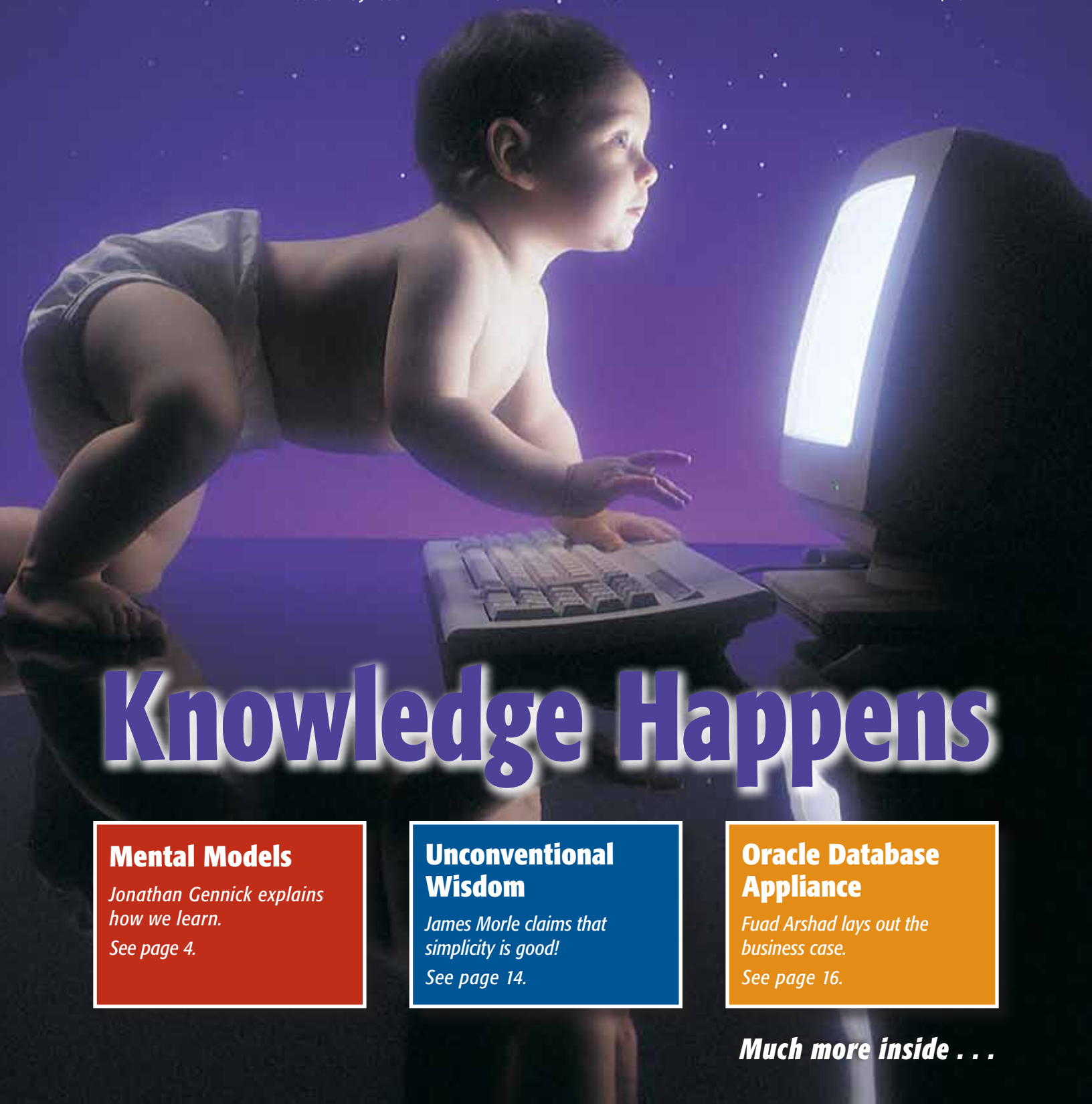
Official Publication of the Northern California Oracle Users Group

NoCOUG JOURNAL

Vol. 28, No. 2 · MAY 2014 $15



Knowledge Happens

Mental Models

Jonathan Gennick explains how we learn.

See page 4.

Unconventional Wisdom

James Morle claims that simplicity is good!

See page 14.

Oracle Database Appliance

Fuad Arshad lays out the business case.

See page 16.

Much more inside . . .

Professionals at Work

First there are the IT professionals who write for the *Journal*. A very special mention goes to Brian Hitchcock, who has written dozens of book reviews over a 12-year period. The professional pictures on the front cover are supplied by Photos.com.

Next, the *Journal* is professionally copyedited and proofread by veteran copy-editor Karen Mead of Creative Solutions. Karen polishes phrasing and calls out misused words (such as "reminiscences" instead of "reminisces"). She dots every i, crosses every t, checks every quote, and verifies every URL.

Then, the *Journal* is expertly designed by graphics duo Kenneth Lockerbie and Richard Repas of San Francisco-based Giraffex.

And, finally, Jo Dziubek at Andover Printing Services deftly brings the *Journal* to life on an HP Indigo digital printer.

This is the 110th issue of the *NoCOUG Journal*. Enjoy! ▲

—*NoCOUG Journal* Editor

Table of Contents

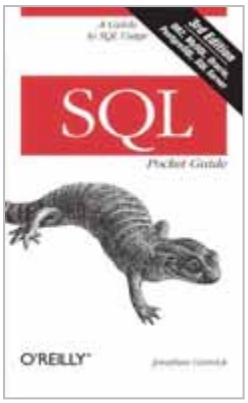
ADVERTISERS

Publication Notices and Submission Format

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

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

The submission deadline for each issue is eight weeks prior to the quarterly conference. Article submissions should be made in Microsoft Word format via email.



Mental Models

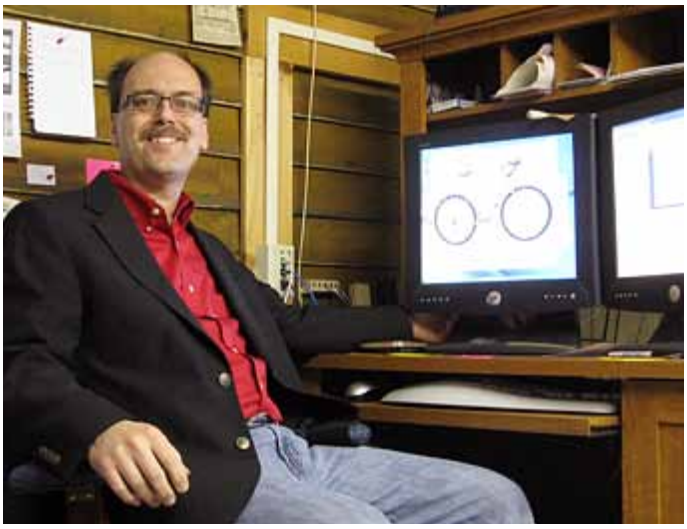
with Jonathan Gennick

Jonathan Gennick

Jonathan Gennick is an Apress assistant editorial director with responsibility for database topics. He is line leader for Apress's Oracle and SQL Server lines. He also publishes carefully chosen database books of a general nature. He maintains a keen interest in books across all lines that touch upon relational databases.

Jonathan, Jonathan, wherefore art thou Jonathan?

You might already know that I'm a published author, having written several books on Oracle Database, SQL and SQL*Plus, and PL/SQL. I've also written numerous articles for *Oracle Magazine* (beginning in 1999 and going through sometime in 2006).



Jonathan edits database books, especially on SQL Server and Oracle.

What do you do for a living today?

My day job is what is termed in publishing an acquisitions and developmental editor, or Ack-Dev for short. I'm constantly in search of writing talent and topic ideas, and work with authors to produce good books on what I hope are up-and-coming topics. My editorial output extends back over a decade.

And you live in a very small town?

Yes. I was very lucky in October 2000 to fulfill a lifelong dream of living "up north" by moving to the city of Munising in Michigan's Upper Peninsula (U.P.). The city name is pronounced "mew-nih-sing," with emphasis on the first syllable. The location is in the central U.P., on the south shore of Lake Superior.

Munising is a wonderful place to live. Nearby we have the Hiawatha National Forest, the Pictured Rocks National Lakeshore, and the Grand Island National Recreation Area, just to name a few. It's also a blessing to live in a small town with a sense of community where people know and recognize you as you go about your daily business. I like that aspect of Munising a lot.

But he'd rather ride his mountain bike.

Do you still drive the local ambulance?

Not anymore. I spent five years as a volunteer EMT with the Alger County Sheriff's Department and was also briefly on the search-and-rescue team. I learned a lot and grew a lot during those five years, but the ambulance service grew too. Eventually it became too much to try to balance the volunteer work against my day job. My last ambulance run was in January 2012.

I do remain active in the community, just in different ways. For example, I'm an active member of the Munising Bay Trail Network. We've spent two summers now building a small trail system in the hills above town. Now there is money for an excavator and a grant by which to build a handicap-accessible trail. I'm also involved in my local church as First Elder and still some-

times teach a weekly Sabbath School class when enough of the young people are home from college.

He lives in a small town in Michigan's Upper Peninsula.

And teaches a Sabbath School class for a group of kids at his local church.

What's with the jail uniform?

We were studying the apostle Paul in the class I was teaching at church. Paul was a prisoner for many years in Rome. There's a certain stigma attached to being in prison. It occurred to me that wearing a jail uniform to church might be a good way to get across to the kids I was teaching a bit of how Paul must have been perceived by society at large.

The sheriff loaned me the authentic article: a blaze-orange uniform with the words "Alger County Jail" across the back. I taught my class in that uniform. I was the deacon on duty that weekend, so I also ended up collecting the offering in my jail outfit.

How did you get into the writing biz?

I blundered into it. That left me with a sort of survivor's guilt for a while. At least two relatives had worked hard and yet unsuccessfully for years to become published, and I just fell into being published without even trying. So there was this sense for a while of a reward that was unearned and not deserved. I'll come back to that in a moment.

My break came through being involved in the community, in my case through a news group. I had been doing some PL/SQL programming and had joined a Usenet group related to that. I would ask questions and would try to give back by answering some as well. An editor from SAMS saw some of my responses, liked how I wrote, and asked whether I'd consider writing a book to be called *Teach Yourself PL/SQL in 21 Days*.

I worked my tail off to put together the proposal. I misunderstood the level of detail required and had every chapter in my outline broken down into three levels of subheading. Every evening for two weeks went into that proposal, and some weekend time as well. The level of detail and polish made my proposal stand out, and I was offered the book.

Then I got cold feet. SAMS wanted the book to be written in something like two months. I was apprehensive about writing 700 pages in 60 days, and rightly so! In the end, I sold the outline to SAMS and picked five chapters that I wanted to write myself; SAMS found other authors for the rest.

Perseverance. That and some attention to detail are two important qualities in nonfiction writing. SAMS didn't publish me because I was brilliant or the next Charles Dickens. They published me because they needed content, and I persevered and did what needed to be done to deliver it.

Later I came to realize that I had been preparing all during my career without really realizing it. I loved to write documentation and did so with abandon, leaving behind binders full of programming and process documentation everywhere that I had worked.

I also have a terrible memory. Relying upon my own documentation taught me a lot about how to write so that I could make sense of myself six months later, and I believe that has played into my success in writing books and magazine articles.

I almost threw away the opportunity. I thought that first email from SAMS was spam, and I had deleted it. That was back when not all email programs had a "garbage can." Mine did, and a few days later I dug that email out of the trash and responded. I shudder to think now what I would have missed out on had I not done so.

The State of the Industry

With so much free material available online, who buys books anymore? What better book on Oracle Database concepts than the free* Oracle Database Concepts *book from Oracle? What better book on data warehousing than the free* Oracle Data Warehousing Guide *from Oracle? What's the prognosis for the book publishing industry?

We in publishing ponder this issue a great deal. It has long been the case that a publisher's greatest competitor is not another publisher but the Internet at large.

Certain types of books tend not to be viable anymore. Syntax references are a good example. Fifteen or 20 years ago there would have been a market for something like an Oracle SQL Syntax Reference. Now the entire documentation set is online, and there's rarely reason to go anywhere but the ultimate source of truth—the vendor documentation—when looking for help purely on syntax.

What you won't find in a manual set, though, are books that distill an author's experience. Christian Antognini's book *Troubleshooting Oracle Performance* is an example. Chris brings into book form many years of experience, and he speaks freely about Oracle Database in a way that a manual writer cannot.

Then there are authors who artfully cut through large feature sets to deliver crystal clear understanding without overwhelming the reader with too many possibilities. Darl Kuhn's *Pro Oracle Database 12c Administration* is an example here. Darl covers database administration with a focus on the latest features and current practices. Readers who learn that "happy path" are in a good position to read about various alternatives and legacy features and syntax in the documentation set. Oracle Database can be overwhelming. Often there are multiple ways to accomplish something. The value an author provides can sometimes lie in knowing what to cover and what to leave out.

Sometimes a good author with passion for a topic will write to evangelize that topic. Look to the upcoming book on EM CLI scripting by Kellyn Pot'vin, Seth Miller, and Ray Smith as an example.

There are also books that are deep dives into a specific niche (my book with Sanjay Mishra years ago on SQL*Loader), and some authors have teaching styles (Doug Gault in *Beginning Oracle Application Express 4.2*) that help their readers learn more readily than from a manual set.

Despite the availability of free content, there are messages that cannot be presented well in short-format mediums such blog posts and question-and-answer sites. Books have a place in presenting long-form content.

Is it only the money that motivates authors? How many Oracle books would one need to write in order to make a comfortable living off the royalties?

Don't quit your day job. I actually tried doing that once. Probably I had too much success too fast, and I went from reducing my working hours to a leave of absence to resigning my position in order to write full time. I lasted a year, just about bankrupted myself, learned a ton in the process about cash flow and managing a business, and then went back to working a day job.

My own motivations are a tangled mix. There's a creative side to me that finds joy in creating value from thin air in the form of good content. Of course, not just thin air is needed but also hard work.

I like when my writing makes a difference. I like to help people, and I put a lot of work into my writing toward that end. I am more motivated by a nice reader email than by the money.

Money is nice too, but in today's book market one really needs other motivations. Sometimes the side benefits, such as increased stature in the industry, can lead indirectly to more money in one's job or consulting business. It is pretty rare in computer-book publishing to be able to make a living from it.

What impact did writing have on your own life?

First, I think of the people whom I've met through my writing. Relationships are some of the most important things we gain in life, and I value tremendously the people whom I've been able to meet as a result of creating good content.

Ken Jacobs is an example who comes to mind. What an amazing guy, and he was around from the beginning of relational database era. I met him without at first realizing who he was. I had numerous questions about the history of SQL*Plus, and Ken was kind enough to answer them. Ken was enthusiastic, saw the value in having authors write on Oracle technologies, and was always helpful—and I never tired of hearing his tidbits from the early days.

I'll mention self-esteem next. My first book deal came at a time when I had passed through a nadir in my career and had barely begun the climb back out. Couple that with self-esteem issues going back to a childhood with an alcoholic father and growing up in a family that could only ever afford rummage-sale clothing. Sprinkle in some work projects that had ended as failures, and I was just not in a good place.

Becoming published provided a success at a time when I needed one, and it was a nice boost to my career as well. I worked then for KPMG Consulting. My boss was more than supportive and gave me wide latitude to write on the side while working for him. I remain grateful to him to this day. Not every author is lucky enough to work for such an understanding and far-sighted employer.

It was a win-win in my boss's eyes. The research I did for books and articles taught me a lot, and I brought back a lot of knowledge about Oracle Database that was helpful to the business. My boss liked being able to send an author to his clients. The clients seemed to like that as well. What I learned while writing fed directly into my day job and allowed me to provide better service to our clients.

Of course, one's self-esteem should not be bound up in whether one can write a book or an article. Yet we all yearn to have some success and to make a difference in life, and now the challenge for me is to help others in doing some of the same things that made such a big difference to me back in the day.

Authors must find it disheartening to find their work pirated as soon as it is published. What are publishing companies doing to fight piracy?

Our parent company is Springer, and their policy is to send a takedown notice, or take other action as appropriate, for every instance of piracy that they find or is reported to them.

We recognize the frustration, and do what we can to fight piracy and ensure that authors reap their just rewards from the hard work they've put into creating their content.

Perhaps ironically, the market has spoken strongly in relation to copy protection. A sure way to reduce sales and cut an author's earnings from a book is to inconvenience and annoy readers by adding a layer of copy protection.

Piracy is a difficult problem. The way to avoid it is to not publish, and running away like that is not a tenable solution. Publishers do what they can to attack piracy and reduce it while recognizing that the world sometimes cannot be made perfect.

Which books sell best? I suspect that the answer is teenage vampire romances but let's stick with technical books. Are book sales correlated with market shares of the respective technologies, and does Oracle Database top the charts?

Books sell best when there is market churn and industry upheaval requiring readers to quickly learn new topics. The iPhone and the subsequent release of Android devices drove strong sales for years in topic areas relating to mobile device programming. There was a "gold rush" mentality—and yes, we use that very term—in the form of developers rushing to learn what it takes to get the next Angry Flying Things app into the App Store.

Books in database topic areas such as SQL Server and Oracle Database tend to be steady sellers over a long period of time, as contrasted to the boom-and-bust cycle seen in Android and iOS.

Market size sometimes has surprisingly little effect. Java book sales all but died for a while due to the long—almost five-year—drought between the release of Java 6 and Java 7. Oracle has turned that situation around by increasing the pace at which the language is revised.

It's difficult to know in advance how well a book will sell. A while back I published a book on what my boss and I both thought was a niche topic for a small number of Microsoft developers. My boss and I were stunned as the book went on to become our #17 best-selling book of that year. The topic may have been aimed at a small slice of the Microsoft audience, but it turned out to be something that small slice wanted a great deal.

How does a book go from being a germ of an idea to a finished product? I have a friend who is an expert in Linux device drivers and teaches a university course on the subject. He dreams of turning his course materials into a book. How should he go about finding a publisher? I'll forward your answer to him.

A good first step is to do just what your friend is doing: Find some colleagues who have been published. Ask their advice. Get introduced to some editors.

Think about the audience and the competition. Are there plenty of people wanting to learn to write device drivers? Is that audience already well served by multiple books? Does the publisher being approached have that topic covered, or would you be helping that publisher to fill a hole in their line?

Publishing the first book on a topic is good. Being second or third isn't so bad. It's tougher when you are, say, the fiftieth book on learning SQL. If you're the *N*th book on a topic, it helps to have an approach that sets you apart.

Then you must make a choice: Do you query an editor to ask about interest in a topic, or do you lead with a full-blown proposal? I lean toward the full-blown proposal. If you care enough about a topic to want to write on it, then put together a written picture of what the resulting book will be like. Share that written picture with an editor.

My book on SQL*Loader was rejected more than once when I verbally pitched the idea to my then-editor. It was when I put a written proposal in front of her that she grasped my approach and was able to share my vision, and then she accepted the book. Similarly, with my SQL*Plus book, my agent at the time worked me very hard and coached me in producing a good proposal that sold itself.

It's probably fine to approach an editor with a question like "What about a book on X?" Just don't give up if at first you get a "no" answer. If you have a strong vision for a book that you want to write, then commit that vision to paper and share it.

Issues and Controversies

You have a vantage point on all the goings-on in the database industry. Where do you come down on the big issues and controversies? For, example, are nulls really all that bad? Relational guru C. J. Date seems to think so. But the rest of us use nulls with carefree abandon.

I'm a pragmatic person who needs to earn a living and get home for dinner with the family. I had the honor of sharing a byline once with Lex de Haan on the topic of nulls. His and my focus in that article was not to argue for or against them, because there they are and we cannot change that. We directed our efforts in that piece toward helping readers to understand the "gotchas," and even more toward helping readers know how to sidestep and avoid some of the problems that can be encountered.

Should we go through PL/SQL for all database interactions? Bryn Llewellyn and Steven Feuerstein preach it. But most of us are quite attached to querying the database using good old SQL from the middle tier or client.

I won't be dogmatic on this issue. Offhand, I would argue that it depends upon the circumstances and the toolset in use, and also on the level of developer expertise. If one is using the Microsoft .NET Entity Framework to access an Oracle Database, then it may be best to simply let the framework access the database in the manner that it is designed to do. With other toolsets and languages, perhaps it is more feasible to have all access go through PL/SQL.

Can the .NET Entity Framework be made to mediate all database access through PL/SQL? That may depend upon the level of developer expertise that is available. It might depend upon whether there's enough budget to pay for additional expertise and developer time. Being able to say that all your access goes through PL/SQL is small comfort if one is late to market and misses a business opportunity.

SQL or NoSQL? Joe Celko has gone and legitimized NoSQL by writing a book on it.

NoSQL has always been legitimate. And what a marketing coup that term is, isn't it? Would NoSQL be half as popular today were it not for its rebellious-sounding name?

A part of me wishes I had some hands-on experience in using a NoSQL database for a real project. I am never fully comfortable unless I have hands-on perspective, and I just don't have it with NoSQL products, so I feel adrift in discussing the pros and cons of NoSQL versus relational databases.

Be pragmatic! That's my advice. Get some experience if you have the opportunity. Firsthand experience with both styles of database will give the best perspective.

Learning

Let's say that I'm a junior or intermediate Oracle DBA and my manager offered to buy books on Oracle Database and related topics for me. Which books would you recommend?

Buy books that you can understand, that you are ready for, that will make a difference by helping to improve your skills. The first book on Oracle Database that I ever purchased was way over my head. It was a book on performance, and it might as well have been a book on ancient Latin syntax. I lacked the foundation to comprehend the content and apply what I was reading to my daily work. I would have been better off in buying a book more suited to my then-current skill level.

Don't be afraid to take a chance on a book that catches your eye. Early in my career I almost walked away from buying C. J. Date's book *A Guide to the SQL Standard*. It was a quarter-inch-

thick book at a high price that made me balk. Yet it was the book that cast me headlong into working with databases. The appendix giving a critique of the language was gold, worth every penny all by itself. I look back at that small, thin book as being one of the hinge points affecting everything that I've done since.

You openly admit that you don't know everything and that you're always learning. Years ago, one of your readers had this nasty comment:* "I find it disturbing that someone billed as an 'editor specializing in database and programming titles' might lack that fundamental knowledge about what relational databases are all about." (http://www.oreillynet.com/pub/a/network/2002/10/01/whatsinacondition.html) *What do you have to say in your defense?

That was a hurtful comment. I hadn't thought about it in years, and it stung again just now as I reread that post to refresh my memory of the event.

I had written that post because I had learned something useful that I felt would benefit others in the field. Most of my writing today is done in that same spirit, to help others and share the joy of learning.

As I write this, I'm reminded of the difference we can make when we stay positive and constructive. Helpful advice can be delivered with tact and grace, and an encouraging word is never amiss.

That echoes what Ralph Waldo Emerson said in his essay on Self-Reliance:* "The other terror that scares us from self-trust is our consistency . . . A foolish consistency is the hobgoblin of little minds, adored by little statesmen and philosophers and divines. . . . Speak what you think now in hard words, and tomorrow speak what to-morrow thinks in hard words again, though it contradict every thing you said to-day." *You mentioned "mental models" in that blog post. Would you explain it to our readers? Perhaps give us an example.

A mental model is a set of beliefs that you hold about how a piece of software, or a software feature, works. Part of my mental model for SQL statements, for example, says that executing SQL statements results in disk I/O. Another part of that mental model says that trips across the network are "expensive" in terms of time, and that given a choice it's better to execute one SQL statement rather than two. Without these understandings, without my mental model of how SQL works, I'd likely write some very inefficient programs.

There's an illustration from one of my first experiences in becoming acquainted with Oracle. My employer sent me to a five-day introductory course on SQL, PL/SQL, and SQL*Plus. Good as the course was, it failed to leave me with the correct mental model of what these three things were and how they played together. Figure 1 shows the model in my head at the end of that course.

Figure 1. My incorrect mental model.

If you have any experience at all with SQL, PL/SQL, and SQL*Plus, you'll immediately see how wrong Figure 1 is. But wrong as it is, that was my mental model of these three features after taking that first Oracle course. As a result, I went back to work and attempted to write PL/SQL code such as the following:

```
DECLARE
   user_response VARCHAR2(1);

BEGIN
   ACCEPT user_response CHAR PROMPT '1=Detail Report, 2=Summary Report: '
   IF user_response = '1' THEN
   ...
END;
```

And when the above didn't work, I tried the following from SQL*Plus:

```
ACCEPT user_response CHAR PROMPT '1=Detail Report, 2=Summary Report: '
IF user_response = '1' THEN
   SELECT for detail report
ELSE
   SELECT for summary report
END IF;
```

But this second approach didn't work either! The first piece of code above won't work because ACCEPT is a SQL*Plus command, not a PL/SQL statement. The second piece of code won't work because IF is a PL/SQL statement, and not a SQL*Plus command. I know that now, but at the time I was frustrated, angry, and not meeting my deadlines.

It's probably different for each person, but what helps me the most is to dig into the details of how a thing works. Having someone explain that SQL and PL/SQL are sent to the database engine for execution is nice, but it's much more impactful to run an Oracle Net trace and see those things—SQL and PL/SQL—as they appear in the packets that are being sent.

Any last words before we go to print?

Rich Niemiec once signed a copy of his book for me with the words "Make an impact." I believe those were the precise words. I've never forgotten the sentiment.

Similarly, I remember many years ago asking a fellow high school student why he wanted to be a doctor. He responded with "I like to help people." I've never forgotten those words either.

Help people. Make an impact. Can we go wrong in pursuing either of those as goals? Sometimes I wish I had done more along these lines. Never do I wish I had done less. ▲

If you are an aspiring author with an awesome book idea, you can contact Jonathan at **JonathanGennick@Apress.com**. *You can also reach him through his website at* **http://www.gennick.com**.

Detailing Oracle CPU Consumption

by Craig Shallahamer

Craig Shallahamer

By now, most Oracle DBAs evaluate Oracle performance based on time—that is, a time-based analysis or, simply, TBA. The beauty of a time-based analysis is that it helps us understand what a user is experiencing. An Oracle time-based analysis includes both Oracle CPU consumption and Oracle wait time. But usually it excludes the time between the Oracle client process and the Oracle server process, because capturing the time between the Oracle client and server process is not simple and requires more than simply looking at Oracle trace files or v$ views.

Gathering CPU consumption is very straightforward via trace files and the Oracle Database v$ views. From the v$ views, CPU consumption can be gathered for a session from v$sess_time_model and for an entire instance from v$sys_time_model. The glaring hole in Oracle time-based analysis is that Oracle does not provide detailed CPU consumption numbers. For an Oracle process, the most detailed information we can obtain from Oracle is the total consumption.

If your boss asks you, "What is Oracle doing with all that CPU?" the best you can do is provide the total amount and then make inferences based on the Oracle wait events, the operating system situation, the application code, and the user experience. That's a lot of analysis information and obviously allows for a very detailed and spot-on analysis.

But still, in my opinion, it should be much easier to answer the "What is Oracle doing with all that CPU?" question, and the answer should be clearer. And compared to the wait interface detail, available CPU consumption detail is simply incomplete. I think as DBAs we can do better.

What I want is the same level of CPU detail as wait event detail. Or, at least, something so I can reduce the amount of inferencing required when answering the question, "What is Oracle doing with all that CPU?" It would be nice to gather Oracle CPU consumption for any Oracle process grouped by Oracle kernel function. Once I know the Oracle function, then I can find out, with some good detail, what the process is doing while consuming the CPU—and answer "the question."

With all the OS tools out there, I knew there must be one that would meet my requirements. There were four tools that caught my attention: strace/truss, DTrace, GDB, and Perf. Here is what I learned in my quest.

System Call Tracing Using Strace/Truss

I started with strace because I am very familiar with it. For many years I have traced Oracle processes at the operating system level using strace. (If you are on Solaris, the command is truss.) It's a fantastic way to learn how Oracle really works. And it's a great teaching tool (which I'm always looking for).

Here is an example of what you can do with strace. Suppose I wanted to know how long it takes an Oracle server process to read a single block from disk. Through Oracle wait interface we would see this as a "db file sequential read." To do an operating system trace on the Oracle server process 2518, I could issue this command:

```
$ strace -rp 2518
...
     0.000324 clock_gettime(CLOCK_MONOTONIC, {504, 52586559}) = 0
     0.000040 clock_gettime(CLOCK_MONOTONIC, {504, 52625324}) = 0
     0.000040 pread(257, "\6\242\0\f\0"..., 8192, 427270144) = 8192
     0.000047 clock_gettime(CLOCK_MONOTONIC, {504, 52712996}) = 0
     0.000044 clock_gettime(CLOCK_MONOTONIC, {504, 52757393}) = 0
     0.000329 clock_gettime(CLOCK_MONOTONIC, {504, 53086771}) = 0
     0.000040 clock_gettime(CLOCK_MONOTONIC, {504, 53125505}) = 0
     0.000040 pread(257, "\6\76 [y\f\0"..., 8192, 427278336) = 8192
     0.000047 clock_gettime(CLOCK_MONOTONIC, {504, 53213583}) = 0
     0.000040 clock_gettime(CLOCK_MONOTONIC, {504, 53253021}) = 0
     0.000327 clock_gettime(CLOCK_MONOTONIC, {504, 53580561}) = 0
     0.000040 clock_gettime(CLOCK_MONOTONIC, {504, 53619199}) = 0
     0.000040 pread(257, "\6\273\f\0"..., 8192, 427286528) = 8192
     0.000047 clock_gettime(CLOCK_MONOTONIC, {504, 53706779}) = 0
     0.000040 clock_gettime(CLOCK_MONOTONIC, {504, 53752611}) = 0
```

The beauty of strace is that all the system calls are shown in the OS manual pages. For example, if I want to learn about the clock_gettime call, I simply do a "man clock_gettime." I actually did that a few months ago when exploring Oracle Database 12*c*. My past research has been on 11*g* and when Oracle asks the OS for the time, I only observed Oracle making the gettimeofday call. What happened to the gettimeofday call? James Morle has a very nice posting at **http://www.scaleabilities.co.uk/2012/12/18/who-stole-gettimeofday-from-oracle-straces**.

What did I learn from the above? I observed that Oracle is instrumenting a single block read using the clock_gettime call, the server process is requesting a single 8K block using the pread call, and the first pread shown above took about 0.047 ms (not 0.040 ms).

While that's great, notice that there is no reference to CPU consumption and no mention of Oracle kernel functions either.

In fact, only the system calls are shown. Also, based on what the contents of the Linux /proc/sys/kernel/vsyscall64 file is set to, I may not see all the system calls.

So clearly, strace is not going to work.

Detailed Tracing Using DTrace

If you are a developer on Solaris, you probably have used DTrace. DTrace is a framework for troubleshooting processes as they are running . . . and in production. DTrace can be used to capture CPU consumption, I/O read times, I/O write times, network activity, and many other things. It is amazing!

DTrace programs resemble awk programs in structure: they consist of a list of one or more probes (instrumentation points), and each probe is associated with an action. These probes are comparable to a pointcut in aspect-oriented programming. Whenever the condition for the probe is met, the associated action is executed (the probe "fires"). A typical probe might fire when a certain file is opened, a process is started, or a certain line of code is executed. The probe may result in some information being displayed.

The good news is that DTrace is very powerful. I think any DBA could see the value in learning how Oracle works by using DTrace. Have you ever wondered how Tanel Poder finds out so much about the Oracle kernel? Tanel is a super big fan of DTrace, and you quickly see this when reading his work.

There is a downside though: there is a considerable learning curve. True, this rarely stops performance researchers, but it sure can slow them down. While DTrace is standard for Solaris, it's a little trickier on Linux. The DTrace package and the Unbreakable Enterprise Kernel (UEK) package are available on the Unbreakable Linux Network (ULN) but not on the public yum server. You must register your system with ULN before you can download the required packages. The Oracle Linux Administrator's Solutions Guide for Release 6 lists the packages you need to download from ULN and shows how to install them. Bummer.

Hmm . . . so DTrace is not installed and not working by default. That's a problem for me, especially in this quest. Whenever I do my research, I want others to be able to duplicate it. I also usually don't want someone to have to install, configure, and learn about another tool in the process. This, combined with the learning curve, put the kibosh on DTrace for this endeavor.

Using the GNU Debugger: GDB

The purpose of a debugger such as GDB is to allow you to see what is going on "inside" another program while it executes—or what another program was doing at the moment it crashed.

The word "debugger" is a clue that GDB is not going to meet my objective. GDB is great for stepping through Oracle's kernel code (and even interrupting or stopping it), but it wasn't clear how I could capture an Oracle process's CPU consumption by Oracle kernel function.

As with strace and DTrace, GDB isn't what I'm looking for.

Using the Linux Profiler: Perf

First of all, Perf is a free Linux process-profiling tool. When I checked on my pretty-much-default OEL 6.2 and Oracle Database 12*c* box, the Perf tools were already there!

On your Linux box, just do:

```
$ which perf
/usr/sbin/perf
```

In many ways Perf is like Oracle's ASH facility in that it samples data. For example, we can ask Perf to sample a process for five seconds at the default rate of 1K/sec and to keep a count whenever it sees the process "on CPU." Now suppose over the 5-second interval, the process was on CPU for 3K samples. Therefore, we know that 3/5ths of the 5-second interval—that is, 60% of the time—it was on CPU.

Oh yeah . . . one other thing: Perf keeps track of the count at the function level. This means I can determine how much CPU time (more correctly, the "counts") each Oracle kernel function consumes. Therefore, it appears Perf will do exactly what I need: keep track of an Oracle process's CPU consumption down to the Oracle kernel function. Perf-ect (sorry about that)!

To help make this crystal clear, now I can get something like this:

```
Count    Kernel Function
-------- --------------
  1500   abc124
   900   td8tr
   200   kcbte
    12   kbce8
------
  2612 counts over a 5 second interval
```

There is a lot more to Perf than I have mentioned. There is actually a suite of Perf tools, all of which begin with the word "Perf." For example:

- **perf stat** is used to obtain, that is collect, the event counts.
- **perf record** is used to record the event counts for later reporting.
- **perf report** is used to break down and report on the events process, function, etc.
- **perf top** is like the standard "top" command but provides function level detail for a specific process.

Here is a quick example of "perf stat" collecting the default events for process 28497 for five seconds. Notice the "sleep 5"? The execution duration of the command "sleep 5" is the collection period duration. I could have used another command, like "ls -1," but "sleep 5" is an obvious way to sleep for five seconds.

```
$ ps -eaf | grep oracleprod35
oracle   28497 28496  8 09:58 ?        00:31:36 oracleprod35 (DESCRIPTION=(LOCAL=YES
oracle   42265 28425  0 15:51 pts/1    00:00:00 grep oracleprod35

$ perf stat -p 28497 sleep 5

 Performance counter stats for process id '28497':

      560.686866 task-clock                #    0.112 CPUs utilized
             32 context-switches          #    0.000 M/sec
              1 CPU-migrations            #    0.000 M/sec
              0 page-faults               #    0.000 M/sec
   1,522,513,670 cycles                    #    2.715 GHz                     [83.11%]
     952,309,947 stalled-cycles-frontend  #   62.55% frontend cycles idle    [83.65%]
     568,827,423 stalled-cycles-backend   #   37.36% backend  cycles idle    [66.93%]
   1,333,413,297 instructions              #    0.88  insns per cycle
                                           #    0.71  stalled cycles per insn[83.54%]
     249,397,610 branches                  #  444.807 M/sec                   [82.99%]
       4,478,117 branch-misses             #    1.80% of all branches         [83.40%]

     5.000696361 seconds time elapsed
```

Here is an example of "perf top" for process id 28497 at a 30-second refresh rate.

```
$ perf top -p 28497 -d 30

PerfTop: 89 irqs/sec  kernel: 5.6%  exact: 0.0% [1000Hz cycles],(target_pid: 28497)
-------------------------------------------------------------------------------

             samples  pcnt function                 DSO
             _______ _____ ________________________ ____________________________

              205.00  7.6% kcbchg1_main             product/12.1.0/bin/oracle
              148.00  5.5% kcbget                   product/12.1.0/bin/oracle
              124.00  4.6% kcrfw_redo_gen_ext       product/12.1.0/bin/oracle
              114.00  4.2% kdkcmp1                  product/12.1.0/bin/oracle
              100.00  3.7% __intel_ssse3_rep_memcpy product/12.1.0/bin/oracle
               99.00  3.7% kcrfw_copy_cv            product/12.1.0/bin/oracle
               89.00  3.3% kcoapl                   product/12.1.0/bin/oracle
               84.00  3.1% kcbgcur                  product/12.1.0/bin/oracle
               77.00  2.9% kdiins1                  product/12.1.0/bin/oracle
               62.00  2.3% kduovw                   product/12.1.0/bin/oracle
               56.00  2.1% ktuchg2                  product/12.1.0/bin/oracle
               52.00  1.9% ktugur                   product/12.1.0/bin/oracle
               51.00  1.9% kdimod0                  product/12.1.0/bin/oracle
               50.00  1.9% updrow                   product/12.1.0/bin/oracle
               46.00  1.7% kauupd                   product/12.1.0/bin/oracle
               42.00  1.6% qertbFetch               product/12.1.0/bin/oracle
               38.00  1.4% kdudcp                   product/12.1.0/bin/oracle
```

Here is where it really gets interesting. I'm going to record just CPU cycles for process 28497 for 30 seconds. It's simple:

```
$ perf record -e cycles -p 28497 sleep 30

[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.093 MB perf.data (~4057 samples) ]
```

At this point the data has been stored in the default perf.data file. Now I'm going to create the default report based on the perf.data file, like this:

```
$ perf report

# Events: 2K cycles
#
# Overhead          Command      Shared Object                      Symbol
# ........  ...............  .................  ............................
#
     7.91%  oracle_28497_pr  oracle             [.] kcbchg1_main
     4.87%  oracle_28497_pr  oracle             [.] kcbget
     4.07%  oracle_28497_pr  oracle             [.] kcrfw_redo_gen_ext
     3.86%  oracle_28497_pr  oracle             [.] kdkcmp1
     3.42%  oracle_28497_pr  oracle             [.] __intel_ssse3_rep_memcpy
     3.37%  oracle_28497_pr  oracle             [.] kcoapl
     3.23%  oracle_28497_pr  oracle             [.] kcrfw_copy_cv
     3.08%  oracle_28497_pr  oracle             [.] kcbgcur
     2.56%  oracle_28497_pr  oracle             [.] kdiins1
     2.22%  oracle_28497_pr  oracle             [.] ktuchg2
     2.03%  oracle_28497_pr  oracle             [.] ktbgwl
     2.00%  oracle_28497_pr  oracle             [.] kauupd
     1.88%  oracle_28497_pr  oracle             [.] kduovw
     1.78%  oracle_28497_pr  oracle             [.] kdimod0
     1.56%  oracle_28497_pr  oracle             [.] kdudcp
```

Oh yeah . . . I forgot to mention that the "perf report" command is capable of outputting the results in comma-delimited format. Can anyone say "external table"? I think you see where I'm going with this. However, we need to get past a potential deal killer first.

Can We Go from Count to Time?

This is actually a very serious issue. Most DBAs will and should pause and consider the situation when jumping from sample counts to time.

Here is an example situation: Suppose over a 30-second interval the Oracle view v$sess_time_model (stat: db cpu) shows that a server process consumed ten seconds of CPU. Also, according to Perf, the Oracle kernel function kcb1 accounts for 60% of the CPU counts and kcb2 accounts for the remaining 40%.

Now the question is, can we attribute 60% of the ten seconds of CPU time to the Oracle kernel function kcb1 and 40% of the ten seconds of CPU to kcb2? If so, then kcb1 would be assigned six seconds and kcb2 four seconds.

Using the count-to-seconds math above combined with Perf's default sample rate of 1000/sec, (i.e., 1/ms), then each CPU count is equivalent to 1 ms of CPU consumption. I suspect it is likely either less than 1 ms or greater than 1 ms. Assuming that either direction is just as likely to occur (and I can't prove this to you), then I suspect assessing one count to 1 ms of CPU is OK.

Based on the above, I am comfortable jumping from Perf counts to time.

So Where Are We in This Quest?

At this point, we are ready to take the Linux Perf tool output and merge it with Oracle wait event data. The output could look something like this:

```
Time Component                                          secs       %
------------------------------------------------------ ---------- -------
cpu : [.] kcbgtcr                                      29.714   66.87
cpu : [.] kdstf000010100001km                           3.716    8.36
cpu : [.] lnxsum                                        3.541    7.97
cpu : [?] sum of funcs consuming less than 2% of CPU ti 2.393    5.38
cpu : [.] kaf4reasrpOkm                                 2.180    4.91
wait: latch: cache buffers chains                       2.100    4.73
```

I suspect you're not surprised that the above output was, in fact, taken from an existing tool. The tool is called "fulltime.sh" and is the result of a collaborative effort between Frits Hoogland and me. You can download the tool from **http://www.orapub.com**. Just do a search for "fulltime," and you'll see it.

How the Tool Works

Obviously the tool collects both process CPU consumption and Oracle wait event data. But we wanted some usage flexibility, so it's a little more interesting than that. Also, the real trick is outputting the Perf data into a comma-delimited file using "perf report -t,". It's the "t" option that is key here. Oracle can easily read the comma-delimited file as an external table and then combine that with the wait event data collected from the v$sess_time_model view.

Here is the basic idea:

```
Help user identify the PID to profile
Initial setup
Loop
  get oracle wait times (snap 0)
  get oracle CPU time (snap 0)
  start oracle kernel cpu details collection
  sleep x
  get oracle wait times (snap 1)
  get oracle CPU time (snap 1)
  stop oracle kernel cpu collection
  do some cool math and combine results
  display results
End Loop
```

The looping capability easily allows for a single collection or multiple collections. The multiple collection option displays much like the "top" tool with a default cycle duration of three seconds.

If you're running on a virtual machine, make sure to change the shell script variable PERF_SAMPLE_METHOD to "-e cpu-clock" instead of the default "-e cycles." If you look near the top of the fulltime.sh script, you will see a comment about this and also where to make the change.

Before you run the script, take a quick look at it, especially the top third. There is some good information about the script, default settings, and usage details.

How to Run the Script

There are two input usages. Let's call them "basic" and "advanced."

The BASIC USAGE helps you find the Oracle process ID, sets some defaults for you, and works on a three-second cycle until you control-C. For example, let's say you don't know the OS process but you do know the Oracle SID, or perhaps you know the machine name or user name. Then you're in luck because the fulltime.sh script will display this information and then prompt you for the OS PID. Here is an example of what you can expect if you simply enter "./fulltime.sh" and selected process 60505.

```
Welcome to the FULLTIME script (v3e)
To see wait time details AND Oracle kernel function CPU details TOGETHER
Use at your own and your organization's risk!

If unable to execute perf, do as root:
   echo 0 > /proc/sys/kernel/perf_event_paranoid

The perf sample method is set to: -e cycles
Use cpu-clock for virtualised hosts, cycles for physical hosts

PID    SID    SERIAL MACHINE  USERNAME  SERVER     OSUSER  PROGRAM
---    ---    -----  -------  --------  ------     ------  -------
16333  326    32577  sixcore  SYSTEM    DEDICATED  oracle  sqlplus@sixcore (TNS V1-V3)
16334  10     61755  sixcore  SYSTEM    DEDICATED  oracle  sqlplus@sixcore (TNS V1-V3)
16335  249    537    sixcore  SYSTEM    DEDICATED  oracle  sqlplus@sixcore (TNS V1-V3)
16336  170    2459   sixcore  SYSTEM    DEDICATED  oracle  sqlplus@sixcore (TNS V1-V3)
16337  406    32615  sixcore  SYSTEM    DEDICATED  oracle  sqlplus@sixcore (TNS V1-V3)

Enter PID to profile : 16333
```

For each cycle, you would see something like this:

```
Fulltime.sh v3e

PID: 16333  SID: 326  SERIAL: 32577  USERNAME: SYSTEM  at 16-Apr-2014 14:07:11
CURRENT SQL: SELECT COUNT(*) FROM DBA_OBJECTS WHERE 1=0

total time: 2.74 secs,  CPU: 2.492 secs (90.94%),  wait: .248 secs (9.06%)
                                                                  Time
Time Component                                                    secs       %
----------------------------------------------------------------- ---------- -------
cpu : [?] sum of funcs consuming less than 2% of CPU time          2.139   78.07
wait: cursor: pin S                                                0.248    9.06
cpu : [.] __intel_new_memset                                       0.203    7.39
cpu : [.] opiexe                                                   0.088    3.21
cpu : [.] audsel                                                   0.065    2.36

To see the Call Graph, press ENTER or to exit press CNTRL-C.
Samples remaining: 985
Gathering next 5 second sample...
```

Notice that both Oracle and OS process details are shown in conjunction with the date and time. Plus, you'll see the SQL statement that the process is running at the end of the sample period. Then the total time is shown along with the two high-level components: CPU consumption and Oracle wait time. Finally, the details are shown. Notice that the details clearly identify the time component as either CPU or wait time.

If the time is tagged "CPU," then the Oracle kernel function is displayed (thank you, Perf) along with its inferred time (based on both Perf and v$sess_time_model). If the time is tagged "Oracle wait time," then the wait event name is displayed along with the wait time (based on v$session_event).

It's important to understand that if we ran a standard wait event report, the output would be similar but with NO CPU information. The wait events detail would still be displayed. This tool simply adds value by incorporating the CPU consumption details.

The ADVANCED USAGE gives us full control. It requires the OS process ID, the cycle duration, and the number of cycles. The advanced option makes it simple to get a single long cycle or multiple shorter cycles. For example, if I wanted to watch process 1234 in 5-second intervals, I could do this:

```
$ ./fulltime.sh 1234 5 9999
```

Or suppose I wanted a single 60-second sample for process 5432. I would enter this:

```
$ ./fulltime.sh 5432 60 1
```

The output will look exactly like the output from basic usage. You are just not prompted or given information to help you pick the OS PID.

Why This Is So Important

Look at the example screenshot below and ask yourself, "Is there an I/O bottleneck?"

```
Fulltime.sh v3e

PID: 19367  SID: 37  SERIAL: 9  USERNAME: MG2  at 16-Apr-2014 14:14:44
CURRENT SQL: SELECT SUM(OBJECT_ID) FROM ALL_OBJECTS

total time: 3.61 secs,  CPU: 3.478 secs (96.35%),  wait: .132 secs (3.65%)

Time Component                                                    secs       %
----------------------------------------------------------------- ---------- -------
cpu : [?] sum of funcs consuming less than 2% of CPU time          2.516   69.69
cpu : [.] kcbgtcr                                                  0.223    6.17
cpu : [.] _intel_fast_memcmp                                       0.211    5.85
cpu : [.] expepr                                                   0.107    2.96
cpu : [.] qerixGetKey                                              0.105    2.91
cpu : [.] qerfxFetch                                               0.103    2.84
cpu : [.] kdxlrs2                                                  0.102    2.81
cpu : [.] ktrgcm                                                   0.094    2.61
wait: db file sequential read                                      0.066    1.83
wait: events in waitclass Other                                    0.053    1.46
wait: latch: cache buffers chains                                  0.008     .22
wait: read by other session                                        0.004     .11
wait: latch: row cache objects                                     0.002     .04
wait: library cache: mutex X                                       0.000     .00

To see the Call Graph, press ENTER or to exit press CNTRL-C.
Samples remaining: 992
Gathering next 5 second sample...
```

If the CPU information was not combined with the wait time details, I suspect 90% of Oracle DBAs would say there is an I/O problem. They would see the top wait event db file sequential read and then conclude that there is an I/O issue. However, as the fulltime.sh script clearly shows, CPU consumption is much larger than Oracle wait time, providing us with a greater opportunity to reduce the total time and devise additional spot-on solutions. Also, the report's total time breakdown clearly shows CPU consumption is 97% of the total time. I wouldn't be surprised if there was a raging CPU bottleneck! ▲

*Craig is an Oracle ACE Director, a performance researcher and blogger, a consultant, an author of two books (*Oracle Performance Firefighting *and* Forecasting Oracle Performance*), a regular conference speaker, and a passionate teacher. He is the founder of OraPub and the creator of the performance analysis tool Stori. His blog, "A Wider View," is where to find his most recent performance research. Contact him at craig@orapub.com, @CShallahamer, LinkedIn, and Xing.*

Simplicity Is *Good*!

by James Morle

James Morle

This article is about the importance of *appropriately simplistic* architectures. I frequently get involved with the creation of full-stack architectures—in particular the architecture of the database platform. There are some golden rules when designing such systems, but one of the most important ones is to keep the design as simple as possible. This isn't a performance enhancement; this is an *availability* enhancement. Complexity, after all, is the enemy of availability.

Despite it being a sensible goal, it is incredibly common to come up against quite stubborn resistance to simplicity. Frequently, the objections will be based upon the principles of the complex solution being a "better way" to do things. I have two closely linked examples of this in action.

Case 1: Real Application Cluster Interconnects

A cluster interconnect is an incredibly important component of the architecture. The cluster exists, after all, as an availability feature (and possibly a scalability feature), and so the foundations of the cluster must be robust in order for it to deliver that availability. The cluster interconnect is the lifeblood of the cluster. And yet, it has such a very simple set of requirements:

- Point-to-point communication between all nodes of the cluster
- Low latency
- n+1 availability of network paths
- Multicast support between the nodes
- (optionally) Jumbo frames support

It explicitly does not need any of the more "fancy" networking features, such as:

- Routing of any kind
- Spanning tree support
- VLANs
- *Access to any other networks*

It just needs a dedicated pair (or more) of discrete layer-2 networks. They don't need to be bonded; the networks do not even need to be aware of each other—they are completely independent (at least, that is certainly the case since the HAIP functionality of Oracle 11*g* Release 2). They *do* need real switches, though—crossover cables fail ungracefully in the event of a peer host losing power. But they don't need anything high end—just something better than crossover cables and with enough bandwidth for the required traffic rates. The latency difference between the majority of switches is barely a consideration. The switches don't really even need redundant power supplies, though it's not a terrible idea to insulate yourself from this type of failure, and it brings no detriment apart from a marginal cost increase.

So, something like a pair of *unmanaged* layer-2 GbE Ethernet switches are the perfect solution. Something like a Netgear JGS516 would probably do the job, from a brief scan of the specification. They are about $166 (£100) each, net cost of $332 (£200) for a nice, robust solution. Or if you wanted to really push the boat out, something like a fully managed L2 switch with redundant power such as the HP E2810-24G will set you back all of $1160 (£700) each. Cisco shops might spend a bit more and go for something like a 3750 G for about $4650 (£2800) each.

But . . . somebody will always push back on this. They will plumb the cluster nodes into the full core/edge corporate dream stack topology with fully active failover between a pair of core switches. Surely, at a cost of more than four orders of magnitude more than the bargain basement Netgear solution, this must be better, right? Wrong.

There are numerous aspects that are incorrect in this assumption:

a. Higher cost means better.
b. There will be an increase in availability.
c. Every networking requirement is the same as every other one.

First of all, these network topologies are not designed for cluster interconnects. They are designed for corporate networks, connecting thousands of ports into a flexible and secure network. RAC interconnects are tiny closed networks and need none of that functionality. More precisely, they need none of that *complexity*. Corporate networks also have a different level of failure sensitivity to cluster interconnects; if a user's PC goes offline for a couple of minutes, or even half an hour, the recovery from that failure is instant once the fault is rectified—the user is immediately back in action. Cluster interconnects are not so forgiving; if a cluster's networks go AWOL for a few minutes, the best you can hope for is a single node of the cluster still standing when the fault is rectified. That is how clusters are designed to operate: if the network disappears, the cluster must assume it is unsafe to allow multiple nodes to access the shared storage. The net result of this failure behavior is that a relatively short network outage can result in a potentially lengthy full (and manual) restart of the whole cluster, restart of the application, balancing of services, warming of caches, and so on. It would not be an exaggeration for this to be a one-hour or greater outage. Not terrific for a highly available cluster.

But hang on a minute—this über-expensive networking technology never goes down, right? Not true. What exactly is this active/active core switch topology? Think about it. It's a kind of cluster itself, with each switch running complex software to determine the health of its peer and managing a ton of state information between them. The magic word in that sentence was the word "software"— anything that is running software has a great deal of failure potential. Not only that, but clustered software has a great deal of potential to fail on *all nodes concurrently*. This is a unique attribute of distributed software and one that does not exist in discrete hardware designs. In discrete hardware designs it is incredibly unlikely that more than one component will fail concurrently. Software is great at catastrophic failure, most particularly when it is combined with some element of human error during upgrades, reconfiguration, or just plain tinkering. Not even humans can make two independent hardware switches fail concurrently, unless they are being creative with power supply.

Just to highlight this point, I should state here that I have personally witnessed failures of entire core/edge switch topologies on three occasions in the last five years. It does not matter that the cluster nodes are connected to the same edge switches when this kind of failure occurs, because every component in the network is a logical contributor to the larger entity and will become unavailable as part of a larger meltdown. If you are a Blackberry user, you have experienced one yourself recently. The Blackberry issue proves the potential, but in their case the topology was at least appropriate—they have a requirement to interconnect thousands of devices. In our clusters, we have *no such requirement*, and we should not be implementing overly complex and thus unreliable network topologies accordingly.

Case 2: The Great SAN Splurge

Now let's think about Storage Area Networking. And let's not restrict this thought to Fibre Channel, because the same principles apply to an Ethernet-based SAN. In fact, let me just clear off the Ethernet SAN piece first: Don't use your corporate network for storage connectivity. It's the wrong thing to do for all the reasons stated in the first case on this page.

So, now we can focus on Fibre Channel SANs. Fibre Channel has become the backbone of the data center, allowing storage devices to be located in sensible locations, perhaps in different rooms to the servers, and for everything to be able to be connected to everything else with optimized structured cabling. The zoning of the fabric then determines which devices are allowed to see other devices. All very well and good, but how is this implemented? Unsurprisingly, it is implemented using an exactly analogous solution to the core/edge Ethernet network design in the previous case. Two active core switches lie at the heart of a multi-tier network and provide failover capability for each other. A cluster. This cluster can (and does) fail for exactly the same reasons given in the former case, and yes, I have also seen this occur in real life—twice in the last five years.

The failure implications for a SAN meltdown can be even more serious than a cluster meltdown. *All* I/O will stop and, if the outage goes on long enough, *all databases in the data center* will crash and need to be restarted.

There are a few other implications with this topology in large data centers. Notably, it is common for the storage arrays to be connected via different physical switches than the servers, implying that there are a number of Inter-Switch Links (ISLs) to go through. These ISLs can become congested and cause severe bottlenecks in throughput that can be extremely tricky to track down. In extreme cases, ISLs can be the cause of *multi-minute* I/O response times, which will also cause clusters and databases to crash.

So that preamble paints the SAN picture and sets the stage for the following questions:

Why are all devices in the SAN connected to all other devices? Why are the handful of nodes that make up your critical database part of a SAN of thousands of other devices? Why are they not just connected via simple switches to the storage array?

There is only one reason and that is data-center cabling. But it doesn't really follow: If your database servers are in a rack, or a few racks next to each other, put a pair of physically and logically discrete switches into the top of the rack, attach all the nodes, and then connect the storage array using the same number ports that you would have connected to the switches if they had been edge switches. The destination of those cables would be the storage array rather than the core switches, but the number of cable runs is pretty much the same and results in a more robust solution. There is no exposure to catastrophic loss of service in the SAN, because there are two completely discrete SANs between the servers and the storage.

Fibre Channel networks are *vertical* in nature: server nodes do not communicate with other server nodes over the SAN; they only communicate with the storage array. Server nodes do not need to be connected to thousands of storage arrays, either. The connectivity requirement for a given platform is actually rather simple.

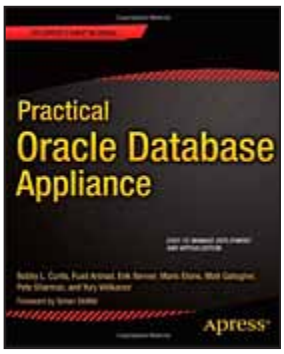
Note: I am writing from the viewpoint of a typical RDBMS implementation, not from the viewpoint of massively parallel HPC or big data systems. Clearly, if there truly are thousands of devices that *do* need to be connected, this argument does not apply.

Conclusion

The common theme between these two cases is this: Don't connect things that don't need to be connected. Yes, it is easier to cable up, and arguably easier to manage, but it has a knock-on effect of dictating an implementation that does not suit the requirement. It results in a less reliable, more complex solution, with the cart very much before the horse. Don't trade off administrative simplicity against architectural simplicity: it will sneak up and bite you.

As Albert Einstein said, *"Make things as simple as possible, but not simpler."* Wise words indeed. ▲

James Morle is a specialist large-scale Oracle-based systems consultant with over 20 years experience in IT consulting. His success is based on two key factors: deep domain expertise in the full Oracle stack (he speaks every language) together with huge credibility with senior IT and business leaders. James is one of the co-founders of the OakTable Network and an Oracle ACE Director. He is also the founder of Scale Abilities Ltd., based in the UK and serving clients across the world. Scale Abilities solves implementation, storage, and performance problems with large, complex Oracle-based systems. International companies rely on its extensive full-stack expertise, which successfully solves the wide range of cross-boundary problems that single-discipline consultancies and experts struggle to handle. Scale Abilities consultants will tell you what others are afraid to share, and they understand how to unpick politics from performance.



Business Values for the ODA

An excerpt from *Practical Oracle Database Appliance*

by Bobby Curtis, Fuad Arshad, Erik Benner, and Maris Elsins

This chapter is an excerpt from the book Practical Oracle Database Appliance *published by Apress, January 2014, ISBN 1430262656; copyright 2014. Reprinted with permission. For a complete table of contents, please visit the publisher site:* **http://www.apress.com/9781430262657**.

In September 2011, Oracle announced the availability of the Oracle Database Appliance as an innovative, entry-level engineered system optimized for databases. The announcement promised easy deployments, pay-as-you-grow licensing, and single-vendor support.

When a new technology is introduced, and business interest builds, someone within a company will ask the question: "Is this a better solution than the way we do things today?" The first step in answering this question and overcoming resistance to change is investigating the business case for the new technology. The Oracle Database Appliance has become a very successful product in the Oracle engineered systems product line. This chapter outlines the business benefits offered by ODAs, which you can use to consider the business case for deploying an ODA solution and assessing any potential technology risks.

Business Challenges

Many companies face the same challenges in meeting an ever-expanding range of business pressures. IT departments are being asked to become agile and deliver infrastructure quickly, even before all of the business requirements have been fully developed. Project timelines are becoming shorter and multiphased. While projects may run late during the early stages, this doesn't mean that the project due date will necessarily be changed. Instead, later project phases may need to make up the slack with shorter timelines. Database departments are usually at the end of the infrastructure provisioning chain. DBAs are being asked to deploy new database infrastructure rapidly and to deliver business value to IT customers sooner. If a project is late by the time it's time for DBAs to do their work, the lead times effectively drop to zero. The old practices of following a cycle of gathering requirements, ordering hardware, and engaging multiple teams with numerous handoffs between them to build database infrastructure simply don't meet modern business timelines.

Many companies face costs pressures from static budgets. New external public cloud offerings are competing for their infrastructure budgets. For a number of years, there has been a slow shift to commodity hardware solutions to reduce costs. While hardware related costs have fallen from commodity solutions, the costs of building and supporting database infrastructure solutions remain high.

Companies often face challenges from shortages of people with high-end skill sets. In general, there isn't a shortage of IT people, DBAs, and other staff. However, the mantra that "good people are hard to find in any economy" does persist. Companies are always looking for people that can solve any technical problem, get work done quickly, and build specialized subject matter expertise quickly.

The rapid introduction of new technologies has placed additional pressures on the availability of people with high-end skill sets. The rapid introduction of these new technologies can require the support staff to get up to speed without a lot of hand-holding. Support staffs have to react quickly to the challenges at hand, and task completion has to be quick. There is little margin for errors or problems that delay projects. Common issues include:

- People's plates are literally full today. There is no time for "OJT"—on-the-job training—and research. Everything has to be jump-started.
- Hourly consulting costs on projects can be very high, adding to the cost pressures. This adds to the pressures for rapid deployment and efficiency.

Security compliance is becoming vital in an age where audits and industry regulations are governing the actions that companies have to take to secure their systems. It is very difficult to achieve compliance across your application and database portfolio in an environment where database servers and databases are custom built.

Enter the ODA

Oracle developed the Oracle Database Appliance, or ODA, to develop a solution to the business challenges mentioned in the preceding section. If the ODAs have an overarching theme, it is "Infrastructure Deployment for Dummies." While some people may be taken back by this term, it is actually meant as a testament to the simplicity and elegance of ODA deployments. The other ODA themes include the time and ease of deployment using a one-button process, the ease of RAC deployments, and the ease of management. After viewing the ODA setup poster, you know that Oracle is delivering a new way of doing business.

The current X3-2 ODA model is an engineered system. An engineered system is hardware and software designed to operate together as a single, integrated packaged solution. While a com-

pany can build its own commodity hardware solution, very few have the ability to build their own engineered systems. Most companies simply don't have the resources, or business benefits, to build their own engineered systems, let alone a system that is optimized for highly available Oracle databases.

The X3-2 ODA comes prepackaged with two servers with a total of 32 –2.9Gz Intel E5-2690 CPU cores. An ODA comes with a storage cell that contains 20 900G data drives totaling 18T of raw storage, which can be expanded by adding a second storage expansion rack with an additional 18T of raw storage, for a total of 36T of raw storage. ODAs are deployed with either double or triple mirroring (configurable at installation time), bringing the usable storage to approximately 6T with triple mirroring, or 9T with double mirroring, on a single storage cell. The ODA Getting Started Guide should be consulted for the exact sizing of the mirrored storage options and disk groups, based on the deployment options selected. The current X3-2 servers come with 256G of memory each. Four SSD drives with a total of 800G of storage are included for the online redo logs.

Note: *ODA capacity is certain to change with every generation of the underlying Oracle server hardware product line. Almost as soon as a hardware model is released, Oracle starts development of the next generation replacement.*

The first generation of the ODA was a fixed, self-contained appliance unit. For the X3-2 second-generation model, the server and storage units were modularized for the product line to take advantage of an Oracle product roadmap for releasing new hardware to follow the Intel product line developments.

ODAs offer an engineered system from a single supplier, with a single-issue escalation point. The management of the appliance is mainly self-contained and is integrated with Oracle's comprehensive Oracle Enterprise Manager (OEM) management and monitoring solution.

ODAs have been purchased and deployed by a large customer base, consisting of companies of every size. A number of Fortune 500 companies have reported that the majority of their databases can be deployed on ODAs from a resource sizing standpoint. Models with more capacity and higher performance can be expected in the future.

Virtualization is also available on ODAs to support the rapid deployment of databases and applications using Oracle Virtual Manager (OVM) templates. A template for WebLogic has been released, and Oracle has announced the availability of additional templates.

ODAs will maintain their place within the Oracle engineered system product lineup for the foreseeable future. In August 2013, Oracle expanded its engineered system product lineup to include a new offering called the Oracle Virtualized Compute Appliance, or OVCA. This system offers a generalized Oracle Virtual Machine (OVM) provisioning solution, whereas ODAs offer a lower-cost entry point and the virtualization has been optimized for running databases.

Fast, Simple Database Infrastructure Deployments

The first business benefit offered by ODAs is fast and easy database infrastructure and database instance deployments. Deploying an ODA does not require spec'ing equipment, provisioning Storage Area Network (SAN) infrastructure, deploying RAC private interconnect networks, installing an Operating System (OS), or installing RAC, ASM, the grid infrastructure, Oracle database software, or project managers to coordinate all of this work.

The deployment of an ODA simply requires the steps outlined in the setup poster and deployment documents. Once the server is on the network, and the deployment configuration file is built, the time to build the ODA is slightly less than one hour. The result is a two-node RAC cluster, including the configuration of the file systems, ASM storage, grid and database homes, a local OEM database control, and a starter database. Your company can be running on a RAC cluster in less than a day.

If the time to deployment matters to your company, then ODAs will meet the need to have fully deployed high-availability databases and virtualized application solutions within days of completing the server cabling and other network dependencies. An ODA can be deployed by the average DBA by just reading the documentation. If your company is deploying a larger portfolio of ODAs, then adding additional skill sets to your team may make sense.

After deploying our first ODA, we counted the number of pre-deployment, install, and post-deployment steps involved in building databases on our other build-it-yourself platforms. The number of steps to cover all of the bases for the database server, without creating the databases, was between 103 and 107. The equivalent number of steps on an ODA to accomplish the same result, and also deploy a RAC cluster, numbered only in the twenties. Needless to say, we estimated that the reduction in the build and setup time was geometric in nature. Every company should perform their own count of the number of steps and handoffs between teams required to build a database server and high-availability databases using their current build processes.

We experienced a similar reduction in the costs for deploying ODA database infrastructure. Every company has their methods and rules for defining organizational roles and responsibilities. Let's just say that you can build an ODA without the involvement of a SAN team and sysadmins if that aligns with your company's views on roles and responsibilities. There is also a reduction in the amount networking work required to deploy and ODA.

A number of companies we have talked to regarding what resources are needed to build an ODA vs. how things are done today at their company, report that this change in paradigm can result in some interesting internal discussions as they shift their strategy toward engineered systems. The shift in roles and responsibilities can be worked out up front, or they can slowly change over time, as companies become more familiar with the platform and gain more experience.

So that leads to the question: "How long does it take to deploy an ODA?" To answer this question, you have to realize that an ODA has dependencies just like any other engineered system. You still have to order the ODA and have it shipped. An ODA requires fairly simple network cabling, and you still need to complete any DNS and firewall requests. However, much of this work can be done in parallel, even before the ODA arrives. With careful planning, an ODA can be deployed very quickly, with the actual build work taking less than a day.

The author's first experience with an ODA was for a project that required high availability and a very fast deployment. We needed to order equipment and have a set of four RAC clusters running in two weeks. At the time, the ODA platform was new, but we decided to take a chance because this was our best option. We ordered four ODAs and asked for immediate shipment. The

ODAs arrived on a Monday, and we racked and cabled them on Tuesday. Some Oracle people flew in on Wednesday morning to see if they could help out, since the platform was very new. We drove to the data center and built all four ODAs at the same time in parallel Wednesday afternoon—with Oracle looking over our shoulders.

The process went so well that we decided to take one of the ODAs and bare metal a brand-new running production ODA RAC cluster in front of the entire DBA department over lunchtime. We finished the deployment just as everyone finished their lunches. People knew that they were looking at a new paradigm in building database servers. Building four RAC clusters in parallel in an afternoon with just a couple of DBAs was unthinkable before that day. Table 9-1 lists many of the tasks that we no longer had to be concerned with as a result of deploying an ODA.

Task	Traditional	ODA
Spec equipment	Standards dependent.	Standardized hardware and software.
Cabling for external RAC network	On RAC systems.	Included in the hardware.
Provision SAN storage	In most cases.	Included in build.
IO virtualization	Standards dependent.	Included in build for for virtualized systems.
Server partitioning	Standards dependent.	Not needed for databases, but can be done to accommodate nondatabase virtual containers.
OS install	In most cases.	Included in build.
Grid Infrastructure install	In most cases.	Included in build.
Database software install	In most cases.	Included in build.
Project management to coordinate team handoffs	Company dependent.	Included in build.

Table 9-1. *Work That Doesn't Have to Be Done to Deploy an ODA*

So what happened next? Today it isn't unusual to be approached by application teams or project managers with the question, "Can we put this database on an ODA?" You're in a good place when you are getting this type of question.

The ODA install process eliminates unpredictable, time-wasting setup issues and handoffs that result in additional coordination, dependencies, and delays. Deployment times are measured in days, not weeks or months. The ODA provides a lot of flexibility in meeting project requirements challenges, which is something that a lot of database and infrastructure departments have to deal with.

ODAs can be pre-deployed quickly to avoid the need to buy capacity before it is needed. They are the perfect platform for the majority of your Oracle databases in support of Infrastructure as a Service (IaaS), Databases as a Service (DBaaS), or cloud-based services in support of your company's IT strategy. When you get the question: "How are you provisioning your databases so quickly and efficiently in your DBaaS offering?"—you are in an even better place. ODAs won't meet every project requirement. There are limits to the number of IOPS, and the compute nodes can't be scaled beyond two at this time. However, you may find that ODAs will meet the requirements for the majority of your databases.

RAC Without Tears

Building RAC clusters that will be stable and meet high-availability requirements requires a great deal of up-front planning and work, as well as pre- and post-install checks to validate the environments. Besides the RAC install manuals, there are additional documents that have to be analyzed:

- The RAC generic (MOS 810394.1) and any platform specific starter kits
- The RAC FAQ (MOS 220970.1)
- Oracle's support note (MOS 1344678.1) giving steps to stabilize a cluster, with platform-specific extensions
- The RAC Information Center note (MOS 1452965.2)
- Your platform-specific RAC setup recommendations
- RAC Known Issues support documents

In addition, you'll need to consult any documentation you have that gives advice on setting up RAC to support other Oracle products, such as eBusiness Suite and ATG Web Commerce.

There are additional tools you'll need to learn if you are building your own RAC clusters. Oracle has built various tools to help in validating RAC environments, such as:

- RACcheck (MOS 1268927.1)
- Cluster Verify (MOS 316817.1)
- RDA MOS 314422.1 and 250262.1 for Database checks

And, of course, there are all of the OS-specific settings, package requirements, and checks. Going through all of this effort on a traditional platform is well worth the effort. If you are fortunate, all of this will be documented one time and someone will make the effort to keep up with all of the documentation updates and RACcheck results to keep the "do it yourself" documentation updated. However, doing this takes a lot of time, the right people, a lot of handoffs between departments, and a lot of resulting QA work to make sure the work was done right.

Documenting and distributing complex technical information and training people to understand the information is a challenge for most companies, especially smaller and midsize companies. Adding to challenges are all of the handoffs between multiple teams that are required to implement all of the requirements. Someone may forget to route the interconnect traffic to redundant RAC-only switches (which are also expensive), or forget to configure Jumbo frames on both the server and switches. There can be additional issues related to external vendor software supporting IO multipathing. Because all of these scenarios are very real, Oracle has found the need to invest in extensive RAC diagnostic facilities.

Oracle is helping to manage this complexity by extending RACcheck and other tools to catch setup issues. However, the conclusion is that there are real reasons the people in the Oracle RAC assurance group and Oracle support are very busy these days. Engineered systems and ODAs take a lot of the work to correctly deploy RAC off your shoulders.

ODAs provide an answer to all of these challenges by making RAC configurations self-contained. There are no external interconnect requirements or instructions for making RAC configuration changes outside the standard ODA deployment process. If you are concerned about the ability of your company to successfully complete all of the engineering work required to build truly high availability RAC clusters, then ODAs may help provide a solution. The expertise to successfully deploy RAC clusters is greatly reduced. This is a major advantage for companies of all sizes. ODAs extend the ability to successfully deploy two-node

RAC clusters to the masses, which is part of the reason that ODAs are a solution for "RAC Without Tears."

RAC One, a two-node RAC cluster, or a single instance database can be deployed during an ODA install. ODAs deploy RAC clusters in a matter of hours and according to Oracle's best practices. Oracle builds the physical RAC interconnect into the appliance. The quarterly automated patching process simplifies the process for keeping the grid/clusterware and Oracle Homes patched with the latest PSU patches and bug fixes. These quarterly patches are tested as a complete unit, along with the OS and firmware changes, making ODAs a great solution for maintaining ongoing RAC stability.

Note: *The author has discussed RAC installations on non-ODA systems with a number of small companies. Many small and midsized companies simply don't have the resources to perform that sort of installation by themselves. This includes the technical resources to research all of the RAC requirements and the cost of buying additional equipment, such as dedicated RAC switches, for the interconnect traffic.*

While ODAs do a great job deploying and supporting RAC clusters, the need for understanding good RAC design best practices for eliminating contention between nodes is still something that teams deploying RAC on any platform need to be aware of. Also, a two-node cluster requires that one server is able to handle the entire processing load during a node switchover. Supplementing RAC with Data Guard can alleviate the failover capacity concerns.

High Availability Without Tears

ODAs provide high availability through easy-to-install RAC and RAC One deployments. In addition to provisioning high availability on a single appliance, ODAs make it easy to deploy Data Guard to extend the high availability solution. These deployments can cross data centers to provide geographic protection. Oracle assists the Data Guard deployment process through OEM, one-button ODA Data Guard deployment scripts, and supplemental documentation focused on deploying Data Guard on ODAs. In addition to streamlined Data Guard deployments, ODAs support high availability through higher levels of stability because it is a packaged solution tested as a single unit.

The grid and database portions of ODA patches can be deployed in a rolling fashion. At the time of writing, the infrastructure portion of ODA patches requires the clusterware to be completely down on both nodes, but this may change in the future. RAC One provides high availability capabilities for both production and nonproduction instances that are designed to run on only one of the two ODA nodes, but can be failed over to the second ODA node.

ODAs jump-start high availability implementations by greatly reducing the expertise levels required to deploy a RAC solution. This includes eliminating the need to perform a RAC interconnect design, researching all of the RAC requirements and best practices, provisioning ASM, eliminating all single points of failure, and all other aspects of a well-architected RAC cluster.

High availability is also achieved from the deployment of standard, tested configurations and the choice of either double or triple mirroring all ASM disk extents. The loss of a disk doesn't take the system down. The ODA operating system also implements the Oracle Linux unbreakable kernel (UEK). ODAs are supported by a dedicated Oracle engineered system support team that has an in-depth knowledge of the platform.

The majority of the hardware components are redundant. An ODA has two server nodes and redundant power supplies. The system interconnects between the system components are redundant and are implemented by SAS cables in the current X3-2 model.

Disk replacements are easy to perform on an ODA. Simply flip a switch on the front of the storage unit, pull the drive out, slide the new drive in, and flip the switch back.

ODAs reduce the number of moving parts needed to implement high availability. Fewer moving parts and fewer handoffs result in higher availability in most cases.

There are a number of high availability deployment patterns:

- *Active-Active.* This pattern is implemented as RAC deployed on an ODA, providing the ability to load balance processing across two nodes and failover between the two servers.
- *Active-Passive.* This pattern is implemented through RAC One, RAC, and/or Data Guard. Processing is configured to run on a single node. Failover is supported by RAC One or RAC within an ODA. Failover across ODAs, including the ability to failover across geographic distances, is handled by Data Guard.

Achieving high availability requires additional designs, such as RAC services, and role-based services if Data Guard is part of the solution. The effectiveness of your high availability implementation will be directly proportional to the amount of design and testing that go into the solution.

Costs and Value Proposition

ODAs do not fit the traditional build vs. buy purchasing model. A company can certainly choose to build their own commodity hardware-based infrastructure. However, the typical company doesn't have the resources to build its own engineered system. Regardless, check out the Oracle "build vs. buy" ODA videos. You'll find some videos on the main Oracle ODA product page and Oracle's YouTube channel detailing the ease of deployment and the reasons why you want to give serious consideration to the value proposition of buying a better, prepackaged solution. Some of the videos have an element of humor, so they are very viewable. The content is dynamic, so we're not including any web links because they are subject to change. However, finding them is easy enough.

From a hardware perspective, you get a lot of resources when you purchase an ODA. For the current X3-2 model, this includes two servers with 256G memory each, with a total of 32 cores, and a lot of storage. The current storage volume is 18T of raw disk, which can be deployed as approximately 9T usable double mirrored, or 6T usable triple mirrored. The storage can be doubled again to 36T of raw storage with a storage expansion rack. The ODA Getting Started manual details the exact amount of usable storage, depending on the deployment options that are selected.

However, the secret sauce of the ODA is not the hardware. The secret sauce of an ODA is the software. The ingredients include RAC deployments in hours, the automated patching of all tiers (BIOS, firmware, OS, ILOM, grid, database) resulting in systems that are kept up-to-date, stable, and compliant. The

ODA hardware—with the standard, embedded Oracle hardware support facilities and accompanied by a lot of value-added management software—offers a great value proposition. Benefits include the following:

- ODAs come with an embedded validation toolkit (`oakcli`—Oracle appliance kit command-line interface). `oakcli` contains a full set of diagnostic and management utilities to maintain the "keep it simple" management theme of ODAs.
- An ODA allows you to cap the licensing and resources that are deployed on the hardware in a "pay as you grow model" vs. paying for all of the capacity and licensing up front. Additional capacity is enabled through applying a key.
- The prepackaged software and deployment model provides the ability to deploy a standardized database platform across your database portfolio. ODAs are a solution to the traditional model of building servers and databases by hand, resulting in a data center full of "totem poles," where no two totem poles look the same.

There is no question that the ODA standardized deployment model lowers management and administration costs. This is in part due to the reduced number of build steps and the reduction in the number of people and teams that need to become involved, as well as reducing all of the handoffs between teams. After an ODA is deployed, the unified nature of the system and platform-specific automations lower ongoing administration costs. The ongoing administrative savings come from the combination of reducing the number of teams needed to support an ODA, the platform's stability, automated patching of all of the appliance components, a lower number of moving parts, server and component redundancy, appliance kit automation, and the embedded support facilities, including ASR and the ILOM.

The rapid setup and standardized model is an excellent building block for DBaaS or cloud services. The process to create new single instance, RAC One and RAC databases on an ODA has been reduced to 15 minutes. ODAs are engineered systems that can be deployed by companies of any size. ODAs often serve as low-cost lower–life cycle environments for Exadata, as well as an entry point and starting success story for companies that want to invest in Exadata in the long term.

Oracle is investing in the platform. New functionality is released every quarter, and the software changes are backward compatible to previous hardware generations. ODAs started as a small to midsized company solution, but then big companies started buying them. This helped to drive more momentum for the ODA product. Since ODAs are a single vendor product, intervendor handoffs are eliminated, and the testing of all components as a unit is self-contained within Oracle. From a troubleshooting standpoint, customers only have one vendor that needs to be contacted for all support issues. When a support call is initiated, it is handled by a specialized engineered system support team.

The ability to deploy infrastructure quickly, or pre-deploy infrastructure, has overall savings by getting projects launched quickly to achieve business benefits. ODAs come with sufficient resources to serve as a consolidation platform and support your internal shared services initiatives. ODAs also support the standard Oracle resource management facilities, such as instance caging and database resource management to support consolidation efforts.

In the end, companies need to decide how much work and overhead they want to take on by developing their own solution vs. buying a prepackaged solution like the ODA. A key benefit of the ODA is that it lets companies spend their time on core business functions instead of routine support tasks.

What about the *total cost of ownership* (TCO) numbers? Oracle publishes numbers—just like all vendors do for their competing products. Oracle's ODA TCO studies can currently be found on Oracle's main ODA product page. Oracle's numbers can be helpful in supporting your initial purchase. However, the best TCO numbers are the ones that you put together for your company as you gain experience with the platform.

Oracle Hardware

ODAs take advantage of the standard Oracle (formerly Sun) hardware facilities to administer the appliance and harden it from routine failures. The management facilities include the ILOM and Automatic Service Request (ASR) facilities. As you've seen previously in this book, ILOMs let you remotely administer the hardware, such as powering it down and up, and running a complete set of diagnostics. The ILOM functionality is so comprehensive that at the time of this writing, Oracle has published eight separate manuals to cover all of the facilities. Similarly, the ASR software provides phone-home capabilities to automatically issue alerts and upload hardware fault messages to Oracle to initiate the service request process.

The ODA hardware is both modular and resilient. The servers are modular, consisting of two separate units cabled together. Similarly, the storage is modular. The power supplies are redundant and field replaceable. The disk drives are hot replaceable.

The Oracle hardware is supported by a field services unit that will fix any hardware failures that are encountered. Backing up the field services unit is the Oracle support organization that will respond to service requests by assisting with diagnostics to determine the cause of the issue, and then dispatch the field services group to fix the problem.

Security and Compliance

The first security feature of ODAs is the inclusion of the quarterly PSU bug fix and security patching in the quarterly ODA release. The quarterly ODA patches are documented in MOS note 888888.1. Oracle will release the quarterly ODA releases approximately two to four weeks after Oracle releases the quarterly PSU patch set. Oracle follows this approach to include the latest security patches in each ODA release, and continues the process of testing all components of a patch set as a complete unit. The ease of the one-button patching process is a key enabler of keeping ODAs patched regularly to keep the systems compliant. Keeping systems patched with the most up-to-date security patches is a requirement of PCI (the credit card industry) and other security compliance certifications.

The second security implementation feature for ODAs is security scanning during the development release process. Oracle uses security scanning software from a third-party vendor to independently perform these checks. The scan results are used to upgrade the Oracle Linux package versions deployed on an ODA to keep the system compliant.

The third security mechanism for locking down ODAs is the ODA-specific Oracle-supplied STIG script (MOS notes 1456609.1 and 1461102.1). The STIG script is part of the US Department of Defense's Security Technical Implementation Guide (STIG) process. Running the STIG script is a two-step process. In first step, the script can be run in check mode to search for security violations. In the second step, the "fix" process allows these flagged violations to be corrected.

Some of the STIG security checks include the following list, developed by looking at the code. The list of security checks performed is not detailed in the MOS notes.

- **Category 1: This is DOD speak for "You had better fix this."**
  - Is the password for `grub` enabled?
  - Is the `sendmail decode` command commented out in `/etc/aliases`?
  - Is the privilege account `halt` present?
  - Is the Ctrl-Alt-Del combination available to shut down the system?
  - Is the RealVNC rpm installed on the server?
  - Is support for a USB device found in the kernel?
- **Category 2: This is DOD speak for "Document these and develop a plan to fix the issues."**
  - Is single user mode boot-enabled without a password?
  - Is the pam-tally account configuration and login failure management tool configured to lock accounts after three consecutive login failures?
  - Does the system prohibit the use of past passwords?
  - How secure is the password strength? Is the password less than eight characters?
  - Is a delay configured to make users wait before trying to log in again after a login failure?
  - Do passwords have to be changed no less than every 60 days?
  - Can passwords changed be more than once every 24 hours?
  - Is `cron` access controlled?
  - Can you log into the system directly as root through `ssh`?
  - Is the `tcp dump` rpm installed on the system?
  - Do all of the file systems have the correct permissions?
  - Are there any unnecessary accounts present?
  - Is the `sendmail help` command enabled?
- **Category 3: This is DOD speak for "Document the risk and decide what you want to do about it."**
  - Are the UNIX `man` page permissions correct?
  - Is the `sendmail` version hidden?

There are some additional checks in the STIG scripts that even your seasoned sysadmins may need to look up and figure out. The purpose of the STIG scripts is to check your security setup practices rather than the ODA out-of-the-box security implementations. A search of "My Oracle Support (MOS)" only found STIG implementation documents for ODAs and Exadatas. This security implementation check script is another value-added proposition for ODAs.

The implementation of which STIG script fixes that you choose to implement will depend on your company's security standards. Not every company has the same security requirements as the US Department of Defense. However, security is an area where some companies choose to err on the side of caution. At the minimum, the STIG process performs an ODA-specific DOD security analysis.

Oracle also publishes Oracle Linux security manuals and the Oracle Linux group publishes additional blogs for steps to lock down your systems. While it isn't always easy to translate these steps directly to an ODA, they do serve as valuable guides for security lockdowns on your systems. While a full coverage of server security implementation is outside the scope of this chapter, additional steps can be taken to lock down your ODA systems, such as limiting access to ODAs through jump servers.

Virtualization

In March 2013, Oracle released support for the Oracle virtualization solutions on ODAs in software release 2.5. Oracle has since been enhancing the virtualization capabilities of ODAs with every release. Virtualization on ODAs is a major business case for the use of ODAs. The Oracle virtual machine implementation on ODAs is currently unique because the virtualization has been optimized for databases. Expect virtualization to continue to be a major initiative for Oracle, including enhancing the capabilities on ODAs.

Oracle has simplified the management of ODA virtual machines by implementing VM management in `oakcli`, including the ability to clone, start, and stop VMs. `oakcli` has been enhanced to import templates. There is no separate VM manager needed to support virtualization on an ODA.

The ability to "pay as you grow" on the ODA platform means that additional capacity can be available on ODAs to run nondatabase applications and other infrastructure-support software. Oracle has coined the phrase "Solution in a Box" for ODA-packaged application solutions. Oracle is porting their software to ODA-specific VM templates. At the time of this writing, a WebLogic template has been released for ODAs, and a JD Edwards template had been announced. Additional templates for more Oracle software products are likely to come. Oracle's partners have jumped on the bandwagon by beginning to develop their own application "solutions in a box."

The business value proposition to maximize the use of the hardware and to rapidly deploy applications is huge. Companies of every size often deploy large consulting firms, system integrators, and consulting staffs to deploy complex software. Deploying applications on ODAs through VM templates offers the promise for major cost savings through the rapid deployment of standardized images. Oracle has stated that they will be releasing additional ODA VM templates in the future.

ODA Technical Solutions

ODAs are a technical "solution in a box." That solution includes a number of major components that you should be aware of. We introduce those in this section.

The Oracle Appliance Kit (`oakcli`) is a built-in appliance management jack-of-all-trades. The `oakcli` command reference can be found in MOS 1417879.1, the latest ODA version release notes, and the ODA Getting Started Guide. The ODA Release Notes and Getting Started Guide are generally more current than the MOS notes. New `oakcli` functionality is released quarterly, so you have to proactively check the quarterly release notes to keep up with the newest features.

Here are some of the things you can do with `oakcli`:

- **ODA management**
  - Deploy an ODA
  - Patch an ODA
  - Create a new database in approximately 15 minutes
  - Create new database software home from the current ODA release
  - Delete databases and database homes
  - Run a complete check of the system
  - Run a disk calibration
  - Configure and test ASR (Automatic Service Request)
  - Display the hardware details and versions
  - Display software and firmware versions
  - Display disks, disk groups, and other storage details
  - Clean up patches that have been applied
  - Activate cores when additional Oracle licenses are purchased
- **Diagnostics**
  - Run a complete check of the system
  - Locate a disk drive and light up its LED indicator to signal which specific disk drive needs to be replaced
  - Run disk diagnostics
  - Run an ODAchk health check
  - Collect and package diagnostic information
- **Virtualization**
  - Create or clone an ODA virtual machine
  - onfigure resources on an ODA virtual machine
  - Start and stop ODA_Base (the database partition on a virtualized ODA) or another ODA virtual machine
  - Display the information for ODA VM templates
  - Deploy/import an Oracle VM template
  - Create CPU pools for ODA virtual machines
  - Display all of the details for one or all of virtual machines on an ODA
  - Apply IPs to an ODA virtual machine
  - Open a VM console to manage an ODA VM

Oracle also provides additional ODA diagnostic utilities as part of the ODA software product set. The main utility is ODAchk (MOS 1485630.1). This is an ODA-specific version of RACcheck. Oracle recommends that you deploy the latest version of ODAchk, so you have to check the MOS note as new versions are released following changes to the RACcheck and Exadata Exachk tools.

ODAs ship with the standard `top`, `vmstat`, and `sar` server resource monitoring utilities. ODAs also ship with the detailed server resource collection utility OSWatcher (MOS 301137.1 and 461053.1). Two additional standard RAC utilities common to self-install platforms are also included with the ODA RAC software:

- Cluster Health Monitor (MOS 736752.1 and 1328466.1)—Granular resource monitoring
- Trace file analyzer utility (MOS 1513912.1) —Real-time event capture

ODAs ship with a set of database templates designed for performance and consolidating DBs on ODAs. The Database templates sizing is documented in the Getting Started Guide. The templates range in size from very small, small, medium, large, extra-large, and extra-extra large. Besides configuring memory, redo log sizes, and the starting DB size, the templates determine the number of databases that can be deployed on an ODA.

The following are some other software utilities and tools to be aware of:

- ODAs come with two utilities that allow GUI-based tool sessions to be started. They are VNC Server and StartX from the ILOM remote console.
- Server software includes the previously discussed ILOM and ASR facilities. These server management utilities are an extra Oracle hardware value-added proposition vs. generic commodity hardware.
- ODAs pre-deploy a utility called Logwatch, which monitors file system space and authentication failures and logins through `ssh`. It is automatically scheduled.
- ODAs can run other proactive support tools (MOS 1459344.1). A good example is `oratop` (1500864.1), which produces a one-panel display of overall database and database server activity.

ODAs have a unique "worst case" recovery capability. When all is lost, and the server OS disk configurations are out of action and the databases can't be started, ODAs provide the ability to rebuild the appliance all over again using a bare metal restore, as long as the source of the problem isn't a major hardware failure. An ODA can be rebuilt in about two hours using this method. Of course, you will need a backup of your database and any additional objects that need to be restored to get the ODA in an operational state, such as exports and the OCR for RAC services.

Finally, ODAs come with a license for Oracle's Secure Backup product to use in backing up ODAs.

ODA Performance

The current X3-2 ODA model runs the same compute node model as an Exadata. ODAs do lag the Exadata server node release cycle. Regardless, they are running the same CPU model most of the time. Exadatas support a memory expansion from 256G to 512G per server, but ODAs do not. Regardless, logical IO performance on an ODA is excellent.

Another source of ODA performance is that everything is self-contained within the appliance. RAC traffic doesn't leave the appliance and travel through an external network to communicate between the compute nodes. The two server nodes and storage units are all next to each other and are connected through high-speed SAS cables. There are no intervening switches, firewalls, or other external network layers.

ODA physical IO storage performance is good—in fact, better than the typical SAN fiber channel disk. Oracle has published X3-2 Physical IO benchmarks of 5 milliseconds at 3,500 IOPS on a single storage unit, and 7,000 IOPS with an additional storage expansion unit. Above those IOPS levels, the service levels decrease. Oracle's ODA X3-2 benchmarks show service levels of 6 to 7 milliseconds at 5,750 IOPS, and 11,500 with a storage expansion rack. These last numbers are still very good and compare nicely to the levels seen on a well-tuned Fiber Channel disk-based SAN. However, as the IOPS build above these levels, at some point performance will start to drop noticeably. Since ODAs come with a lot of memory, caching data in memory is a key scalability factor to avoid reaching the ODA IOPS limitations.

(continued on page 26)

NoCOUG Spring Conference

Session Descriptions

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

–Keynote–

With Oracle Database 12*c*, there is all the more reason to use database PL/SQL—*Bryn Llewellyn, Oracle Corp. 9:30–10:30*

Fifty-year-old wisdom instructs us to expose the database to client-side code as a PL/SQL API and to securely hide the implementation details—tables, and the SQL statements that manipulate them. Yet many customers don't do this. They claim that PL/SQL in the database cannot be patched without causing unacceptable downtime; that the datatypes for passing row sets between the client-side code and the database PL/SQL are cumbersome to use and bring performance problems; and that it is impractical to give each developer a private sandbox within which to make changes to database PL/SQL. This session describes changes brought by Oracle Database 12*c* that demolish each of these objections.

Bryn Llewellyn has worked in the software field for over 30 years. He joined Oracle UK in 1990 at the European Development Center in the Oracle Designer team. He transferred to the Oracle Text team and then into Consulting as the Text specialist for Europe. He relocated to Redwood Shores in 1996 to join the Oracle Text Technical Marketing Group. He has been the product manager for PL/SQL since 2001. In 2005, he took on the additional product management responsibility for the Oracle Database capabilities that support online application upgrade. (The main one of these is called edition-based redefinition—EBR for short.) It's hard for Bryn to remember his life before Oracle. He started off doing image analysis and pattern recognition at Oxford University. He then worked in Oslo, in the public sector and in a startup. While in Norway, Bryn programmed in Simula, whose inventors were his close colleagues. This language was the inspiration for Smalltalk and C++.

–Room 106–

Edition-Based Redefinition: The Key to Online Application Upgrade—*Bryn Llewellyn, Oracle Corporation 11:00–12:00*

Large, mission critical applications built on Oracle Database are often unavailable for tens of hours while the application's database objects are patched or upgraded. Oracle Database 11*g* Release 2 introduced revolutionary new capabilities that allow online application upgrade with uninterrupted availability of the application. These have been enhanced in Oracle Database 12*c*. Existing sessions can continue to use the pre-upgrade application until their users decide to finish; and, at the same time, new sessions can use the post-upgrade application. When no sessions are any longer using the pre-upgrade application, it can be retired. The application as a whole therefore enjoys hot rollover from the pre-upgrade version to the post-upgrade version. This session explains how it all works.

PL/SQL Enhancements in Oracle Database 12*c*
—Bryn Llewellyn, Oracle Corporation 1:00–2:00

Oracle Database 12*c* brings PL/SQL enhancements in these categories: transparent performance improvements; new PL/SQL language features for better performance, new functionality, and improved programming usability; and new supplied package APIs. This session explains improvements in PL/SQL-to-SQL interoperability, improvements in how PL/SQL subprograms can be invoked from outside the database, new constructs to improve modularization, and new APIs to assist tracing and the diagnosis of runtime errors. It also describes other improvements.

Doing SQL from PL/SQL: Best and Worst Practices—*Bryn Llewellyn, Oracle Corporation . . . . . .2:30–3:30* **Editor's Pick**

The PL/SQL developer has many constructs for executing SQL statements, and the space of possibilities has several dimensions: embedded SQL versus native dynamic SQL versus the DBMS_SQL API; bulk versus non-bulk; implicit cursor versus parameterized explicit cursor versus ref cursor; and so on. Deciding which to use might seem daunting. Moreover, as new variants have been introduced, older ones sometimes have ceased to be the optimal choice. This session examines and categorizes the use cases for doing SQL from PL/SQL and explains the optimal approach for the task at hand.

–Room 107–

Live Demo of Heat Map and Automatic Data Optimization in Oracle Database 12*c*
—Ganesh Sankar Balabharathi, Fusion-io 11:00–12:00

Enterprise data doubles every two to three years. Database performance degrades and the storage cost increases. Oracle's Information Lifecycle Management (ILM) solution provides ways to assign data to different storage and compression tiers. Heat Map in Oracle Database 12*c* tracks database usage at the row and segment levels while Automatic Data Optimization (ADO) automatically moves and/or compresses the data according to user-defined policies.

Live Demo:

1. ILM, Heat Map, and ADO options.
2. Enable Heat Map.
3. Setup demo tables (massive data on flash storage).
4. Define ADO policies.
5. Manually enforce ADO policies.
6. Check the effects of ADO.

Ganesh Sankar Balabharathi has been a database administrator for over 15+ years and currently works as an Oracle solutions architect. Ganesh has hands-on experience in large database installations, resolving performance issues, and establishing backup/recovery and business continuity plans.

(continued on page 26)

Many Thanks to Our Sponsors

NoCOUG would like to acknowledge and thank our generous sponsors for their contributions. Without this sponsorship, it would not be possible to present regular events while offering low-cost memberships. If your company is able to offer sponsorship at any level, please contact NoCOUG's president, Hanan Hit. ▲

Long-term event sponsorship:

Chevron

Oracle Corp.

Thank you! Gold Vendors:

- Axxana
- Confio
- Database Specialists
- Delphix
- Embarcadero Technologies
- Kaminario

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

TREASURER'S REPORT

Dharmendra Rai, *Treasurer*

Beginning Balance		
January 1, 2014		**$ 56,867.13**
Revenue		
Individual Membership	5,295.00	
Corporate Membership	8,250.00	
Gold Vendor	4,000.00	
Silver Vendor	1,000.00	
Conference Walk-In	900.00	
Conference Sponsorship	–	
Training Day	–	
Journal Advertisement	1,000.00	
Interest	1.48	
Total Revenue		**$ 20,446.48**
Expenses		
Conference	8,733.61	
Training Day	–	
Meetup	–	
Journal	3,671.09	
Board	680.29	
Paypal	596.00	
Office	149.00	
Marketing	–	
Insurance	–	
Taxes and Filings	–	
Software Dues	–	
Total Expenses		**$ 13,829.99**
Ending Balance		
March 31, 2014		**$ 63,483.62**

(continued from page 24)

Memory Structure Control: How It Works (Latches & Mutexes) —*Craig Shallahamer, OraPub* 1:00–2:00

Memory structure control is core to all Oracle database operations and one of the most fascinating Oracle internals and performance topics. Memory structure activity must be strictly controlled or the database gets corrupted. This presentation will explore how latches and mutexes are implemented, their algorithms, where diagnostic data comes from, and how Oracle continues to improve serialization control in newer releases. The cache buffer chains and the library cache are used as examples throughout the presentation. This is a very practical yet deep internals presentation, filled with amazing discoveries about how Oracle works.

*Craig Shallahamer is a long-time Oracle DBA (since 1989) who specializes in Oracle performance. He is also an Oracle ACE Director, a performance researcher and blogger, a consultant, an author of two books (*Oracle Performance Firefighting, Forecasting Oracle Performance*), an enthusiastic conference speaker, and a passionate teacher to thousands of Oracle professionals. He is the founder and president of OraPub, the creator of the performance analysis tool Stori, and he clearly pushes the teaching envelope with his online seminars. His blog, "A Wider View," is where to find his most recent public performance research. He can be contacted at craig@orapub.com, @CShallahamer, LinkedIn, and Xing.*

Two Creative Applications of Serialization Control —*Craig Shallahamer, OraPub* 2:30–3:30

Now that you understand memory structure control, let's apply it in a couple of non-traditional yet practical areas. First, I will focus on cache buffer chain performance analysis. I will lead you step-by-step through the process focusing on situations when there is a hot buffer. Second, I will teach you why tuning serialization control broadly impacts performance and then how you can use this knowledge to anticipate the elapsed time of a specific SQL statement. It will be a fast-paced hour, filled with amazing insights.

–Room 116–

Big Data Lab—*Gwen Shapira, Cloudera* 9:00–12:00

This tutorial is for experienced database professionals who are interested in using Hadoop as a way to scale ETL processes. In the tutorial, we will look at an entire ETL process and for each stage demonstrate how to implement it in Hadoop.

The tutorial will contain a live demonstration of Hadoop ETL techniques. We will provide demo Hadoop servers in the cloud for all attendees to run the examples with prepared datasets. Attendees will need to bring their own laptops. Knowledge of Hadoop is not required.

We will begin with a quick Hadoop primer, in which we will explain Hadoop's architecture and its core components: Map Reduce and HDFS. The rest of the tutorial will build on this understanding and show how Hadoop's unique architecture can be applied to optimize ETL processes.

The tutorial will continue with a discussion of different data ingestion methods and a demonstration of Sqoop to pull data from MySQL to Hadoop. Once the data is loaded to the system we will demonstrate the use of Hive for data transformation. We'll compare HQL to more familiar SQL, and attendees will experiment with running Hive queries and using Hive UDFs. We will also demonstrate the use of Oozie to manage the ETL workflow.

Attendees will leave knowing how to take the first steps in using Hadoop as part of their data processing pipeline and familiar with some of the popular tools they can use in their implementation.

Gwen Shapira is a Solutions Architect at Cloudera, where she helps customers build productions applications using Hadoop ecosystem components. With 15 years of data warehouse experience, Gwen loves showing customers how open source tools can be used to build a faster and more scalable data warehouse. Gwen shares her experience on her blog **http://prodlife.wordpress.com**, *on twitter (@gwenshap) and at conferences.*

RAC Attack—*Terry Sutton and Ian Jones, Database Specialists* 1:00–4:00

Install and configure Oracle Database 12*c* RAC on your own laptop with the assistance of the RAC experts at Database Specialists.

Terry Sutton is the Director of Managed Services at Database Specialists. He is a specialist in Oracle Database administration—particularly remote administration—and performance optimization. Ian Jones is a Senior DBA with Database Specialists. He has administered Oracle databases for 25 years in the U.K. and the U.S. Nowadays he works as a "remote DBA," assisting a wide range of clients across North America and beyond.

(continued from page 22)

ODAs do not currently have the supported option to use PCI Flash or SSD. The four SSD drives that come with ODAs are only supported for use by the online redo logs. ODAs do support expanding storage to NFS, including Oracle's ZFS storage appliance. The use of DNFS (Direct NFS) to increase performance is supported. However, I'm not aware of any physical IO performance benchmarks that have been published for ODAs using an NFS storage extension. While ASM is not supported on the NFS extension, Oracle has announced that ODAs do support Hybrid Columnar Compression when the NFS mount is an Oracle ZFS storage appliance.

Since Oracle is investing in the ODA platform, the IOPS and disk performance may change with each new model. You will need to relook at the physical performance and IOPS limits when a new model that replaces the X3-2 is released.

Summary

ODAs are a high availability application and database infrastructure solution in a self-contained engineered appliance. ODAs offer low-cost hardware with lower setup and ongoing support costs than traditional database solutions. Deployments are automated and fast, including the creation of RAC One and two-node RAC cluster databases. The ODA patching process patches the OS, grid, and database components as a single unit using Oracle supplied software. ODAs come prepackaged with management and deployment software.

ODAs offer a unique "pay as you grow" Oracle licensing option. Applications and databases can be deployed on ODAs using Oracle's virtualization product. ▲

Database Specialists: DBA Pro Service

DBA PRO BENEFITS

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

CUSTOMIZABLE SERVICE PLANS FOR ORACLE SYSTEMS

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

DBA Pro's mix and match service components

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

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

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

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

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

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

DatabaseSpecialists

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

NoCOUG
P.O. Box 3282
Danville, CA 94526

RETURN SERVICE REQUESTED

NoCOUG Spring Conference Schedule

Thursday, May 15, 2014—UCSC Extension Silicon Valley, Santa Clara, CA

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

8:00–9:00 a.m.	Registration and Continental Breakfast—Refreshments served
9:00–12:00	**Lab #1** **Room 116:** *Big Data Lab*—Gwen Shapira, Cloudera
9:00–9:30	**Welcome:** Hanan Hit, NoCOUG president
9:30–10:30	**Keynote:** *With Oracle Database 12c, there is all the more reason to use database PL/SQL* —Bryn Llewellyn, Oracle Corporation
10:30–11:00	**Break**
11:00–12:00	**Parallel Sessions #1**
	Room 106: *Edition-Based Redefinition: The Key to Online Application Upgrade*—Bryn Llewellyn, Oracle Corporation
	Room 107: *Live Demo of Heat Map and Automatic Data Optimization in Oracle Database 12c* —Ganesh Sankar Balabharathi, Fusion-io
12:00–1:00 p.m.	**Lunch**
1:00–4:00	**Lab #2** **Room 116:** *RAC Attack*—Terry Sutton and Ian Jones, Database Specialists
1:00–2:00	**Parallel Sessions #2**
	Room 106: *PL/SQL Enhancements in Oracle Database 12c*—Bryn Llewellyn, Oracle Corporation
	Room 107: *Memory Structure Control: How It Works (Latches & Mutexes)*—Craig Shallahamer, OraPub
2:00–2:30	**Break and Refreshments**
2:30–3:30	**Parallel Sessions #3**
	Room 106: *Doing SQL from PL/SQL: Best and Worst Practices*—Bryn Llewellyn, Oracle Corporation **Journal Editor's Pick**
	Room 107: *Two Creative Applications of Serialization Control*—Craig Shallahamer, OraPub
3:30–4:00	**Raffle**
4:00–6:00	**NoCOUG Networking and No-Host Happy Hour at Faultline Brewing Company, Sunnyvale**

RSVP *required* at http://www.nocoug.org

The *NoCOUG Journal* design and production: Giraffex, Inc., S.F.
Front cover photo by Comstock, Photos.com.