future. It is important that these organizations understand the business reasons to move their applications and determine if the effort and dollars spent will justify the migration costs. While migration to Java can provide many alternatives, it is not always the best solution for every organization. Understanding the risks of migration, the options available, and the benefits of Java is imperative to ensure that an organization is not spending precious IT dollars on an alternative that does not make business sense. ▲

One-Day Performance Seminar with Craig Shallahamer

Wednesday, August 18, 2010—CarrAmerica Conference Center, Pleasanton, CA

OraPub's One-Day 2010 Performance Seminar was created specifically for experienced DBAs and is squarely focused on advanced Oracle performance analysis. To develop the seminar, OraPub surveyed the most experienced students in the firefighting and advanced analysis courses and specifically asked what were the most memorable, valuable, and relevant aspects of the courses.

Craig Shallahamer is planning a day of intensive learning, which includes multiple aspects of Oracle performance tuning. We will learn how to analyze the resources consumed by competing processes and how to avoid hitting resource capacity limits. We will map the different performance problems and the specific techniques to approach each problem type. We will explore the relations between different components that impact performance and how they influence one another. Then we will take all this knowledge and learn how it applies specifically to the problem of free buffer waits, taking a deep dive into Oracle architecture in the process.

If you want to learn how to quickly solve Oracle performance problems, come take advantage of a full day of training by Craig Shallahamer on August 18, at the CarrAmerica Conference Center in Pleasanton for $400.

Craig Shallahamer, founder of OraPub, Inc., is the ultimate Oracle guru. He is technical—having just released his latest book, *Oracle Performance Firefighting,* he has also authored more than 23 technical papers and the revolutionary book, *Forecasting Oracle Performance.* He is experienced—having co-founded the Core Technologies and the System Performance Groups at Oracle and taught more than 10,000 students on 6 continents in 23 countries. And, most importantly, he is a passionate educator who brings his life experiences and engaging presentation style to every class.

What makes this seminar unique?

- What you learn can be immediately applied and demonstrated to your management
- The content has been selected from OraPub's most advanced students
- Delivered by Craig Shallahamer, well-known Oracle performance expert and author
- A practical and insightful weaving of internals, method, and advanced performance analysis

Seminar Objectives

- Learning about the many methods Oracle uses to gather performance data, their advantages and limitations, and how the analyst retrieves and uses this data
- Analyzing Resource Consumption: Learn how to measure network, CPU, and IO consumption and, when possible, their capacity limit using both Oracle and non-Oracle data sources
- Mapping the Terrain: Learn how to create a map of the performance terrain enabling advanced analysis, including how to visualize the situation and extract insightful solutions
- Computing System Behavior Analysis: Learn how a specific performance solution affects the relationship between the user experience, application work completed, and resource utilization
- Resolving the Free Buffer Waits Event: Learn how to resolve the complex wait events "free buffer waits" by delving into Oracle internals and applying the previous seminar content

Regular price: $400 for NoCOUG members / $600 for non-members

Contact training_day@nocoug.org or register at www.nocoug.org

Database Specialists: DBA Pro Service

DBA PRO BENEFITS

- *Cost-effective and flexible extension of your IT team*
- *Proactive database maintenance and quick resolution of problems by Oracle experts*
- *Increased database uptime*
- *Improved database performance*
- *Constant database monitoring with Database Rx*
- *Onsite and offsite flexibility*
- *Reliable support from a stable team of DBAs familiar with your databases*

CUSTOMIZABLE SERVICE PLANS FOR ORACLE SYSTEMS

Keeping your Oracle database systems highly available takes knowledge, skill, and experience. It also takes knowing that each environment is different. From large companies that need additional DBA support and specialized expertise to small companies that don't require a full-time onsite DBA, flexibility is the key. That's why Database Specialists offers a flexible service called DBA Pro. With DBA Pro, we work with you to configure a program that best suits your needs and helps you deal with any Oracle issues that arise. You receive cost-effective basic services for development systems and more comprehensive plans for production and mission-critical Oracle systems.

DBA Pro's mix and match service components

Access to experienced senior Oracle expertise when you need it
We work as an extension of your team to set up and manage your Oracle databases to maintain reliability, scalability, and peak performance. When you become a DBA Pro client, you are assigned a primary and secondary Database Specialists DBA. They'll become intimately familiar with your systems. When you need us, just call our toll-free number or send email for assistance from an experienced DBA during regular business hours. If you need a fuller range of coverage with guaranteed response times, you may choose our 24 x 7 option.

24 x 7 availability with guaranteed response time
For managing mission-critical systems, no service is more valuable than being able to call on a team of experts to solve a database problem quickly and efficiently. You may call in an emergency request for help at any time, knowing your call will be answered by a Database Specialists DBA within a guaranteed response time.

Daily review and recommendations for database care
A Database Specialists DBA will perform a daily review of activity and alerts on your Oracle database. This aids in a proactive approach to managing your database systems. After each review, you receive personalized recommendations, comments, and action items via email. This information is stored in the Database Rx Performance Portal for future reference.

Monthly review and report
Looking at trends and focusing on performance, availability, and stability are critical over time. Each month, a Database Specialists DBA will review activity and alerts on your Oracle database and prepare a comprehensive report for you.

Proactive maintenance
When you want Database Specialists to handle ongoing proactive maintenance, we can automatically access your database remotely and address issues directly — if the maintenance procedure is one you have pre-authorized us to perform. You can rest assured knowing your Oracle systems are in good hands.

Onsite and offsite flexibility
You may choose to have Database Specialists consultants work onsite so they can work closely with your own DBA staff, or you may bring us onsite only for specific projects. Or you may choose to save money on travel time and infrastructure setup by having work done remotely. With DBA Pro we provide the most appropriate service program for you.

Database Specialists

CALL 1-888-648-0500 TO DISCUSS A SERVICE PLAN

NoCOUG
P.O. Box 3282
Danville, CA 94526

NoCOUG Summer Conference Schedule

August 19, 2010—Chevron, San Ramon, CA

Please visit **www.nocoug.org** for updates and directions, and to submit your RSVP.
Cost: $50 admission fee for non-members. Members free. Includes lunch voucher.

8:00 a.m.–9:00	Registration and Continental Breakfast—Refreshments served
9:00–9:30	**Welcome:** Hanan Hit, NoCOUG president
9:30–10:30	**Keynote:** *A Day in the Life of an Oracle Server Process*—Craig Shallahamer, Orapub
10:30–11:00	**Break**
11:00–12:00	**Parallel Sessions #1**
	Room 1220: *Optimizing Internal Serialization Control*—Craig Shallahamer, Orapub
	Room 1130: *Solving SQL Injections That Exploit Zero-Day Vulnerabilities*—Slavik Markovich, Sentrigo
	Room 1140: *Oracle and/or Hadoop—The Answers You Are Looking For!*—JP Dijcks, Oracle Corp.
12:00–1:00 p.m.	**Lunch**
1:00–2:00	**Parallel Sessions #2**
	Room 1220: *Maximize the Value of Your Oracle Stack with Integrated Application-to-Disk Management* —Sandra Cheevers, Oracle Corp.
	Room 1130: *Edition-Based Redefinition: The Key to Online Application Upgrade*—Dan Morgan
	Room 1140: *BPM and SOA Alliance via Dynamic Distributed Data* —Amir Bahmanyari, Franklin Templeton Investments
2:00–2:30	**Break and Refreshments**
2:30–3:30	**Parallel Sessions #3**
	Room 1220: *Workload Management for an Operational Data Warehouse*—JP Dijcks, Oracle Corp.
	Room 1130: *New Features in Oracle Database 11*g *Release 2 You Won't Hear About from Oracle* —Dan Morgan
	Room 1140: *The Modern Enterprise: Realizing the Benefits of Enterprise 2.0*—Brian Dirking, Oracle Corp.
3:30–4:00	**Raffle**
4:00–5:00	**Parallel Sessions #4**
	Room 1220: *52 Weeks in the Life of a Database*—Iggy Fernandez, Database Specialists
	Room 1130: *The Oracle ACE Program*—Dan Morgan
	Room 1140: *Modernizing Oracle Forms Applications to Utilize Web Services*—Jennifer McNeill, Unify.com
5:00–	**Networking and No-Host Happy Hour at Izzy's Steaks and Chops, 200 Montgomery Street, San Ramon**

RSVP online at www.nocoug.org/rsvp.html

The *NoCOUG Journal* design and production: Giraffex, Inc., S.F. Front cover photo–Grapes on the vine in California. Photo courtesy of USDA Natural Resources Conservation Service.