Official Publication of the Northern California Oracle Users Group

NoCOUG JOURNAL

VOL. 18, No. 2 · MAY 2004 $15

Don't Go inSANe!

A humorous and insightful look at RAID and more, thanks to Mogens Nørgaard and James Morle.

Oracle Performance Expertise

Don't miss this interview with Oracle performance guru Cary Millsap, author of Oracle Performance Optimization

Mass Deployment Strategy

An overview of the choices for Oracle 10g.

Java in Oracle

A look into the best of emerging Java technologies.

Boarder Your Knowledge with NoCOUG this Spring!

Editor's Note

Putting the NoCOUG Journal together each quarter is a big job, and we hope you like the changes we've been making. We couldn't do it without the help of many people. This quarter, we would especially like to thank the following individuals for researching and contributing articles: Krishna Kakatur, Sudip Datta, and Chris Lawson. Thank you to Mogens Nørgaard and Randy Samberg for the interviews they conducted and the articles they contributed. Thank you to Sumathy Panicker for contributing our Tech Tips for this issue, and finally, thank you to Cary Millsap and James Morle for taking the time to be interviewed for articles in this issue.

If you are reading this and have an idea for an article or a Tech Tip, please contact us at **journal@nocoug.org**. We look forward to hearing from you!

—Lisa Loper and Laurie Robbins, *Journal Editors*

Table of Contents

Editor's Note 2
NoCOUG Board 2
Publication and Submission Format....2
Advertising Rates....2
President's Message....3
Profiling Oracle Performance Expertise 4
Tech Tip 7
Making the Best Use of Java in Oracle....8
Don't Go inSANe 14
Sponsorship Appreciation 18
Treasurer's Report 18
Choosing the Right Mass Deployment Strategy for Oracle Database 10g Software 19
Isolating Performance Problems....23
NoCoug Conference Register 26
Tech Tip 27
NoCOUG Spring Conference Schedule 28

–ADVERTISERS–

Database Specialists....10
Embarcadero Technologies....10
MissionCritical 24/7 13
LECCOTECH....13
EMC²....15
Quest Software 17

NOCOUG BOARD

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

Vice President
Darrin Swan, Quest Software
darrin@leccotech.com

Treasurer/Secretary
Judy Lyman, Contra Costa County Public Works
gooma@california.com

Membership and Past President
Joel Rosingana, Independent Consultant
joelros@pacbell.net

Webmaster
Eric Hutchinson, Independent Consultant
erichutchinson@comcast.net

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

Laurie Robbins, Remtech Services, Inc.
lrobbinsmis@yahoo.com

Vendor Relations
Colette Lamm, Independent Consultant
colette_lamm@yahoo.com

IOUG Representative
Hamid Minoui, Independent Consultant
hminoui@pacbell.net

Conference Agenda Coordinator
Eric Buskirk, Mission Critical 24/7
ebuskirk@mc247.com

Director of Marketing
Jen Hong, Cisco Systems
hong_jen@yahoo.com

Director of Public Relations
Les Kopari, Corner Pine Consulting
(650) 802-0563

Members, At Large
Vilin Roufchaie, Cingular Wireless
vilin.roufchaie@cingular.com

Randy Samberg, Independent Consultant
rsamberg@sbcglobal.net

Publication and Submission Format

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

The submission deadline for the upcoming August issue is July 1, 2004. Article submissions should be made in electronic format via email if possible. Word documents are preferred.

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

Copyright ©2004 by the Northern California Oracle Users Group. Permission to reproduce articles from this publication, in whole or in part, is given to other computer user groups for nonprofit use, with appropriate credit to the original author and the *Northern California Oracle Users Group Journal*. All other reproduction is strictly prohibited without written permission of the editor. Two copies of each reprint should be sent to the editor.

ADVERTISING RATES

Contact: Nora Rosingana

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

The NoCOUG Journal *is published quarterly.*

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

Personnel recruitment ads are not accepted.

"Common sense is very uncommon."

—*Horace Greeley*

Roger Schrag

I recently attended the Hotsos Symposium on Oracle Performance and was struck by the fact that three keynote speakers touched on a common topic: Jonathan Lewis discussed the burden of proof (which might sound familiar if you attended NoCOUG's conference last August), Thomas Kyte spoke about what we *think* we know (but perhaps don't), and Cary Millsap spoke about some things he thought we *should* know. In one way or another, all three speakers addressed the importance of common sense in the IT world.

Thomas Paine wrote his famous pamphlet "Common Sense" in 1776 and helped ignite the American Revolution. Now, revolutions aside, I believe common sense is a critical character trait for the successful Oracle professional. Perhaps the importance of this trait is more easily seen when we look at situations where common sense is absent.

Suppose you went to the doctor and said you had a pain in your side. Would you be alarmed if the doctor put you on a gurney straight away and sent you into the operating room? I would! Shouldn't the doctor ask some questions first? And run some tests before fetching the scalpel?

Now suppose you are the DBA and your users complain that "the system is slow." Imagine you respond by turning to Statspack reports—you find the three top resource SQL statements and spend the next two days tuning them. You are proud of your achievement of making these three queries run ten times faster, but should you be surprised when your users don't notice any improvement? Probably not. The users might be having response time issues with a particular data entry screen, while the queries you tuned made some overnight batch job that nobody cares about finish at 1:30 a.m. instead of 1:45 a.m.

Understanding a problem before jumping in with a proposed solution seems like common sense, right? In the example of the doctor, this is probably obvious to everyone. But how many of us can remember a case where a DBA or developer failed to ask enough questions and ended up fixing the wrong problem? On these occasions perhaps common sense was forgotten.

As another example, suppose you are chaperoning your son's band class on a road trip. You pull in at a fast food joint and run inside to order takeaway food for the ten of you. Would you order one chicken sandwich, pay for it, order the next one, pay for it, and repeat the cycle ten times? Of course not. Instead you would order ten chicken sandwiches at once. This allows the restaurant staff to optimize the fulfillment of your order and get you back on the road as soon as possible.

But now put your Oracle developer hat on. How many times have you seen PL/SQL code that fetches rows from a cursor one at a time and performs half a dozen SQL statements on each iteration of the cursor loop? How many of these times do you think you could have replaced the whole mess with one or two mass update statements or bulk fetches and binds?

Letting involved parties know your full requirement at once instead of breaking it into tiny bits unnecessarily seems like common sense. The staff at the fast food restaurant can prepare and ring up ten chicken sandwiches in one shot much faster than preparing each individually as a separate order. Likewise, Oracle can update 1000 rows in one shot much faster than it can process 1000 requests to update one row each. We all have probably seen (and perhaps even written) inefficient code like this at one time or another—another case of forgetting to use common sense.

These are just two examples of the importance of using common sense in your role as Oracle DBA, application developer, or IT professional in general. The most successful and productive folks out there are the ones who don't forget to apply a little common sense to everything they do. This allows them to work smarter, instead of harder.

Speaking of being smart, don't forget that NoCOUG's Spring Conference is coming up Thursday, May 13th at Lockheed Martin in Sunnyvale. This conference will include a dozen technical sessions offering you great learning opportunities. You can get further acquainted with Oracle Database 10g and Oracle Application Server 10g, as well as pick up real-life experience from others that you can apply to your daily work routine right now.

If you haven't joined NoCOUG as a member for 2004 yet, an individual membership costs just $70. That gets you free entry to the Spring Conference, as well as the Summer and Fall conferences later this year. That's three full days of Oracle education for just $70. Common sense will tell you it's the right thing to do!

See you at the NoCOUG Spring Conference on May 13th! ▲

Profiling

Oracle Performance Expertise

An Interview with Cary Millsap

Author of *Optimizing Oracle Performance*, published by O'Reilly and Associates, Inc., and cofounder of Hotsos, a company dedicated to Oracle system performance

NoCOUG had the opportunity to talk with Cary Millsap at the Rocky Mountain Oracle User's Group (RMOUG) Training Days this past February where he was a presenter. Cary is a frequent speaker at Oracle User Group meetings and recently completed a book titled *Optimizing Oracle Performance* published by O'Reilly. Cary's experience and accomplishments in the space of Oracle performance date back to 1989 when he started at Oracle Corporation. After ten years at Oracle, Cary resigned and cofounded the company Hotsos (pronounced *hot sauce*) where he works today.

NoCOUG: ***Let's start at the beginning, can tell us where you grew up and went to school?***

Cary: I went to school 1st grade through college in Oklahoma and studied mathematics for my undergraduate degree. Then I moved to Colorado Springs to get a Masters in Computer Science, and finally ended up in Dallas in 1988 to attend Southern Methodist University (SMU) for an MBA. After SMU, I got a job with Oracle (1989), where I worked for 10 years. Then, in 1999 we started Hotsos, so I guess I've lived in Dallas since 1988.

NoCOUG: ***What is the biggest difference between what you were doing at Oracle and what you're doing now at Hotsos?***

Cary: Well, there were several different "phases" while working at Oracle. When I started, I was hired for the combination of my business and technical backgrounds. I think my first job was setting up printers for financial application installations, so it took about a year or so to find a niche.

Eventually, I moved into a group; I think it was called National Technical Response Team (NTRT), and that's where I met a guy named Bob Rudzki. This team went to these busted Oracle Financial Application sites, and there were tons of them. It sounded like the most horrible thing I'd ever heard of but it turned out to be a really fun team mostly because of Bob. Over time, the team grew in skills and experience and someone decided to have other product specialists join the group so we could cover sites other than Oracle Financials. That's when we went from NTRT to the National Product Specialists Team (NPST).

A few years later, Ray Lane and Robert Shaw came to Oracle. In the days before Ray and Robert, consulting was this team that went with a broom and a dust pan to all these busted sites and tried to fix everything. We were basically at the beck and call of whatever sales division screamed the loudest about their busted accounts.

Cary Millsap

We were basically at the beck and call of whatever sales division screamed the loudest about their busted accounts.

However, when Ray and Robert joined, consulting became more of a function that had a goal with a motive and a mission.

As a result of changes Ray and Robert wanted to make within the organization, Robert asked me run a new group to do installation and upgrades. Although Robert was initially in favor of farming out the installation and setup to other companies, Bob and I were able to convince him to keep that technical work. We explained, the installation and setup is where the foundation of whether a site will succeed begins. It can be virtually impossible to do anything on top of a poor foundation.

So that's where the Core Technologies Team got its start. For the first three or four years, I was technical in the field going to 30-40 client sites a year, about one a week. Then, I became a manager.

From there, I started the System Performance Group (SPG). Somewhere in the middle of becoming a manager and leaving Oracle, I guess about the last five years of my career, the job became playing a video game in Excel and PowerPoint. I spent a lot more time in PowerPoint, Word

and Excel, doing budgets, and staff reviews, than I spent working with Oracle products. Those last five years at Oracle were radically different from what I wanted to do.

I resigned at Oracle in 1999, and we started Hotsos. Hotsos has been a place where I can do research and that's been the fun part. Once I reached the level of management

We explained, the installation and setup is where the foundation of whether a site will succeed begins. It can be virtually impossible to do anything on top of a poor foundation.

at Oracle, I was doing the occasional hot project, and it was a lot of fun to work with really talented people of course, but what I was doing day-to-day was really staff management; what my business partner, Gary Goodman, calls Adult Day Care. That wasn't the fun stuff, and now I'm able to do the fun stuff.

NoCOUG: ***So for you, the fun stuff is the research?***

Cary: Yes, I think that's probably the best way to put it. Since my undergraduate degree was in mathematics, I think what I enjoy doing the most is predicting and explaining. More than making something work once and then going some place else and making something different work. What I really enjoy is doing the same process a couple times, and then trying to work out a formula that allows me to teach someone else what will happen without them having to experiment. It's also economically good because it saves people money to be able to figure things out without actually having to try them. The work is challenging and it's got all the fun elements from the math background, the computer science background and the business background all put together.

During one of Gary Goodman's presentations, he mentioned that the Hotsos team exclusively uses the 10046 trace event for diagnostic tuning. In discussing tuning with others, there are typically two approaches, the "top down" approach (gathering system wide statistics) or the "bottom up" approach (using the 10046 trace event). Why does Hotsos use the bottom up approach exclusively?

Well, something strange happened in about '95. I went to customer sites weekly and tuned systems for a living. I thought I was pretty good at it, but in retrospect I wasn't. I was probably good at it for the period, but compared to now, absolutely horrifying. About the time Oracle released version 7, one of the top tier people in my group, a guy named Virag Sakena, started submitting engagement summary reports that showed by noon on Monday he was as far along as I had gotten in my best week by Friday. His reports showed exactly what was creating the problem.

That was significant, because typically, when we got called into a site, it was not something specific—the problem was defined by the customer as something more general, such as we can't bill our customers because the invoices take too long to print. Everywhere we went we had a business problem to solve. We never went to a site where they said, our problem is our cache hit ratio is horrible. Who cares about the cache hit ratio except by virtue of the fact it might link into why they can't print invoices or shipping labels.

Virag is the first guy I ever met that was able to pinpoint the root cause of the problem correctly from the first guess. It wasn't, 'I think if we increase the size of the buffer cache hit ratio by 20 percent it might make a difference.' It was, 'If we fix this piece of SQL it will make the process which now takes an hour to run, 3 seconds to run.' And he was right over and over. It was phenomenal, and it's exactly what you want to know as a business; what's going to happen as a result of an investment.

Virag was using the wait interface, and he was very successful as an individual and as a project leader using that approach, but we also had the task of teaching the rest of my team. At Oracle, I didn't have the time to dedicate to the research necessary to construct not just a set of cool techniques but literally a method that works—a method that anyone could use which doesn't have any point in it where you have to guess; basically a decision tree that anyone with reasonable intelligence could walk and succeed in fixing a performance problem.

The book and training classes present this methodical approach that anyone can execute. And I think we have succeeded extremely well in being able to teach people how to do it; and people are doing it correctly and solving problems. At Hotsos, we probably have 17 different case studies and examples showing why the old methods of looking at systems statistics don't work and why they fail so spectacularly. So for performance problem diagnosis, I'm convinced there's no other way than the bottom up approach.

There are other areas in the performance domain than solving problems. There is capacity planning, performance acceptance testing, activities designed to be more forward looking. In these cases, the standard types of statistics that

Who cares about the cache hit ratio except by virtue of the fact it might link into why they can't print invoices or shipping labels.

people usually think about are far more applicable for capacity planning. For example, Statspack reports are great when you are doing capacity planning. The summary reports you get from Statspack are far more effective for these more forward looking activities than the detail reports you get from the bottom up approach.

NoCOUG: ***So to effectively approach tuning, you have to have some idea of what process is causing the problem?***

Cary: Absolutely, you have to know the business. The business doesn't exist to support the system; it's the other way around. Using the business school approach, the right question to ask is *What's causing the company to hemorrhage cash*

flow or net profit or return on investment?; that's what's important. Whatever it is, I don't care how cool or not cool the technology is, the business needs to maximize cash flow, net profit, and return on investment.

NoCOUG: ***Have you had a chance to work with 10g and test some of the new features, specifically how they affect the use of the 10046 trace event?***

Cary: It finally becomes properly supported, the method we teach. Oracle has advertised a lot of interesting features in 10g, but I haven't heard anybody talk about what I think is the most exciting part of Oracle 10g, DBMS_MONITOR.

Currently, tracing someone's specific user action can be a big problem at sites that do a lot of connection pooling or session pooling. For instance, Nancy comes to a technology expert and complains, "I can't print shipping labels because it takes too long." The technical expert hears, "It hurts when I *blah blah blah*," and thinks I don't know exactly what she's complaining about, but I'll use the trace event and figure out where her time went. In Oracle versions 7, 8*i* and 9*i*, if your architecture causes Nancy, Mary and Jim to connect to the same Oracle server process, the tracing attribute can only be turned on and off at the server process level. When you turn on trace for Nancy's problem, you are not only getting Nancy's data, you're getting Mary's, and Jim's data as well. It can be really cumbersome to try and separate everything you want from everything you don't want.

In 10*g*, the tracing mechanism is now detached from the session and it becomes an attribute of an individual user action where it ought to be. It becomes: I can now trace Nancy and I don't care whether she's running across a grid

Whatever it is, I don't care how cool or not cool the technology is, the business needs to maximize cash flow, net profit, and return on investment.

on this node and that node. All the trace data shows up and there are tools that allow me to view it like a linear sequential list of steps that Nancy motivated. I don't have to worry about having Mary's or Jim's data being infused, which I have to take time to separate out; it's just perfect. With 10g, they have promoted the 10046 tracing event to full tier 1 support. Now with, DBMS_MONITOR, it's fully documented and fully supported and it does what we wish it had all along. It fully legitimizes the method that we created over the last four years.

NoCOUG: ***Do you see Oracle being challenged by open source software the way the OS market has been affected by Linux?***

Cary: Hugely, yes. Oracle is a really cool product and with version 7.3 it probably did almost everything that most users needed it to do. Now there's the odd 11 terabyte data warehouse, but probably 99% of businesses that need to use a database don't need partitioning, or RAC, or a lot of the features that version 8*i*, 9*i* and 10*g* provide. MySQL is perfectly suitable and adaptable to handle the situation, so what it comes down to is the economic argument of do I really want to spend $20,000 on a product that is absolutely superior to this something free. Now, there's no such thing as a free puppy, you've got to buy a house for it, food, toys, and training. Open Source software has some of those trappings attached to it as well. You have to pay someone to manage it you have to buy books, or consulting, basically

Now, there's no such thing as a free puppy, you've got to buy a house for it, food, toys, and training. Open Source software has some of those trappings attached to it as well.

you have to make it work. So, it's not free, it's just $20,000 less to get going free. Although, I do see Open Source solutions being a factor in some Oracle decision making including pricing, what features go into the product, and how features get bundled. Because, frankly, there are a lot of websites out there that run fine on top of MySQL.

I try not to be a MySQL expert, but it's clearly moving in a direction where smart people are committed to making a really good product. They are even instrumenting their kernel in a manner similar to what Oracle emits in the 10046 trace facility. This will make MySQL a lot more diagnosable and manageable from a performance perspective. And it works really well. It does most of what most people need very well. Oracle does it better, but the question is: Is it enough to justify the cost?

NoCOUG: ***What technology trends do you see having the biggest impact on the IT profession in the next 5 years?***

Cary: I don't think of myself as much of a visionary except perhaps in the little areas that I work in; however I do think the method used to optimize Oracle is a big deal. The new trend that the Optimizing Oracle Performance book describes, that's a big deal. I think that revolutionized IBM about 15 or 20 years ago. They were looking at all that ratio garbage as well, and then somebody figured out, what we really want to answer is the question *Why did this take so long?*, not *What things were being used while this was running*. And that's what this approach is all about.

The optimization technique of looking at what the problem is in a better way is going to continue to unfold. The way to tune the database is to get rid of wasted work load. The traditional methods of looking at system performance can cause you to never see wasted work load. So I think that will continue to be a minor revolution over the next five years.

NoCOUG: ***How did you get involved in writing the book?***

Cary: In a way, the book got pulled out of me. We had to write material for our class members because giving someone PowerPoint slides is not as effective as giving them PowerPoint slides and a stronger deeper document that they can study when they get home. My plan was to write a good

bit of it privately and go to three publishers with a proposal. I envisioned this book being more like a regular computer science book you get in college. It's not a product book at all, it's a method book about a particular product. So, I sent a message to the Oracle-L mailing list explaining I've got this book in progress. Jonathan Gennick from O'Reilly responded and said he'd be interested in knowing more about it. I talked to Jonathan a little bit. He actually came to the class to see our method so he could be fully immersed and not become an antagonist to the method. The method can seem bizarre at first especially if you are more familiar with looking at system wide statistics. The primary goal in the book is to expose the method and legitimize it so people will give it a chance. It's a lot easier now; I've said books are like neck ties for ideas. An idea with a book that goes with it is instantly more legitimate in someone's mind than an idea that doesn't have a book. I guess people have a fair right to say, if this is so great how come no one else is talking about it. So now with the book, more people are talking about it.

NoCOUG: ***Did you pick the animal on the cover of the book?***

Cary: Yes, actually I chose an insect, and they promptly did not use it. I chose the robber fly a really nasty, ugly looking bug (**http://www.insects.org/entophiles/diptera/dipt_004.html**). I decided on the robber fly because it's got a fabulous targeting system. It's got one of the most sophisticated abilities to fly, intercept and catch something. It can identify its prey from a long way away, know what it is, and not target something it doesn't want. That's the whole point of the method in the book, getting to the point instead of getting distracted by something else. But, it turned out O'Reilly didn't have a robber fly in their library of wood cuttings or whatever they use for the book covers. They ended up giving me a yellow jacket instead. I know Jonathan Gennick tried really hard. I even told them I'd pay them to hire an artist because I thought the link was so cool, but they just decided it's not worth it to buy a bug.

NoCOUG: ***And lastly, what do you enjoy doing in your spare time?***

Cary: My family is very important to me, and I have a beautiful wife, and two boys (ages four and six). I also have a fully equipped wood shop. I spend time out there and built our own oak table (**http://www.oaktable.net/oaklist.jsp**) and lots of other furniture in the house. I like to play almost any sport with a ball; softball, golf, racquetball are some that come to mind, although, my relationship with golf is kind of stormy. I enjoy golf when I play with my wife, but when I try to be competitive I usually walk home with a lot less money than I went to the golf course with and that's frustrating (laughs). ▲

For more information on the tuning method used and taught by Hotsos, or to download white papers on the subject, visit **www.hotsos.com**.

Acknowledgements

Interview conducted by Laurie Robbins, NoCOUG Journal assistant editor. Bug illustration by Hans John, 1929 [Museum Wiesbaden]. Thanks to Andy Rivenes, Bill Reid and Dale Snearly for their input on technical questions. And, a special thanks to Lisa Loper for her editorial expertise and assistance.

TECH TIPS

Dropping Database Links

When you clone a database, the global_name of the database has to be updated – this is essential if you have global_name = true in your environment and you want to create or drop db_links with the same name as your database.

Consider the following scenario.

1. You cloned FINDEV from FINTST.
2. You are trying to get rid of a database link called FINTST and you get the following error:

```
SQL> DROP PUBLIC DATABASE LINK FINTST;
DROP PUBLIC DATABASE LINK FINTST
            *
ERROR at line 1:
ORA-02082: a loopback database link must have a connection qualifier
```

3. This is because the global name is set to FINTST. Issue the following command:

```
SQL>UPDATE GLOBAL_NAME SET GLOBAL_NAME ='FINDEV'
SQL>COMMIT;
```

4. Drop the database link. (You might have to logout and login for the command to be successful.) ▲

Contributed by Sumathy Panicker

Making the Best Use of Java in Oracle

by Krishna Kakatur

Whether you are using Java in your Oracle database programming or not, it is time you relook into your programming strategy and adapt the best out of emerging Java Technologies.

Java has emerged as the most versatile object-oriented programming language of choice for web based, user-interface and embedded applications. Java has been recognized as a safe language because of its strong typing and automated memory allocation techniques, such as garbage collection. Java virtual machine (JVM) provides the fundamental basis for platform independence.

To grab these benefits and integrate them with relational features of RDBMS, Oracle started supporting Java with its 8i Release. This article introduces Java Stored Procedures for those not using Java in their database programming, while giving practical tips for those already using Java Stored Procedures.

The Oracle Aurora JVM is tightly integrated with the Oracle database server which supports Oracle's database session architecture. Java session space is divided into two memory areas, in a way to provide the same isolation, scalability and data integrity capabilities as SQL sessions. The first area has all the shared Java classes either from Oracle JVM or user created, and is shared by all sessions. The classes are read, resolved and loaded only for the first time; subsequent sessions then benefit from this shared memory. The second area has private instances of the classes that are loaded in the first area, which are kept and garbage collected in an individual session space.

Advantages of Using Java in Database Programming

There are a number of benefits that Java can bring into your database.

Safety

The prominent reason for the widespread popularity of Java is that it is a safe language. Languages, such as C and C++, can introduce security problems within the database because memory allocation is handled by the programmers instead of the language itself. Java, by its design, is a safe language for web-based and database-based programming. Its type of safety and automatic memory management allows for tight integration with the RDBMS.

Reusability

Component based development using JavaBeans lets you reuse the components, thus saving resources rather than having to reinvent the wheel every time.

Robustness

As opposed to PL/SQL, Java requires declaring exceptions that can be thrown by methods in any class, thereby making Java stored procedures more robust.

Krishna Kakatur

Cross-Database and Cross-Tier Portability

As Java has been accepted as an open standard and supported by all major databases, it is best-suited for inter-database independent programming. It can also serve as a portable language for both persistent logic in the database-tier and business logic in the middle-tier.

Complex Database Persistent Logic

The limitations in implementing complex logic in the database programming using PL/SQL can now be overcome using Java programming which extends PL/SQL functionality and allows implementing more complex database logic.

Performance Advantage

Java offers significant performance advantages for computationally expensive routines. If your stored procedure requires less database access and more computations, math operations, variable processing and manipulation of data types, Java stored procedures prove to be faster than PL/SQL.

Java vs. PL/SQL

When you start with your database programming, how do you choose whether to program in PL/SQL or Java? Here is a simple guideline:

Use PL/SQL, when your requirement is centered on database tables and other schema objects, meaning you have less business logic and more database access. As a simple example, assume you are developing an application logic to query, update, add or delete employee records from employ-

ee tables and you don't have any more business requirements. PL/SQL is the best solution.

Use Java, when your business requirements are exhaustive and can't be easily handled by PL/SQL. Also, when you want to use XML and Web Services. As a simple example, assume you are developing a Multimedia application, where you want to manipulate audio and video while communicating with the database objects at times. Java is the best solution.

Best of all, you can use both Java and PL/SQL interchangeably within your application. One of the greatest benefits about running Java code directly in the database is the ability to create Call Specifications, which serve as transparent interfaces available to SQL, PL/SQL, and Java. The Call Specification makes the underlying implementations transparent, which can change over time from PL/SQL to Java or vice versa.

A Step-by-Step Approach to Create Java Stored Procedures

Let us create a simple Java Stored Procedure that prints "Hello World".

STEP 1: Write Source Code for Java Class

In your favorite editor, type the following code and save it as a file named HelloWorld.java.

```
public class HelloWorld {
   public static void print () {
      System.out.println("Hello World");
   }
}
```

STEP 2: Compile the Java Class

Invoke the java compiler and compile the source code, as below.

```
javac -CLASSPATH
c:\oracle\ora10\jdk\jre\lib\rt.jar;c:\oracle\ora10\jdbc\lib\classes
12.zip HelloWorld.java
```

In the above example, JDK supplied with Oracle 10g, installed under Oracle Home "c:\oracle\ora10\", is used. You can use compiler included with Sun's JDK as well.

The Java compiler translates the source file into Java byte-codes and produces a binary file, called HelloWorld.class.

Note: It is always a good idea to specify your CLASSPATH on the javac command line, instead of depending on the environmental variable setting. This is particularly important when writing shell scripts or make files.

STEP 3: Load the Class into your Schema

Use loadjava utility to load the java class file into your schema.

```
loadjava -user scott/tiger HelloWorld.class
```

Oracle supplied loadjava stores the class as a Java Schema Object in the database.

Note: Loadjava command-line utility and DBMS_JAVA.LOADJAVA() can load Java Source, Java Classes, Java Resource Files and Java Archives (.jar files) into the Oracle database. If you are accessing loadjava script over the network, you may have to make a local copy and modify the variables.

STEP 4: Publish the Class

To make the Java Class accessible to SQL and PL/SQL, you need to Publish the Class through a call specification. This defines the SQL types method takes as input parameters and returns as output parameters.

```
SQL> CREATE OR REPLACE PROCEDURE print_hello AS
  2  LANGUAGE JAVA NAME 'HelloWorld.print()';
  3  /

Procedure created.
```

Classpath Resolution: In contrast to an Operating System where the resolver uses CLASSPATH, the Oracle database applies resolver on a per-class basis, as the classes are stored as schema objects. As such, when you execute a Java Stored Procedure, the default resolver looks for necessary supporting classes, such as String, in the current schema first and then in PUBLIC.

STEP 5: Test the Stored Procedure

Now that we have a PL/SQL wrapper for our Java Class, we can invoke it exactly the same way as any other PL/SQL Procedure, either from SQL*Plus, or a trigger or another procedure or a package or a Java program running in the middle-tier.

```
SQL> set serveroutput on
SQL> exec dbms_java.set_output(10000);
SQL> exec print_hello
Hello World
```

Note here that the output will be written to trace files by default, not to the user screen. We have called dbms_java.set_output() procedure to redirect output to the SQL*Plus text buffer.

Due to the consistent call specification, you will never observe a difference between a procedure written in PL/SQL or Java, unless you see the error message below!

```
ORA-29532 Java call terminated by uncaught Java exception
```

This message is returned by a Java stored procedure when Java Class throws an exception not caught within the Java Class.

Sample Scenarios for Java Stored Procedures

SCENARIO 1: Your company has two on-line transaction processing application systems. The first one, hosted on a Sybase database, is used by your Engineering team. This database has product_master table, which stores Product Information, including Product Number and List Price. The second application, used by your Sales team, is hosted on an Oracle database. This application has some persistent logic in the form of Oracle PL/SQL stored procedures, that need real-time product information from the Sybase database. Here is a simple solution to have such a data access.

In this case, we load the Java source and compile it within the database, thus eliminating the necessity of compiling the Java source with a JDK installed in our operating system and then loading the resulting class into the database. Also, we write just one source code file for both Java Source and Call Specification and execute it in a SQL*Plus window. Here is the source code file:

ProductDetails.sql:

```
CREATE OR REPLACE AND COMPILE
JAVA SOURCE NAMED ProductDetails AS
import java.util.*;
import java.sql.*;

public class ProductDetails {

    public static Double getPrice (String productNo) {

       Connection conn;
       Statement stmt;
       ResultSet rs;
       String sqlQuery = "";
       Double productPrice = new Double(-999999.00);

       try {

          String dburl = "jdbc:sybase:Tds:mrp.world:20000";
          Properties props = new Properties();
          props.put("user", "mrp");
          props.put("password", "mrp");
          props.put("database", "mrp");
          props.put("server", "MRP");

          // Check that Sybase JDBC Drivers are installed
          // and accessible from the schema in which this
          // Java Function is created.
          Class.forName("com.sybase.jdbc.SybDriver");
          conn = DriverManager.getConnection(dburl, props);
          stmt = conn.createStatement();
          sqlQuery = "select list_price from product_master"
                   + " where product_no='"+productNo+"'";
          rs = stmt.executeQuery(sqlQuery);
          while(rs.next()) {
             productPrice = new Double(rs.getDouble("list_price"));
          }
          conn.close();

       }
       catch (Exception e) {
          System.out.println("Error executing Query: "+sqlQuery);
          System.out.println(e.toString());
          productPrice = new Double(-999998.00);
       }

       return productPrice;

    }

}
/

show errors

CREATE OR REPLACE FUNCTION get_product_price
(product_no VARCHAR2) RETURN number AS
LANGUAGE JAVA NAME
'ProductDetails.getPrice(java.lang.String)
return java.lang.Double';
/

show errors
```

Caution: When you write one single .sql source code file for PL/SQL and Java, exercise caution. Java is case sensitive, while PL/SQL is not!

Once this function is created, the packages of your Sales application can transparently get the product information, without even knowing that the function uses Java and JDBC to connect to a Sybase database.

```
BEGIN
 ...
 v_list_price := get_product_price(v_product_no);
 ...
END;
```

There is one more point to note here. In the event that the Oracle user does not have permission on the Socket to connect to Sybase database, you will see an error similar to one below:

There's No Substitute for Experience

Our team represents some of the most knowledgeable and experienced in the industry. We are authors and speakers with long careers as Oracle experts, averaging 12 years. Our specialty is providing **remote DBA services** and **onsite Oracle database consulting**.

We offer a free consultation to discuss:

- ➤ Increasing uptime and reliability
- ➤ Minimizing downtime and data loss
- ➤ Optimizing performance
- ➤ Reducing cost of database operations

Call Us Today!
(415) 344-0500 • (888) 648-0500
www.dbspecialists.com

ORACLE | CERTIFIED SOLUTION PARTNER

DatabaseSpecialists

What kind of a person thinks building and managing enterprise databases is fun?

EMBARCADERO TECHNOLOGIES

```
sales@SALESP> select get_product_price('K1-GT-DESKTOP-R42') from dual;

GET_PRODUCT_PRICE('K1-GT-DESKTOP-R42')
--------------------------------------
                                -999998

sales@SALESP> exec null;
java.security.AccessControlException: the Permission
(java.net.SocketPermission
123.456.78.90:15000 connect,resolve) has not been granted by
dbms_java.grant_permission to
SchemaProtectionDomain(SALES|PolicyTableProxy(SALES))

PL/SQL procedure successfully completed.
```

In one of the Oracle databases, I was not able to get server output redirected to SQL*Plus screen, when a Function based on Java Class is executed. It was fine with Procedures based on Java Classes. I am not sure why, but I could resolve the issue using the above "exec null" work around!

Hint: Based on the error message, you can quickly get to know what permissions your Oracle DBA has to grant to the application user. Don't forget to close the session and re-login to the database, after the DBA grants permissions to you.

In this example, DBA has to give the following permissions to sales schema.

```
system@SALESP> exec
dbms_java.grant_permission('SALES','java.net.SocketPermission','
123.456.78.90:15000','connect');

PL/SQL procedure successfully completed.

system@SALESP> exec
dbms_java.grant_permission('SALES','java.net.SocketPermission','
123.456.78.90:15000','resolve');

PL/SQL procedure successfully completed.

sales@SALESP> select get_product_price('K1-GT-DESKTOP-R42') from
dual;

GET_PRODUCT_PRICE('K1-GT-DESKTOP-R42')
--------------------------------------
                              1879.83
```

SCENARIO 2: You are required to Create, Rename or Delete directories and sub-directories on your host Operating System from within your PL/SQL program. You also need to execute other operating system commands to get a list of files in a given directory, and find out if a given file is "Text File", "Binary Executable" or a Directory. Let us write a generic program that can execute Operating System commands from PL/SQL.

OperSystem.sql

```
CREATE OR REPLACE AND COMPILE
JAVA SOURCE NAMED OperSystem AS

import java.io.*;

public class OperSystem {

   public static int executeCommand (String commandLine) {

      Runtime rt = Runtime.getRuntime();
      int retCode = -1;

      try {

         Process p = rt.exec(commandLine);

         int bufSize = 2048;
         BufferedInputStream bis = new
BufferedInputStream(p.getInputStream(), bufSize);
         int len;
         byte buffer[] = new byte[bufSize];

         while ((len = bis.read(buffer, 0, bufSize)) != -1)
            System.out.write(buffer, 0, len);

         retCode = p.waitFor();

      } catch (Exception e) {
         e.printStackTrace();
         retCode = -1;
      }

      return retCode;

   }
}
/

show errors

CREATE OR REPLACE FUNCTION execute_host_command
(in_command_line in varchar2)
RETURN NUMBER AS LANGUAGE JAVA
NAME 'OperSystem.executeCommand(java.lang.String) return
integer';
/

show errors
```

Create the Java Function and the Call Specification together by executing the above script at the SQL*Plus prompt. Now you can execute host commands from SQL or PL/SQL, using the execute_host_command procedure.

```
sales@SALESP> set serveroutput on size 1000000
sales@SALESP> exec dbms_java.set_output(1000000)
PL/SQL procedure successfully completed.
sales@SALESP> variable ret_code number;
sales@SALESP> exec
:ret_code:=execute_host_command('/usr/bin/pwd');
/orahome/product/8.1.7.2/dbs

PL/SQL procedure successfully completed.
```

For the above example to work, your DBA needs to grant the following permissions to you. Note that you have to decide on the level and extent of permissions based on your requirements. For example, you may want to replace "/usr/bin/*" with "/usr/bin/mkdir" in the statement below, if you want permissions only to create directories.

```
system@SALESP> exec dbms_java.grant_permission ('SALES',
'java.io.FilePermission', '/usr/bin/*', 'execute');

PL/SQL procedure successfully completed.

system@SALESP> exec dbms_java.grant_permission ('SALES',
'java.lang.RuntimePermission', '*', 'writeFileDescriptor');

PL/SQL procedure successfully completed.
```

SCENARIO 3: You want to use the value of E, the base of the natural logarithms, in one of your packages. You don't have a readily available constant defined within the scope of Oracle PL/SQL. One of the options is to define a Constant within your package. However, the following solution using a Java Function, is easy and reliable.

NaturalLogarithm.sql:

```
CREATE OR REPLACE AND COMPILE
JAVA SOURCE NAMED NaturalLogarithm AS

public class NaturalLogarithm {

 public static Double getE () {

 return new Double(java.lang.Math.E);

    }

}
/

show errors

CREATE OR REPLACE FUNCTION get_e RETURN number AS
```

```
LANGUAGE JAVA NAME
'NaturalLogarithm.getE() return java.lang.Double';
/

show errors
```

Create the Java Function and the Call Specification together by executing the above script at the SQL*Plus prompt. Now you can access the value of E from SQL or PL/SQL, using get_e function.

```
sales@SALESP> select get_e from dual;

     GET_E
_____
2.71828183
```

SQLJ

You can embed static SQL statements in your Java Code using SQLJ, thus reducing complexity of writing JDBC statements. This language is very similar to Pro*C. Also, as SQLJ performs pre-compilation of static SQL statements, predetermination of SQL types, and uses bind variables to interact with Oracle. It is faster than JDBC.

Here is an example program to display Job History of a given employee.

JobDetails.sqlj:

```
import sqlj.runtime.*;
import sqlj.runtime.ref.*;
import oracle.sqlj.runtime.*;
import java.sql.*;

#sql iterator JobDetails (
   String job_title,
   Date start_date,
   Date end_date);

public class JobHistory {

 public static void display(int empNo) {

   Connection con;
   String dbURL = "jdbc:oracle:thin:@orcl.world:1521:ORCL";
   String userid = "scott";
   String passwd = "tiger";

   try {
      con = DriverManager.getConnection(dbURL, userid, passwd);

      String fullName;
      #sql { SELECT first_name||', '||last_name INTO :fullName
FROM employee_master WHERE emp_no=:empNo };

      System.out.println("Employee Number : "+empNo);
      System.out.println("Employee Name    : "+fullName);
      System.out.println("————————————————-");
      System.out.println();

        JobDetails jobDetails;
        #sql jobDetails = { SELECT job_title, start_date, end_date
FROM job_history WHERE emp_no=:empNo };

        System.out.println("Job Title \tStart Date \tEnd Date");
        while (jobDetails.next()) {


System.out.println(jobDetails.job_title()+"\t"+jobDetails.start_
date()+"\t"+jobDetails.end_date());

        }
        jobDetails.close();
        con.close();

      } catch (SQLException e) {
        e.printStackTrace();
   }
  }
}
```

Once you have SQLJ source code ready, translate it into Java Source Code, Classes and Profile files using sqlj executable.

```
sqlj JobHistory.sqlj
```

Now, load the Java Source Code and Profile files into the Oracle Schema, using loadjava executable.

```
loadjava -resolve -user scott/tiger@ORCL JobHistory.java
JobHistory_SJProfile0.ser
```

Note: When using loadjava to load Java Source Code, it is always a better idea to use the resolve option. This way, you will be able to rectify any errors from unresolved references at the time of loading the Java into the database.

```
scott@ORCL> exec print_job_history (5203);
Employee Number : 5203
Employee Name   : JOHN, WILLIAMS
————————————————————-
Job Title    Start Date     End Date
Technician   2001-01-15     2001-06-25
Supervisor        2001-06-262003-05-13
Store Manager2003-05-14     null

PL/SQL procedure successfully completed.
```

Boosting the Performance of Your Java Stored Procedures

As Java procedures need to go through Oracle JVM during each execution, there will be certain degradation in the performance. However, performance can be greatly enhanced by native compilation and instance parameter tuning.

Using JServer Accelerator

The JServer Accelerator converts Java Source or Java Classes into Operating System dependent native libraries, which are loaded dynamically into Oracle JVM during run-time. This bypasses interpreter and speeds up execution of Java procedures. The Core Java Packages such as java.lang are installed with Native Code under $ORACLE_HOME/javavm/admin directory.

To use Java native compiler, load the Java Classes or .jar files into the schema using ncomp executable, available in $ORACLE_HOME/bin directory.

```
ncomp -user scott/tiger -force EmployeeApplicationClasses.jar
```

Setting Correct Database Initialization Parameters

Database Initialization Parameters used to start your database instance play a key role in the performance of your Java Stored Procedures.

SHARED_POOL_SIZE: Shared pool memory is used when loading, resolving, creating call specifications, and checking access control of the loaded classes. The class loader uses an average of about 8 KB for each loaded class.

JAVA_POOL_SIZE: When calculating the value for this parameter, keep two things in mind. First, how many classes do you want to be cached and shared at any given point in time? Note that the class loader space is shared among all sessions connected to the database. Second, how many Objects will be created for all the shared classes? Note that the object space is private to each session. So you need to estimate the number of instances of each class.

JAVA_MAX_SESSIONSPACE_SIZE: If a session uses Java memory in excess of this limit, for all the private objects

that it has instantiated, Oracle terminates processing and sends out-of-memory errors to the invoking session. Thus you will have control over the maximum space that a session can occupy in your Java Pool.

Oracle Supplied Java Packages

Oracle has greatly extended Database Functionality by using Java in a number of Oracle supplied packages, including XML Development Kit (XDK), Ultra Search, interMedia and OLAP. All these packages have PL/SQL call specifications, so you can access them from either SQL, PL/SQL or Java.

Enhancements in Oracle 10g (10.1)

Support for Java 1.4.1

In this release, Oracle JVM has been upgraded to J2SE 1.4.1.

Native Java Interface

This feature enables client-side and middle-tier Java applications to invoke the Java Classes directly without the necessity of a Call Specification, thus providing simplified application integration.

JDBC and Web Services

JDBC drivers can now support varray enhancements, LONG-to-LOB conversions, INTERVAL DAY TO SECOND datatype. The thin JDBC driver is improved to support PL/SQL index tables, and passing, retrieving, and registering parameters by name. Loadjava performance has also been dramatically improved.

Java-in-Database Web Services

This feature renders Java Stored procedures as Web Services. Java portability allows the partitioning of these classes between the middle-tier and the database.

Conclusion

Integration of the Oracle database with Java VM enables creation of portable, powerful, database-independent persistent logic in the database which improves application scalability, performance, flexibility, and maintenance.

It is time you explore the possibilities of using Java in your Oracle databases for tasks that you cannot or have difficulty accomplishing using PL/SQL. Java is also a powerful language that can be used in XML, .net, Web Services and beyond.

One last idea: If you are struck with some Oracle bug in SQL or PL/SQL, you can try replacing that portion of code with a Java Stored Procedure. Drop me an email, if you use this idea! ▲

*Krishna Kakatur (**kakatur2002@yahoo.com**), an Oracle 9i Certified DBA and Developer with Atlas Software Technologies (**http://www.atlassoft.com**), is currently working as a consultant at Sun Microsystems. He has more than 10 years of experience using Oracle Tools and Technologies.*

Don't Go inSANe

By Mogens Nørgaard

This is an interview with James Morle, who is not only a member of the OakTable Network, but also a close friend and brilliant adviser on anything from Oracle over RAC, Linux, and records with AC/DC. James is also the author of Scaling Oracle8i, which is one of the very few Oracle-related books on the market that actually contains real research (Cary Millsap's book, Optimizing Oracle Performance is another good example).

I just bought 200 copies of James' book. Since the number on the cover is "8", they came very cheap, and all the stuff in the book is still valid, so it's a great gift for my customers (and all my children and ex-wives will get one, too, for Christmas).

I first met James at Oracle World in Berlin in 2001, where Anjo Kolk introduced us—and thank you for that! Since then we have had a lot of fun, James has written the Sane SAN paper, the Unbreakable paper, and much more. We also did the BAARF Musical together. James plays guitar and drums, has a 5-node RAC cluster, a SAN and fibre-optics in his home. He's crazy.

James never has enough time to write articles, so I thought I'd ask him a bunch of questions regarding RAID, and then publish it.

Enjoy!
Mogens

James, what is your opinion today, 1st of April 2004, of RAID-5?

My opinion has not changed since I first came across RAID-5 in the early '90s. I think that RAID-5 was devised at a time when disk technology was relatively expensive, and is an interesting concept in CS terms. However, I think that the street logic about where to apply RAID-5 technology is not adequately reflecting the needs of many enterprises.

Why do you think vendors like IBM (with Shark), HP (EVA) and Hitachi are selling RAID-5 systems as their default, or indeed sometimes as the only, option?

I think this issue highlights the source of much of this erroneous logic. Ultimately, this is a sales-led option, not a requirements-led option. The whole storage industry seems to have become obsessed with dollars-per-GB metric, effectively ignoring the IOs-per-second metric.

This is the equivalent of expanding all the highways to fifty lanes and ignoring the issue of how cars get on and off the road. The highway can now hold many more cars, but getting them on and off takes the same amount of time. In driving down the $/GB, the vendors have been forced to line up behind RAID-5 as the only way to keep the price down.

Has the price of disks actually fallen? If yes, then why does RAID-1+0 or -0+1 end up looking more expensive than a RAID-5 solution?

James Morle (left) and Mogens Norgaard announcing the BAARF Party at a meeting in the UKOUG UNIX SIG in Birmingham the morning after they got the idea.

First of all, let's stop even referring to 0+1. It is a solution that requires the same hardware as 1+0, at the same cost, but has a significantly worse failure profile. The performance is comparable between them, so why should we even think about using 0+1?

Now that's off my chest, we can return to the original question. Yes, the price of disks has fallen, both in terms of $/GB *and* the unit price (and thus the $/IO). This statement includes ATA drives and SCSI and Fibre Channel drives. In fact, the differential between the consumer ATA format and the 'Pro' SCSI and FC drives is now in the region of 2-3:1, a big improvement over previous years.

RAID-1+0 will always be more expensive compared to RAID-5, when comparing simple storage capacity. RAID-1+0 requires twice as much physical storage as logically required, whereas RAID-5 requires only one additional drive to cater for the storage of the parity information (but the actual location of the parity is distributed among all the drives).

Therefore, RAID-5 actually looks even more attractive if you implement very wide ranks of RAID-5. Nothing in life is free, however, and the write penalty and failure landscape deteriorate in direct proportion to the number of disks in the RAID-5 rank.

Do you also (like me) see mirrored RAID-5 systems being sold? What is the logic behind this?

The logic is simple. RAID-5 is not as resilient as RAID-1+0, and the mirroring is there to protect against the failures. Of course, the cost of doing this takes mirrored RAID-

5 beyond that of RAID-1+0, and the write penalty remains. The mirroring protects against running in degraded mode when a failure occurs, and provides protection against a secondary failure.

Has the famous RAID-5 write penalty for small writes (traditionally 4 I/Os versus 1 for non-RAID-5 systems) been improved as claimed by some storage vendors?

No, not from my observations. The logic used by the vendors is that the cache provides enough buffering of these small writes to allow the physical writes to occur as 'full-stripe writes'. Full-stripe writes should not occur the write penalty, because the whole stripe can be written without being preceded by the reads required to recalculate the parity information.

In fact, I can think of at least one situation where the cache could make the writes worse, instead of better. Consider the case where a small write is issued to a RAID-5 array. This write must be cached to avoid the write penalty, but depending on how the cache is implemented, this could result in many reads being issued just in order to refresh the cache lines for each device. If there are seven drives in the array, this could equate to 7*32KB reads in order to satisfy a 4KB write. This is not for reasons of parity calculation, but rather for reasons of cache coherency. The parity must then be calculated for this larger quantity of data!

In the case of small sequential writes, this overhead does not lead to a great deal of additional reads for cache coherency, as the cache information can be reused by subsequent writes. For scattered writes, there is a good chance each one will require this full set of larger reads each time.

Finally, this raises a good point about write cache efficiency. If many cache lines of a minimum size are needed to be retained to satisfy a single write, this means that the cache size needs to be relatively that much larger in order to provide the same benefit as RAID-1+0.

In my mind, I see cache as being good at handling spikes and handling repetitions. Are there other reasons for having cache? Repetitions mean waste, don't they? If yes, are spikes then the only reason for having cache?

There are two reasons for maintaining a cache, whether it is in a SAN or anywhere else: reading and writing. The two cases are different. In the write-case, non-volatile cache is an effective way to handle spikes, and to combine small writes into more efficient larger writes.

Like any kind of buffering, it is ultimately providing a time-based protection against waiting for the physical device. If the underlying disk can only deliver 50MB per second, and you are writing at a sustained rate of 100MB per second, your cache will simply fill, slowing the system down to the physical disk speed once more.

In the case of reads, cache is not providing much protection against spikes, mostly providing repetition benefit. I have fairly strong opinions in this area when using a SAN in conjunction with an enterprise database (such as Oracle) over the top.

Due to the fact that Oracle, for example, is already operating an MRU-based cache before the I/Os are sent to the SAN, the physical I/Os that are eventually requested after a miss in the Oracle cache are almost devoid of any easily cached pattern.

This white-noise of I/O requests makes it very difficult for the SAN cache to pick up any kind of spatial locality of reference, leaving mostly just the temporal locality hits based upon the cycling time of the Oracle cache.

For this cache to be effective, it needs to be orders of magnitude larger than the Oracle cache would need to be for the same effect! This phenomena is very well documented in CS circles as the impact of a multi-stage hierarchical cache. It's the reason why the L1 cache of a CPU is small (like 16KB), and L2 and L3 caches are in the region of megabytes. It's also the reason that the cache in your SAN is many gigabytes before it is very effective.

Have you been able to persuade customers to switch from an installed RAID-5 system to a RAID-1+0? If yes: What arguments did you use? Did the solution become more expensive?

Most of the time, this is only possible as part of an initial implementation, or a platform migration. It is normally relatively easy to persuade customers of the negative aspects of RAID-5 if they are already running that solution!

My arguments normally start from the basis of theory, but very quickly turn into scientific tests that return real data. In real terms, when the number of I/Os per second AND the storage requirements are taken into account, RAID-1+0 need not be a great deal more expensive than RAID-5.

When the write throughput is taken into account, RAID-1+0 can very quickly become a cheaper solution than RAID-5! And when the costs to the business are taken into

account for the failure scenarios, RAID-5 has real problems looking like the best solution.

How many reads can you perform with a RAID-1+0 system with, say, four mirrored disks (total of eight disks) that can each handle, say, 100 I/Os per second - 400 or 800 or somewhere in between? And how many writes?

The theoretical read rate of this array is 800 reads per second. The reality is that it could be a greater or lesser number than this, depending upon the actual read requests issued. For example, using the preferential read policies found in most RAID-1 implementations, the number of seeks could be substantially reduced, thus increasing the number of I/Os per second the array can support way beyond the nominal 100 quoted.

Writes are a different story, because the writes must go to the same location on both drives. The number in this case is close to 400, depending on the implementation.

Has anything regarding RAID technology changed since you wrote your book Scaling Oracle8i, perhaps even to the point where you'd like to rewrite or add to the 20 pages on storage systems?

Not really from the technology perspective. I have been consulting with a wide spectrum of clients since the publication of the book, however, and I think there would be a slightly different spin if I were to rework this chapter. I tried to write that chapter in an objective, mostly non-opinionated way; I think I would approach it in a more strongly opinionated way on a second run.

This reflects the thought process behind BAARF—I think the message needs to get out there so that people stop making the same mistakes over and over again!

It seems that the total price of a SAN is much, much, much bigger than the price of the disks put into it. Are there any SAN/NAS solutions available that actually reflect e.g. the very low price of ATA disks?

The problem is that the cost of entry of these systems is very high, and once all the cache memory is applied, there is very little left in the pot for the drives. The sales pitch will assert that the cache will save your life—don't EVER believe this philosophy.

For a scientific life of certainty, configure the SAN from the disks upwards, leaving the cache spend until last . . . ATA disk solutions are starting to get to market, and can represent good value for money. However, they are not generally aligned with performance systems, rather with high capacity near-line storage.

Subsequently, they often contain relatively slow (sometimes 5400 rpm), high density drives, such as 250GB drives. I am hoping that once Serial Attached SCSI becomes a reality, with the flexibility this affords of using SATA or SAS drives, the price of storage systems could drop significantly depending on the configuration.

Is it true that ATA disks are both cheap, good and can be part of a SAN? If yes, why are they cheap and good at the same time?

Parallel ATA is not very suited to a SAN, because of both electrical and implementation deficiencies. The Serial ATA (SATA) technology addresses most of these deficiencies, and could provide a strong solution. Also, ATA drives have historically not been manufactured to the tolerances and lifecycle requirements of SCSI or FC drives, thus making them theoretically less reliable.

Again, this is a trend that has now started to change, in the most part because of blade servers adopting ATA technology.

Do you have a SAN yourself in your home office?

Err, yes. It is part of my three (soon to be 4-5) node RAC cluster . . . ▲

Soon after this interview was conducted, James flew back to England from Copenhagen, with the expensive national SAS airline of course, losing his guitar in the process. The last phone text message I got from him only said: "Permission to explode?", then silence.

Note: *Membership of the BAARF party is free—go to* **www.baarf.com** *and see for yourself the names of people who have signed up as party members.*

James can be reached on **James@Morle.com** *and I can be reached on* **mno@MiracleAS.dk**.

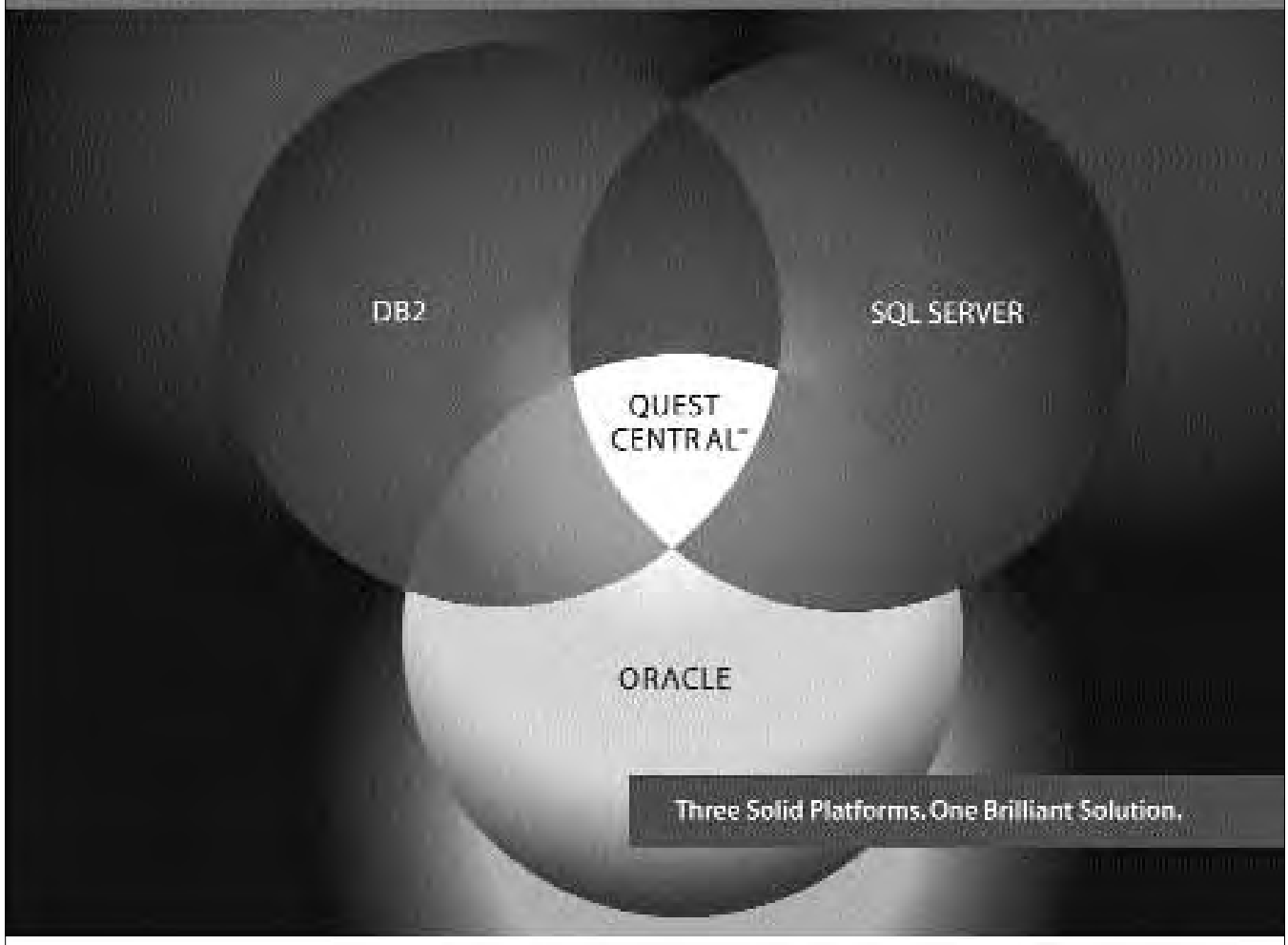
Mogens Nørgaard has over 17 years experience working with Oracle database technology. He was with Oracle Support in Denmark for 10 years where he worked as an RDBMS Analyst, Manager of RDBMS Support and finally Director of Premium Services. One of Mogens' greatest accomplishments at Oracle was to become the first manager to demand his team of about 40 people, use the YAPP method of database diagnosis and tuning. When Mogens left Oracle, he cofounded Miracle A/S, where he also serves as Technical Director. Miracle A/S provides consulting, support, and training on Oracle and SQL Server technologies in Maaloev, Denmark. On a more personal note, Mogens and his codirector, Lasse, hope to use the profits from Miracle A/S to start Miracle Breweries A/S. They believe in the importance of competition to add quality to the Danish beer industry.

THANK YOU

A big "thank you" to NoCOUG board member Vilin Roufchaie for spearheading NoCOUG's Training Day on Oracle Peformance Optmization. The event took place on April 8th and was a great success. Participants are already asking for more! ▲

Gaja Vaidyanatha (left) and Vilin Roufchaie at NoCOUG's recent Training Day.

(APPLICATION CONFIDENCE)



You work in a heterogeneous environment. Your database management software should, too. With Quest Central you can manage more critical databases with astonishing ease regardless of platform. It delivers comprehensive production management from administration to diagnostics, analysis, space management, SQL tuning and more — all with built-in expertise and a single intuitive, console-based interface. By Quest Central, it's not just for Oracle anymore. Download "Surviving in a Multi-database Environment," at **www.quest.com/nocoug** and find out how Quest Central changes database management forever.

QUEST SOFTWARE®

QUEST CENTRAL™: MANAGE MORE DATABASES TODAY

Many Thanks to Our Sponsors

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

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRONTEXACO

ORACLE CORP.

PG&E

Thank you! Year 2004 Gold Vendors:

- Churchill Software
- Database Specialists, Inc.
- EMC²
- Golden Gate
- LECCOTECH
- Mission Critical 24/7
- Oracle Corporation
- Quest Software, Inc.
- Veritas

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

TREASURER'S REPORT

Judy Lyman, *Treasurer*

Beginning Balance		
January 1, 2004		**$ 47,648.63**
Revenue		
Membership Dues	14,183.69	
Meeting Fees	560.00	
Vendor Receipts	11,905.00	
Training Day	3,145.00	
Advertising	---	
Interest	21.91	
Total Revenue		**$ 29,815.60**
Expenses		
Regional Meeting	618.73	
Journal	5,694.39	
Membership	---	
Administration	---	
Website	75.00	
Board Meeting	550.89	
Training Day	1,161.87	
Marketing	25.00	
Miscellaneous	---	
IRS	---	
FTB Tax	---	
Total Expenses		**$ 8,125.88**
Ending Balance		
March 31, 2004		**$ 69,338.35**

Choosing the Right Mass Deployment Strategy for Oracle Database 10g Software

by Sudip Data

Supid Data

Typically a large enterprise consists of hundreds of servers spread over one or more data centers. Quite often these servers are meant to serve the same or similar applications and hence require same database software. Typical examples include the nodes behind a load balancer for workload management or nodes participating in an Oracle Real Application Cluster environment for high availability. Other examples could be the nodes across geographies serving identical regional needs for a multinational enterprise. The software in each of these nodes is usually expected to be the same in order to ensure identical functionality, not only to start with, but also over the lifetime of the application. Before these nodes are put into operation, the software needs to be deployed to all of them.

According to Carzaniga, et al.[1], software deployment process refers to all the interrelated activities, which represent the lifecycle of the application. These activities are related to the release, install, de-install, update, adaptation, activation, and deactivation of software. In this document, we confine ourselves to the installation of software bits to various nodes.

This document seeks to explore the solutions to challenges around deploying Oracle database software to the nodes and also the various factors that are taken into consideration. The document does not cover specific tools or commands in detail, though it is broadly prescriptive in terms of practices and processes. In terms of technical scope, it assumes and is restricted to the Oracle 10g database software.

Basic Considerations for Mass Deployment

Any enterprise that wishes to distribute software to multiple nodes has to strategize and put together the right model not only for initial deployment but also for supporting the subsequent lifecycle operations. The factors that may influence the strategy include:

1. The number of nodes

If the number of nodes are too few and are connected by a local area network, one may want to install the software on a 'network attached storage' and share it.

2. Distribution of nodes

Do the nodes reside locally in the same time zone or do they need different time zones? Based on this one could decide if the deployment can be accommodated under the same change window.

3. Network and Hardware Topologies

Network topology is important in a suite or multi-tier deployment. The 10g database is a single-tier product does not fall in this category. The hardware topology could be important, especially for ASM (Automatic Storage Management) and RAC (Real Application Cluster) configurations. We will however, assume a simple topology in the course of this discourse.

4. Heterogeneity of Platform

Do the nodes have the same platform or different platforms? Is the software compatible with the hardware and operating System population?

5. Languages

Do the nodes have the same national language environment?

6. Ease of deployment and rollback

What tools are available to deploy software to multiple nodes and what is the rollback support in case the operations fail?

7. Size of software and bandwidth

Do the bandwidth and network usage support distribution of software of a particular size?

8. Frequency of updates

How frequently does the software need to be updated and hence redeployed either fully or incrementally?

The answers to the above questions may guide the administrator to choose one deployment strategy over another. In the remaining part of the document, we will explore some of the techniques that can be adopted as a part of the strategy.

[1] Carzaniga, A. Fuggetta, R.S. Hall, A. van der Hoek, D. Heimbigner, A.L. Wolf. "A Characterization Framework for Software Deployment Technologies". Technical Report CU-CS-857-98, Dept. of Computer Science, University of Colorado, April 1998.

Dimensions for Deciding Deployment of Oracle Software

There can be several key dimensions on which the Oracle database deployment can be structured. These dimensions are not orthogonal and often operate in conjunction. Assuming that the same software needs to be 'consumed' by multiple nodes, there method of deployment can be thought of along the following dimensions and sub-dimensions.

- **Number of deployment destinations:** Oracle database software can either be shared or installed individually. If the software is not shared, then the mode of replication needs to be considered.
  - **Mode of replication:** If the software is not shared, they need be distributed from a common source. The operations that result in final 'clones' can result from a physical bit by bit copy or a logical set of operations.
- **Mode of installation:** Oracle database software can be installed either in interactive or silent mode.
- **Point of initiation:** In case of most data centers, the decision to deploy software to nodes is usually a centralized and hence software is 'pushed' at a specific change interval. However, there could be specific cases where the decision is federated and hence nodes 'pull' software from a depot. A typical example is a subsidiary of a multi-national enterprise that would deploy software from the corporate IT staging area. It is not necessary that such a subsidiary be the leaf node in the software distribution tree. It could again act as a central server to 'push' the software to other regional data centers

Choosing the Number of Deployment Destinations

Shared versus Individual Installations

The first consideration when an enterprise has multiple consumers of same software is whether to have shared versus individual installations. The obvious way to simplify the mass deployment problem is by having multiple nodes share the same software. With the advancements in storage technology it is common to have a Network Attached Storage acting as a File Server. The nodes can mount the file systems through NFS and use the software. In case of clusters, one can also implement Cluster File System (CFS)[2] to share the software. The advantages of sharing are obvious: It allows centralized, single point management and eliminates the issues around multicasting software to many nodes. It also means that 'inventorying' the software becomes so much simpler.

But it also means that the software would suffer from inherent availability issues. When the systems need to be patched or upgraded the relevant services on all the nodes have to be stopped.

A reasonable low-downtime compromise would be to clone before the upgrade and upgrade the software 'out-of-place'. Once the software is upgraded the nodes can be 'switched over' during a convenient 'blackout' interval, to use the new software, and thereby suffer minimal downtime only during the 'switchover'.

[2] OCFS Version 2 and onwards will support Shared Homes.

For effective shared deployment of software, important considerations include whether software subscribes to a file system layout policy that enables cross-system sharing of software and whether the installation procedures are organized into one-time install and per-node configure phases. In case of Oracle 10g database software, the per-node configuration may relate to instance, Oracle Net (previously known as SQL*Net), and OC4J applications.

General conclusion

It makes sense to share the software home if the nodes are few in number, share the same change window and are in close proximity with the storage (it makes little sense today to share software across a wide area network).

Choosing the Mode of Installation

The Interactive Installation Approach

Oracle provides an interactive GUI based utility called Oracle universal Installer that is used to install the database among other products. Though the GUI based interviews offer user friendliness in terms of usability, it is not scalable for mass installs since administrators would have to repeat the tasks over and over again. Since this install is interactive and cannot be done unattended, it consumes a lot of administration time and effort without yielding any major benefit to the end goal. The interactive mode, however, can come handy for the first time install selectively choosing components for the 'gold image'. The interactive install can also be used in 'record mode' to capture user responses in a response file that can used with scripts to deploy software in a silent manner. The silent install approach is discussed in the next section.

Scripted Silent Install Approach

Oracle also supports silent install of database products. The silent installation is has to be driven by install time inputs provided via a response file (the inputs can also be provided as command line variables but that is rarely practiced because of obvious inconvenience). Oracle 10g and onwards, the silent install can also chain the configuration assistants in silent mode and the whole set up can be done completely unattended. The generation of response files is a one-time affair that can be aided by the interactive install in 'record mode' as described earlier. The variables of the response file can also be instantiated through simple scripting languages like Perl or Python, a technique that can be used to generate the instance specific configuration variables through some external logic.

The silent install itself can be scripted and can be scheduled by the operating system or a management tool. The silent install can also be chained with installation of other products (including those of Independent Software Vendors) and hence provides a complete end-to-end solution. The response file can be staged centrally with specific access controls so that the product components to be installed are centrally decided and controlled.

Though silent install helps administrators to scale during initial deployment, it does not help in deploying already patched software or help targets keep up in sync with the

source over the lifecycle. In most data centers, the software needs to be upgraded and patched to attain a production quality. That's why cloning proves to be a more effective technique for deployment.

General conclusion

Interactive installation is useful for initial setup in test environments and for creating response files. For large-scale deployments, the silent installation is the more scalable option.

Choosing the Mode of Replication

Physical Cloning

Unless of course, the applications on different nodes demand heterogeneous software versions data centers typically, do not install software to all the nodes. They usually install, patch and test the software on a single node and only when the software is well tested and approved by the Engineering Department, they mark it ready for mass deployment. The 'build and test' at source ensures standardization as well as quality. Some organizations also strip down the software by removing unnecessary components and directories while adding some of their own application files. The final outcome of this process is usually a 'standard, proven and approved software' for the whole enterprise, typically referred to as a 'gold software' that is staged centrally in the enterprise within its software library and deployed using various distribution mechanisms or tools. This method is typically referred to as 'physical cloning'. Before we delve into the various aspects of physical cloning, let us classify and define Oracle software files into two major types:

Immutable files: Files that do not typically change over the lifetime unless the software is patched or upgraded through official releases. This type includes archives (jars, wars, etc.), object libraries and binary executables. Immutable files can be supplied as a part of the product (object files) or derived (like the executables which are generated after relinking) during installation. Message files and Oracle provided scripts are not binaries, and yet remain unchanged.

Mutable files: These are files that can be modified during the lifetime of the installation. This typically consists of configuration files like init.ora, tnsnames.ora, emoms.properties etc.

Once cloned through physical cloning, the immutable files (binaries, libraries, etc.) would be identical between the source and the target. Some mutable files may need modification to accommodate environment specific changes immediately after cloning; this is typically done through post cloning configuration operations. Physical cloning is definitely the deployment of choice in a clustered environment where the binaries on each node have to be the same to the last bit.

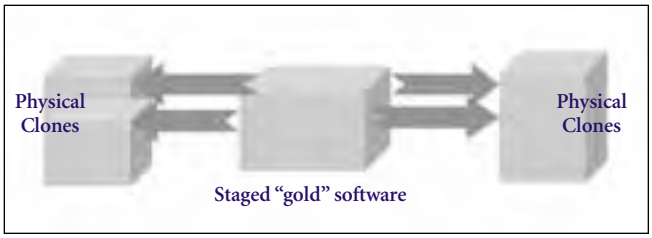
Cloning is also provided as an important feature in the Oracle 10g Enterprise Manager Controls and can be used to clone Oracle software if the nodes have Oracle agent up and running. It is highly recommended to use Oracle's cloning technique rather than Operating System backup and restore, since the cloning makes environment specific inventory changes that are specially important for subsequent management of the software home. In 10g, the backend APIs for cloning form an integral part of the Oracle Universal Installer.

In high-security distributed environments, mechanisms must be put in place to validate the transferred bits against the source through a combination of checksums and Public Key Infrastructure (PKI). Currently administrators have to use third party tools or mechanisms to secure the distribution of bits.

The physical cloning technique however suffers from some limitations:

Handling Localizations–Software for different geographies may have different language or Localization requirements. The localization is typically an install time decision and once installed, remains immutable over the life of the installation. This essentially means that for cloning, several 'local variants' of the same software has to be produced at source and deployed at specific targets.

Handling different platforms–Typically, nodes running similar application have same hardware and operating system. However in the 'not-so-common' cases where they are different, the cloning technique does not work. Here again, platform-specific variants of the same software have to be produced and staged at source.



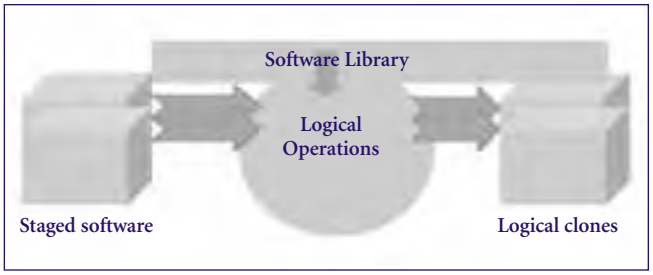
Logical Cloning

Assuming that the final state of the software is a function of the components chosen and the install environment, software on two different nodes can be functionally equivalent though the bits may be different. Similarly, it can be deduced that same bits do not necessarily guarantee same functionality if the underlying technologies are different. For example, the same bits may behave differently on two different implementations of the file system.

Though physical cloning is simpler, one may choose to do a logical cloning based on specific requirements. Logical cloning implies that the target nodes install the same set of components as but not necessarily the same bits. In this case, the pre-cloning operation simply takes stock of the Bill of Materials or high-level components that exist in the source. It then uses that Bill of Materials as a reference to deploy the appropriate software to the target. The software components and directives constitute what most deployment tools consider a 'package' (analogous to a cooking recipe which consists of ingredients and instructions). In brief, logical cloning implies cloning the operations rather than the physical bits. This technique can be employed to attain identical software levels between two heterogeneous

systems e.g. Linux and Solaris. Technically, logical cloning would include, among other steps the silent install method discussed earlier. It would also include the scripted upgrade and patch to bring the targets to the right level.

The logical operations like silent mode installation and silent mode patching can be scheduled and kicked off using the Enterprise Manager 10g Grid Control job subsystem. Third party tools like Opsware, Red Carpet, Tivoli and Asdis are also capable of distributing the software as rpm, MSI or other bundles and running the logical operations silently against the target nodes.



Irrespective of the method of replication, physical or logical, it is imperative that in order to build a 'grid' of identical systems, the method should be preceded by the same assertions of constraints (commonly known as pre-requisites) to ensure stability and predictability. Similarly the post cloning process can have basic component (in case of logical cloning) and checksum (in case of physical cloning) validations to ensure the integrity of the software.

General conclusion

Physical cloning is a highly scalable deployment mechanism that deploys trusted bits across the enterprise. In heterogeneous environments however, one must replay the same logical operations over the same set of components.

Replicating Incremental Operations

Once the software is cloned to the target, they are expected to remain stable unless another major upgrade occurs. However, in real life the software may need to be updated if say, a critical security patch is released. Complete re-cloning of the source after every such operation is not efficient, especially if the number of such targets is high or if the software volume is large. As such, the software patching cost in an enterprise is high, compounded if the deployment spans locations.

An efficient way to find the incremental changes is by comparing checksums at the source before and after the patch application and propagating the files whose checksums have changed. This can be thought of as Physical Incremental cloning.

A Logical Incremental cloning would mean replaying the operation at the target that changed the source itself. This is the preferred way that software on two different Operating Systems can be brought to the same level. Again it is presumed that the same operation on two 'sister nodes' would lead to similar functionality, if their previous states and environment match. Oracle Enterprise Manager 10g supports patching multiple targets at a time by staging and applying the patch(es) through the job subsystem. The patching engine, called 'opatch' supports full silent capabilities in 10g to enable easy scripting and scheduling.

There are some hardened enterprises that do not apply patches frequently. In some cases, it could be as infrequent as once every year or so. There could also be situations where the patch or the update touches a significant portion of the software (patchsets fall in this category). In these cases, a complete cloning of the database software is something that could be a better choice and can be borne by data center administrators.

It is also important that the distribution of bits be done at hours when the network is idle, especially if the final deployment targets are large in number. This would help avoiding intervention with network centric applications.

In reality, an enterprise can adopt a 'hybrid' strategy for deploying the Oracle database software. For initial deployments of 'base images' and 'major upgrades' it can use the physical cloning method. For incremental one-offs it can use logical patching operations. In Enterprise Manager 10g Configuration Management Pack, the 'Home Cloning' and 'Patching' applications implement this strategy.

Point of Initiation: Centralized distribution ('push') versus Federated retrieval ('pull')

Once the method of distribution has been settled, one needs to think on how to initiate the process. Depending on who initiates the file transfer it can be classified as 'push' or 'pull' (analogous to the 'put' and 'get' file transfer mechanisms, respectively). In most cases today, the source 'pushes' the software to a list of targets. Even if they don't, the directive to 'pull' is always initiated at the point of central management to avoid random downloads resulting in 'chaos'. Conceptually, therefore, centrally directed, time synchronized 'pull' is the similar to 'push' since the directive and the bits always flow from the source to the target. Administrators usually take care in scheduling the 'push' or the 'pull' during periods when the network load is conducive to such heavy traffic. In both 'push' and 'pull', the packets have to be encrypted, if possible signed, and trust

(continued on page 25)

Physical Cloning	Logical Cloning
Clones bits	Clones components and processes
Starts with an already engineered ORACLE_HOME	Starts with the software and a list of operations and parameters
Predictable, guarantees checksum match for immutable files	May yield the same end result
Not effective for different platforms or different locales	Effective for different platforms or different locales

Table 1:Physical versus Logical Cloning.

Isolating Performance Problems

by Chris Lawson

Database performance specialists don't have the luxury of assuming that a performance problem is always due to a database problem. In fact, it is quite common for Oracle to have nothing to do with the actual root cause of a performance bottleneck. Of course, if the database is "cleared from suspicion," the question then becomes, "Where is the delay coming from?"

When faced with mysterious performance slowdowns, it is very common to hear the vague explanation, "There is a problem with the network." Naturally, a good performance expert will not be satisfied with such ambiguous explanations. Instead, he or she will want to precisely isolate the problem by reviewing all traffic across the network. But how can this be done? How does one find out what is really going on in the network?

In this article, we illustrate a convenient way to capture, quantify, and summarize all data that is being transferred over a network. To do this, we will us a freeware "network sniffer" called "Ethereal."

Network Sniffing Using "Ethereal"

The DBA using a network sniffer can easily extract the "mysteries" of the network traffic to and from a database. But why such a crazy name for a network utility? Why the term "Ethereal?"

The term *Ethereal* is actually a good name for a network sniffer. For the majority of us who aren't network engineers, the network infrastructure might as well be the wispy substance of outer space that was called "ether" in years gone by.

The dictionary lists these definitions for *ethereal*:

- Immaterial, intangible
- Of or relating to the regions beyond the earth
- Celestial, heavenly
- Unworldly, spiritual, lacking material substance

Well, I must admit that our foray into the world of network data capture will not quite approach the "heavenly" realm. Nevertheless, I think our brief investigation will make analyzing network traffic a little less mysterious.

Setup

The Ethereal tool is freeware that can be downloaded from **www.ethereal.com**. As noted on their website, you should first download and install the *WinPcap* packet capture driver (they provide the link to download this utility). Then, you download and run the ethereal-setup.exe file. These steps are simple, and worked for me without a hitch.

Once the network sniffer is installed, it is easy to begin "sniffing." From the Ethereal top menu, select ***Capture/Start***. This will bring up the window shown here:

Start with this window to select options.

Most of the selections in the "Capture Options" window can be left at their default values. A few options, however, should probably be changed. For instance, I have found it useful to select the button, "Update list of packets in real time." With this button checked, the network traffic is continuously displayed, as opposed to waiting until your run is finished.

The "Filter" button is also important because it allows you to restrict your data capture to only that network traffic that is of interest to you. Most times, you will want to restrict your data capture to only those network packets going to (or from) a certain host. By entering a filter you will not be deluged with all network traffic related to routers or other non-interesting packets of data.

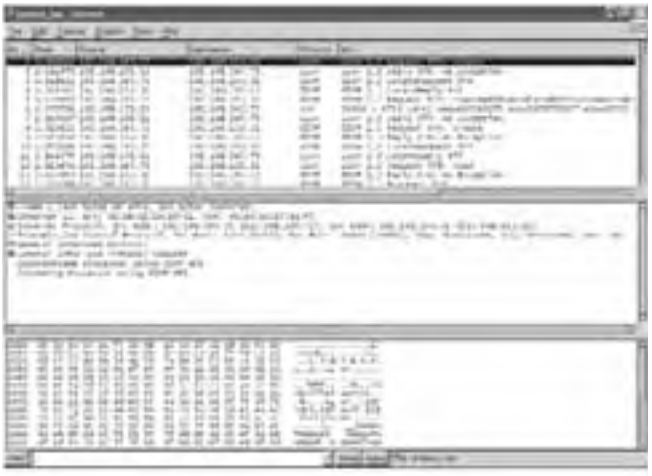
If you intend to use the same filtering criteria repeatedly, it is a good idea to save the filtering criteria in a file. In the example below, we are creating a filter called "Oursite_Filter." This filter employs the filtering command **host oursite.com**. The "host" command tells Ethereal to only display traffic to or from a certain host (in this case the server **oursite.com**.)

Use a "Capture Filter" to restrict captured data.

With a "Capture Filter" set to only gather data from certain servers, we eliminate all the unnecessary "chatter."

Sample Displays

Once you are satisfied with your capture options, simply select the "OK" button in the Capture window. This will



activate the network capture process. As Ethereal detects network traffic that fits with the filtering criteria selected, the data will appear in a window similar to this:

Remember that the data will only appear right away if you have selected the "update list of packets in real time" button. Otherwise, you will have to wait until you end data capture.

Once you have collected all the network traffic needed, simply select "Stop" in the data-capture window. Ethereal will stop capturing packets, and you can now continue to review the network data, save it for future use, or display a summary of the run.

As shown above, Ethereal divides the window into three panes which show slightly different perspectives on the data captured:

- ***Top Pane:*** Packet time, source, destination, and protocol
- ***Middle Pane:*** Summary of protocols contained in a given packet
- ***Lower Pane:*** Actual data transmitted in a given packet

In this example, we have highlighted the first packet, which happens to contain a "GIOP" message (related to IBM's Websphere application.). The middle pane shows that there were 342 bytes in this packet. The lower pane shows the actual data for packet #1.

Summary Information

After the desired information has been gathered into Ethereal, a summary of the network traffic is displayed when you select ***Tools/Summary*** from the top menu:

This summary is particularly useful if you are trying to quantify network traffic for a performance problem. For instance, the summary above shows that about 233 kb of data was transferred to or from the specified server over a total period of 4.4 seconds. Naturally, if there is only a small amount of information sent over the network, it is unlikely that the network is really a bottleneck.

The Summary window provides a concise overview of network statistics.

Checking out Oracle Traffic

It is interesting to watch the traffic during a simple SQL*Plus session. To do this, I started Sql*Plus and activated network sniffing using Ethereal. Then, I entered the simple query:

```
select 'hi there' from dual;
```

In the Ethereal screen capture window below, we can see exactly what information my workstation sent to the database server:

In the first packet of information, we see that Ethereal recognizes the protocol, "TNS" that is used by SQL*net. (TNS stands for *Transparent Network Substrate.*) The lower pane shows the actual text (both hex values and ASCII) for packet #1.

When we select the *second* packet, we can see what the database sent to my workstation:

In the screen capture below, notice that the database server has sent the headings in uppercase, followed by the actual "results" of the SQL query.

The network capture reveals an interesting fact about this simple SQL query. Pay close attention to packet #4, shown below. Notice there was actually some *more* text sent to my workstation that is never displayed! The screen

below shows that the database server actually sent the error message: ORA-01403, "no data found." Perhaps, this was due to a select from "dual."

Wrap-Up

These examples should make it clear how useful network sniffing tools can be. Whenever there is a question about the impact of network traffic on performance, it is very easy to identify and quantify the exact information that is being transferred—packet by packet. This information can be reviewed, summarized, and saved in a file for future examination.

With network sniffing tools, the DBA can identify *exactly* what is happening over the network. This in turn allows the performance analyst to isolate where the problem *really* is. The Ethereal network capture utility is one convenient way to find and eliminate performance bottlenecks. ▲

Chris Lawson is a DBA consultant. His recent book, The Art & Science of Oracle Performance Tuning *has received numerous 5-Star reviews, and is available on* **Amazon.com**.

Choosing the Right Mass Deployment Strategy for Oracle Database 10g Software *(continued on page 22)*

relationships have to be established between the source and the target, ideally though Public Key Infrastructure (PKI).

For desktop type environments, where Oracle client software need to be deployed to thousands of targets, a target initiated 'pull' can be thought of, since the delegation to users to manage their client mobile devices and laptops is quite customary.

General conclusion

There is little difference between 'push' and 'pull' technologies. The actual implementation depends on the IT environment, bandwidth and the software involved. For a large scale deployments of client software the pull model can be more useful.

Overall Conclusion

Oracle 10*g* makes software deployment easy and predictable. At the core of this capability is the Oracle Universal Installer framework. Even end data centers can utilize the framework to develop deployment solutions of their choice. The framework is flexible to handle different use cases and needs, some of which have been further facilitated by the Enterprise Manager job subsystem and tools. As per "Agent-based Software Configuration and Deployment" by Richard Scott Hall:

> *"Software producers have specific and separate concerns from the software consumer. In particular, software producers are concerned with software system packaging and release mechanisms that are simple, flexible, and extensible. After a software system is packaged and released, the software producer must have an effective mechanism to advertise the release in order to notify interested consumers of its existence. There is little benefit for the software producer if its customers, both current and potential, are unaware of its products and services."*[3]

This is why, the story does not end with the initial deployment. It ties in with Oracle Enterprise Manager 10*g* Grid Control whose agent will discover the new installation and its components during host collection and transfer the data to the Management Server for ongoing management. Though outside the scope of this document, deployment extends to end-to-end software configuration management that clearly could make Oracle 10g the software of choice. ▲

Sudip Data has more than ten years of industry experience and has been working at Oracle Corporation for the last eight years. In these years, he has assumed several responsibilities in Oracle Support and Development. He started his career in Oracle as a database kernel support specialist and provided onsite consulting for major customers across industry segments. In Oracle Support, he also worked as a member of the Diagnostics and Escalations team where he handled escalations from key customers like Amazon, EBay, GE, and Boeing to name a few. Later on, he moved into development of System Management Products as a Product Manager. Currently, he is working on the deployment framework that forms the core of 10g Software Configuration Management. The framework spans operations like installation, patching, cloning and configuration that constitute the software deployment lifecycle.

[3] "Agent-based Software Configuration and Deployment" by Richard Scott Hall, B.S., University of Michigan, 1990,M.S., University of Colorado, 1993.

NoCOUG Conference Reporter:
NoCOUG Members at Our Last Conference

Member Profile: Haniet Bourshrocku

Randy: ***Where do you work?***

Haniet: Santa Clara Valley Transportation Authority.

Randy: ***How are you enjoying the conference so far?***

Haniet: Today was really good! We skipped a year, budget issues, but we joined again. This is my first time back again, and today, I enjoyed it a lot. The keynote speaker this morning, Ken Jacobs, and also, the panel, "Ask Oracle" was great, and basically learning about 10g, it's my first exposure to 10g, and I've enjoyed that.

Randy: ***What release are you working with right now?***

Haniet: We are basically at 8.1.7, but upgrading to 9.2, and I'm working also on upgrading our OEM to 9.2.

Randy: ***How long have you been working with Oracle?***

Haniet: I'd say about 3 years, but we also have Sybase, SQL Server and stuff like that.

Randy: ***Do you have any suggestions on ways that we can improve this conference?***

Haniet: One thing I found is that it's really geared toward very experienced DBAs. I'm much more comfortable now, so thing are all making sense, but at the beginning it was very overwhelming, so you might be losing some entry level people. Some lower level, entry level information would help. That's what I would like to see.

Member Profile: Dan Kukla

Randy: ***And how long have you been a member of NoCOUG?***

Dan: Actually more than 10 years. I don't remember exactly how much longer it's been.

Randy: ***And are you enjoying the conference?***

Dan: I think it's a great opportunity. It's a tremendous value. It's still so incredibly cheap for the value that it provides.

Randy: ***What company do you work at?***

Dan: C A small biotech in Concord.

Randy: ***Who was your favorite speaker today?***

Dan: Probably Ken Jacobs. He's always entertaining to listen to. I had already heard about 10g, but it was still interesting.

Randy: ***What releases of Oracle are you running in your shop?***

Dan: All the way from 7.3 to 9*i*.

Randy: ***How many databases do you have?***

Dan: Five, if you include SQL Server databases. We have a mixed environment.

Randy: ***Are you running all Windows, or are you running UNIX?***

Dan: Mostly Windows. We have some development systems on Linux. No UNIX. Linux may be more important down the road for us.

Randy: ***Is there anything that you feel that we could do better with the conferences?***

Dan: I think you're doing a great job! I have really no complaints. Maybe transportation. If I could get to a conference by public transportation [from El Cerrito], then that's just perfect for me.

Member Profile: Glen Murphey

Randy: ***Have you been to a NoCOUG Conference before?***

Glen: Well right now, I am between jobs. This is my first time here at the conference. I just graduated from St. Mary's College. I have earned my Bachelors degree in Management. Now it's time to specialize.

Randy: ***Which seminar at this conference did you like the best?***

Glen: All of them have really been good, but the one that really held my interest was Database Security. The guest speaker, Noel, did an excellent job of holding my interest.

Randy: ***Will we see you at some more NoCOUG Conferences?***

Glen: You will.

Member Profile: Annita Auyang

Randy: ***Are you enjoying the conference?***

Annita: Yeah, very much! I learned quite a bit about 9*i* and 10*g* features.

Randy: ***How many conferences have you been to?***

Annita: I joined NoCOUG in January 2003, so I've been to five conferences.

Randy: ***What releases of Oracle are you working with now?***

Annita: Right now we're working with Oracle 8.1.7, but soon we're going to go to 9*i*.

Randy: ***How do you feel about NoCOUG as a way to network?***

Annita: It's really wonderful. You get to know the other DBAs, ask them questions, and make friends with them, and win prizes too.

Randy: ***And what did you win today?***

Annita: I won a USB light. I think that's very helpful, because I often use the laptop, bothering my husband while he's sleeping. So this light will dim it, so it will not bother him anymore. It will be a useful prize for me. ▲

Interviews conducted by Randy Samberg, Board Member, NoCOUG.

TECH TIPS

Current SCN

If you are using duplicate database command to refresh databases from production and are attempting an incomplete recovery, the following sql will help you determine the current scn on your production database:

```
Select dbms_flashback.get_system_change_number
from dual;
```

Thank you to Sumathy Thankam Panicker for the tech tips in this issue *Sumathy is a Peoplesoft/Oracle DBA and is currently working at City of Santa Clara. She is Oracle 9i, 8i, 8 certified. She has been working with Oracle for almost 10 years now (the past 6 years as a DBA) and can be reached at* **thankams@aol.com**. ▲

NoCOUG Spring Conference

Thursday, May 13, 2004 · Location: Lockheed Martin, Sunnyvale, CA

NOTE: Lockheed Martin's security requirements have changed this year and RSVP's must be recieved by May 6th for admitance. RSVP's are mandatory. Only US citizens or green card holders will be admitted. Please visit **http://www.nocoug.org** for additional security information as well as technical presentation details, directions to Lockheed Martin, and to submit your RSVP.

8:00 a.m. **Registration and Continental Breakfast**

9:00–9:30 **General Session and Welcome**

9:30–9:45 **Break**

9:45–10:45 **Parallel Sessions #1**
Track 1: *Choosing the Right Mass Deployment Strategy for Oracle Database 10g Software* by Sudip Datta, Oracle Corporation
Track 2: *Increase Java Application Performance with PL/SQL* by Joel Thompson, Rhino Systems
Track 3: *Oracle's Workspace Manager* by Daniel A. Morgan, University of Washington

10:45–11:30 **Break**

11:30–12:30 **Parallel Sessions #2**
Track 1: *10g - ASM (Automated Storage Management) and New Manageability Features* by Bert Scalzo and Lenka Vanek, Quest Software
Track 2: *A Closer Look at Oracle's Join Methods* by Hamid Minoui, Independent Consultant
Track 3: *Beginning SQL: Differences Between Oracle and Microsoft* by Les Kopari, Independent Consultant

12:30–1:30 **Lunch**

1:30–1:55 **Roundtable Discussions in Auditorium**

2:00–3:00 **Parallel Sessions #3**
Track 1: *A DBA's Perspective on J2E* by Debu Panda, Oracle
Track 2: *The Persistence of Memory (Issues)—The Shared Pool, Sorting and Other Nightmares* by Brian Hitchcock, Sun Microsystems
Track 3: *Oracle Backup and Recovery Essentials Every DBA Should Know* by Rob O'Brien, Veritas

3:00–3:30 **Break & Raffle**

3:30–4:30 **Parallel Sessions #4**
Track 1: *Develop, Deploy, and Manage Web Services with Oracle Application Server 10g* by Olivier Le Diouris,Oracle
Track 2: *Reducing Costs of Environment Copies Using Oracle Recovery Manager* by Rich Bernat, ChevronTexaco
Track 3: *The Rap on RUP: An Introduction to the Rational Unified Process* by Jeffrey Jacobs, Jeffrey Jacobs & Associates

4:45–?? NoCOUG Networking and Happy Hour

Cost: $40 admission fee for non-members. Members free. Includes lunch voucher.

Keep checking the NoCOUG website for the most updated Spring Conference schedule!

Session descriptions available at www.nocoug.org.

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

Cover photos by Jessica Schlecht.

NoCOUG
P.O. Box 3282
Danville, CA 94526

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
OAKLAND, CA
PERMIT NO. 1770

Thank you to Lockheed Martin, our meeting sponsor.