

Official Publication of the Northern California Oracle Users Group

NoCOUG

J O U R N A L

Vol. 20, No. 1 · FEBRUARY 2006

\$15

Adapt to Changing Times with NoCOUG

The Present and Future of Database Administration

An interview with Venkat Devraj. See page 4.

Hello, World!

A new column on programming languages. Guest columnist Catherine Devlin introduces us to Python. See page 13.

Book Review

Our regular columnist Brian Hitchcock reviews a thought-provoking book, Oracle Insights: Tales of the Oak Table. See page 18.

Big discount on Jonathan Lewis seminar! See page 27.

Is this your last issue? See page 27.

Much More Inside ...

Passing the Torch . . .

The only thing constant is change! Lisa Loper has passed the torch to me after four stellar years at the helm of the *NoCOUG Journal*. I am overwhelmed with the responsibility but grateful that Lisa did such a wonderful job of streamlining the production process during her tenure as editor. I am also grateful for all the patient coaching that she gave me during the last two months.

This issue is jam-packed with interesting and educational material in keeping with NoCOUG's mission of being "dedicated to the education and representation of the users of Oracle Corporation's database and tools software." Thanks to all the volunteers who contributed material for this issue, to our vendors who produced such a beautiful final product, and to Tom Wagner for contributing the pictures of Mount Timpanogos for the front and back covers.

Here's wishing you every success with all the changes you encounter in 2006. NoCOUG and the *NoCOUG Journal* are here to help you.

Happy New Year!

Iggy Fernandez,
NoCOUG Journal Editor

Table of Contents

Editor's Note.....2	NoCOUG Winter Conference Session Descriptions.....24
NoCOUG Board.....2	Advertising Rates..... 26
Publication and Submission Format.....2	Optimizing Oracle—Performance by Design.... 27
President's Message.....3	NoCOUG Fall Conference Schedule..... 28
The Present and Future of Database Administration.....4	—ADVERTISERS—
Transportable Table Partitions.....8	IT Convergence.....6
Hello, World!.....13	Quovera.....10
SQL Corner.....16	Data Domain.....10
Book Review.....18	Quest Software.....12
News Roundup.....22	Embarcadero Technologies.....14
Sponsorship Appreciation.....23	Database Specialists.....20
Treasurer's Report.....23	Confio Software.....20

Publication and Submission Format

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

The submission deadline for the upcoming May 2006 issue is April 1, 2006. Article submissions should be made in electronic format via email if possible. Word documents are preferred.

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

Copyright © 2006 by the Northern California Oracle Users Group. Permission to reproduce articles from this publication, in whole or in part, is given to other computer user groups for nonprofit use, with appropriate credit to the original author and the *Northern California Oracle Users Group Journal*. All other reproduction is strictly prohibited without written permission of the editor. Two copies of each reprint should be sent to the editor.

NoCOUG BOARD

President

Darrin Swan, Quest Software
darrin.swan@quest.com

Vice President

Jen Hong, Gap
jen_hong@gap.com

Membership Director

Joel Rosingana, Independent Consultant
joelros@pacbell.net

Director of Conference Programming/Past President

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

Webmaster

Eric Hutchinson, Independent Consultant
erichutchinson@comcast.net

Journal Editor

Iggy Fernandez, MCI
iggy_fernandez@hotmail.com

Vendor Coordinator

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

Secretary/Treasurer

Diane Lee, Lockheed Martin
diane.c.lee@lmco.com

Director of Marketing

Naren Nagtode,
Franklin Templeton Investments
nnagtod@frk.com

Training Day Coordinator

Hamid Minoui, Database Specialists, Inc.
hminoui@pacbell.net

Track Leader

Randy Samberg, PalmSource
rsamberg@sbcglobal.net

NoCOUG Staff

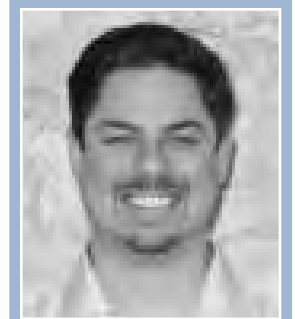
Nora Rosingana,
Independent Consultant

Other Contributors

Vilin Roufchaie
Brian Hitchcock

Adapt to Changing Times with NoCOUG

by Darrin Swan



Darrin Swan

Happy New Year! I hope you and your families enjoyed a safe and festive holiday season. Change is our theme.

There is an old saying that “the only thing constant is change.” The *NoCOUG Journal* is one of the tools that you can use to keep up with the rapidly changing world out there.

*“Here is the chance to take charge of our fate,
Deep down you must know that tomorrow’s too late!
One rule of life we cannot rearrange,
The only thing constant is change,
The only thing constant is change.”*

—Song from the 1990
Broadway musical *Jekyll and Hyde*

In 2006, expect to hear more and more about the changes that impact you as an Oracle professional, including Fusion Middleware, web services, new 10g capabilities, and Business Process Execution Language (BPEL). We have changes at NoCOUG as well, and our affiliation with IOUG brings new benefits to you. This year, we will be sure to pack our conferences and journal with the education that you need to better prepare for and address changes in the world of Oracle.

NoCOUG Winter Conference

The Winter Conference will take place on Tuesday, February 14, 2006, at Oracle’s Redwood Shores headquarters. This full-day conference will start with a keynote address on Fusion’s value proposition by Vijay Tella, Vice President and Chief Strategy Officer of Oracle Corporation. Refer to the back cover of the journal for the complete agenda.

Our IOUG affiliation

In 2006 the Independent Oracle Users Group (IOUG, formerly the International Oracle Users Group) offers additional benefits to NoCOUG and other affiliated regional user groups. We can now automatically sign up new NoCOUG members for a free IOUG Associate membership. As an Associate member of the IOUG, you receive Associate access to the IOUG website, the Tip-of-the-Month, the twice-monthly Associate E-Newsletter, the Five-Minute Briefing, and access to IOUG discussion forums.

Visit www.ioug.org to learn more about the IOUG.

NoCOUG board for 2006

There are several changes on the all-volunteer NoCOUG board of directors. Jen Hong from Gap is taking on the role

of Vice President. Lisa Loper from Database Specialists has passed the *NoCOUG Journal* Editor reins to Iggy Fernandez from MCI. Lisa will take over the Vendor Coordinator role because Diane Lee from Lockheed is now Treasurer and Secretary. Roger Schrag, Joel Rosingana, Nora Rosingana, Hamid Minoui, Eric Hutchinson, and Randy Samberg are maintaining their 2005 roles in '06. Naren Nagtode will become Director of Marketing, and I will continue as President.

Training days with Jonathan Lewis

In addition to holding the quarterly conferences, NoCOUG takes pleasure in facilitating a three-day seminar led by Jonathan Lewis. This is a very intensive seminar on how to get the best out of the Oracle database engine. NoCOUG members will receive a very substantial discount that covers the cost of your NoCOUG membership many times over. Refer to the inside back cover of the journal for additional details, and visit www.jlcomp.demon.co.uk/seminar.html for a complete course description as well as rave reviews by very satisfied attendees of this seminar in the past.

Free lunch for NoCOUG members!

Another old saying is that “there is no such thing as a free lunch.” I might add that this may not be strictly true in the case of NoCOUG. Sponsor support helps us to keep membership fees low, and allows us to bring you four action-packed conferences each year (with continental breakfast, lunch, and snacks), four issues of our award-winning journal, and—lest we forget—significant discounts on NoCOUG-facilitated training events by Oracle gurus. However, as with public television, we depend heavily on your membership dollars to pay the bills. If you have not paid your 2006 membership dues, this issue of the journal is complimentary because past experience tells us that you will sign up again. Please take a moment to visit www.nocoug.org/join.html and sign up for another year.

Appreciation

I would like to thank the all-volunteer NoCOUG board, NoCOUG staff, sponsors, volunteers, and vendors for making 2005 a successful and valuable year for Oracle professionals in Northern California. As changes occur in the new year, we look forward to keeping you up-to-date and informed to make 2006 another successful and valuable year.

See you at Oracle on February 14! ▲

The Present and Future of Database Administration

An Interview with Venkat Devraj

We are pleased to have the opportunity to introduce NoCOUG members to Venkat Devraj, author of *Oracle 24x7 Tips and Techniques*. He is working on a new book—on a hot new subject—titled *Building Scalable Systems with Autonomics*. Venkat is the founder and chief architect of ExtraQuest Corporation, which is based in Denver.

Tell us a little bit about your background. Where did you study, and how did you become so knowledgeable about Oracle?

I started working full time when I was 14 and ended up in a job that required me to program heavily in C. I worked my way through school, acquired a bachelors' in business and finance, and then signed up for a masters' program in software engineering. During that time, I found myself going from flat-file systems built in C and COBOL to Xbase environments and then to RDBMSs. I started off in Oracle 5.1 as a developer. The mid-'90s saw me migrating to the systems, network, and database administration side of things.

Wanderlust eventually took over, and I ended up in a project in the Middle East. After that project, I moved to the U.S., working as a performance and disaster recovery consultant for several companies.

You also seem to have some entrepreneurial spirit, founding a couple of companies. How do you account for this? Is it in your blood?

For me, it's never been about building a company for putting lots of money in my pocket. I repeatedly saw huge inefficiencies in the way database administration was done and decided to do something about it.

DBAs are busy fighting fires and, unfortunately, ignoring what they really need to be doing. They work hard, but often end up directing their energies at the symptom rather than the cause. If an application has database problems after it goes live, that to me clearly means the DBA didn't have the resources to get the project done right. In the end, this is not a DBA problem—it's a

managerial problem.

Tools vendors provide a plethora of “point solutions” catering to the low hanging fruits in the DBA workload—such as monitoring and alerting—which only foster reactive work patterns—repairing mistakes of the large DBMS vendors—mistakes that should never have been there in the first place. Companies worldwide pay for these mistakes every day by retaining an army of DBAs to baby-sit the database.

I saw that this prevailing juggernaut could not be effectively controlled via the existing DBA methods and tools. If the typically reactive database administrator (let's call this role “DBA-1”) needs to get ahead of the curve and transition into more of a proactive database architect (call this “DBA-2”), they need to separate and offload the lower-level mundane tasks and firefighting that the DBA function had ended up morphing into and, instead, focus on higher-value tasks.

Tasks such as business needs analysis, data type and volume analysis, systems design and architecture, configuration optimization, performance management, business continuity planning and testing, workload analysis and segregation, application/database interplay design, security and compliance audits, and what-if analyses are creative things that are better done by these thinking individuals that have the capability to absorb the institutional knowledge in their environments and use it to build better databases.

The lower-level functions could then be outsourced or, better still, automated so the business is not impacted in the short term by this role change.

Sometimes I see an even worse problem. DBA managers make the DBAs take on system, network, or application administrative tasks or even function as developers. This only compounds the problem such that instead of just dealing with fires on the database front, they now have to choose which fire is the largest and, in the process, go even farther away from any proactive environment optimization.



Venkat Devraj

DBAs are busy fighting fires. In the end, this is not a DBA problem—it's a managerial problem.

Many of the people subscribing to this ineffective structure have since been rudely awakened when the new economy bubble burst. IT fell off its pedestal and came under intense scrutiny. Because the bar was low, the babysitting jobs started going offshore, where any trained engineer with a couple of years' experience could take over the job that our DBA-1 had been doing for the last decade—without a noticeable impact to quality. There are companies, both small and large, that have recognized this and have taken the right measures to optimize their structure and job functions; however, they are exceptions to the norm.

Let's talk about your book Oracle 24x7 Tips and Techniques. What led you to write it?

I found myself tasked multiple times with building a highly available database environment. I stumbled my way through this process many times, until I got pretty good at it. I learned that building a highly available database environment is not just about deploying a single magic bullet such as Parallel Server/RAC or Advanced Replication. It is a series of processes and configuration techniques that you need to embed in your approach.

While many portions of that book continue to remain relevant even today, what really excites me today is the ability to use features built within the DBMS kernel, well-written DBA-1 scripts and techniques and advanced analytics to take database administration to the next level—a more creative manner in which DBAs can get engaged to add more value to the business.

My next book discusses the momentum that we are gaining in this new area and what DBAs can do differently in their day-to-day jobs to leverage that momentum more effectively, rather than continue fire-fighting and watch their jobs being offshored, often to less experienced, but equally qualified individuals overseas. They need to creatively raise the bar by leveraging newer technology that is waiting in the wings to be adopted to enhance individual job satisfaction, as well as value to the business. My new book attempts to explain how to get from DBA version 1 (DBA-1) to DBA version 2 (DBA-2).

Is 24x7 really a myth?

24x7 is not about throwing together a system and then doing maintenance on it to keep it running forever. On the contrary, achieving 24x7 takes a good amount of prep time to understand and quantify business requirements and expected service levels, define standards that are appropriate for that environment, build a business case to identify the right budget based on such requirements versus the cost of downtime, assembling the right software and hardware components, then configuring them correctly to meet the business needs, and then testing the heck out of them to ensure they meet expected standards. 24x7 is largely established even before a system goes live. Getting a sub-optimally constructed live system to be 24x7 is a difficult endeavor and often results in failure and job loss.

Isn't 24x7 very expensive and out of the reach of most organizations?

I would argue to the contrary. There are many simple things that DBAs can do and attain much higher availability, performance, and scalability than they thought possible—if only those DBAs had the time to take a step back and think about it. Unfortunately, they are too busy dealing with the mundane. So 24x7 often seems daunting.

The lack of standard operating procedures in database administration astounds me.

At the end of the day, “very expensive” is a subjective term. If you are a financial clearinghouse and you stand to lose a couple of million dollars for every hour of database

downtime, would you think spending \$3M in robust infrastructure is “too expensive”? Anyway, expensive synchronous geo-mirroring is no longer the only option to get to 24x7. With some of Oracle's (and DB2 and SQL Server's) newer capabilities, 24x7 is within the reach of most companies—provided they have a legitimate business need to attain 24x7 and are willing to invest the time and resources to do it right the first time.

Isn't human error the major cause of outages? What advice do you have in this regard?

Yes, unfortunately in my limited experience, human errors do seem to account for the majority of outages. The lack of standard operating procedures in database administration astounds me. What astounds me even more is that people are content to keep DBA work shrouded as a black art. But business just cannot continue to operate that way. Oracle Corp. had an interesting observation in one of their recent ads, where they said a huge percentage (80%, I think) of outages are caused by human errors. That's why we need to work harder and smarter to automate our day-to-day tasks. As DBAs, whether we know it or not, whether we accept it or not, the fact is, most of our day-to-day workload is mundane and is automatable. The sooner we embrace the facts and actually get these low-level tasks standardized and automated, the less human intervention we would have and the lesser the chances of human error. Anything that's repeatable or complex needs to be automated, so we can move on to projects that have been on the back burner.

Last year, you made a presentation on “autonomics” to the Denver chapter of the IEEE. It is also the topic of your next book. Would you briefly explain the subject to our readers?

Autonomics brings together an expert system with the operational procedures to interpret the current state of an environment and automate its care and feeding both proactively and reactively. At an abstract level, it is meant to simulate an experienced administrator capable of gathering and

analyzing facts and thinking on her feet.

Let's take some common examples to understand this better: when you get a cut and start bleeding, the blood begins to thicken and clot to avoid further loss of blood. When you haven't eaten in a while, your body goes into conservation mode. These are natural reactions to stimuli built inside every normal human body and different parts work in tandem to achieve the desired response. Autonomics attempts to replicate this model to proactively carry out tasks that nurture and nourish the system, as well as select the best way to respond to a given stimuli caused within the environment (inside or outside the database). The operational actions and reactions are hardwired in the DNA. A good autonomics system attempts to build a baseline of environmental patterns by collecting key performance indicators and understanding acceptable deviations. This allows it to identify real problem patterns as significant deviations from the baseline emerge.

What support does Oracle have for autonomics, if any? What about other database engines such as DB2/UDB and SQL Server?

As far as I can see, newer versions of Oracle including 10g are getting better and better at one type of automation—self-repair—and so are DB2 UDB v8+ and SQL Server 2005. Such self-repair automation also allows the DBMS to repair itself when it runs into certain types of pre-

The thing that really hurts our company is employee turnover in India.

determined problems. They are also reducing the need for maintenance and downtime and, overall, are heading in the right direction.

For a long time, Larry Ellison has been predicting that database administrators will become extinct. Will autonomics hasten that day?

Yes and no. DBA work in the U.S. as we know it today (i.e., the DBA-1 work) is sure to be offshored more and more in the short run and will taper off in the mid to long run because autonomics will replace it. It's a natural evolution.

However, that will still not eliminate the need for a different breed of DBAs—the database architect (DBA-2)—because we still need people to use the newer database technologies and features to shorten deployment cycles and enhance service levels. That kind of creative DBA work is not likely to go away in the foreseeable future. Also, it will be a long time before the need for development DBAs and to a lesser extent, application DBAs, goes away.

What are some of your pet peeves about Oracle products? What new features would you like to see in Oracle?

Let me start out with a positive comment. In many ways, Oracle still has a much better product than many others out there. For instance, I haven't seen any other database provide the kind of granular performance metrics that Oracle's instrumentation layer provides—it pervades the database and goes deep into the network, hardware, storage sub-system, and application layers. It makes advanced troubleshooting so much easier.

My biggest peeve about it is that Oracle products tend to be much more bulky (in terms of resource consumption) and complex than required.

Database administration jobs are being sent to India (or Romania or the Philippines) all the time. Is the trend increasing or moderating and even declining?

In the short term, I expect the trend to accelerate more until a good chunk of DBA-1 work is offshored. Corresponding short-term challenges such as security concerns and false patriotism will evaporate as DBA outsourcing further matures. But as salaries and employee turnover rates continue to go up in India, the window of opportunity will become larger for other places like China, Romania, and the Philippines. If they play their cards right and sufficiently polish up their English language skills, these places will be in a good position to replace India as the offshoring destination of choice. However, real business issues such as lack of standard operating procedures resulting in inconsistent quality of work, problems with communications and cultural differences, different work ethics and customer expectations, and eventually rising salaries in China, Romania, etc., will lower the offshoring curve again. Autonomics will begin to bridge the gap at lower costs and better quality.

Meanwhile, human DBAs everywhere will continue to innovate and climb upstream, they do not have a choice.

When you are ready to choose who to partner with for your Consulting and Support Services needs...

We have the solution for you!

IT Convergence
EXPERIENCE THE POWER OF IT SOLUTIONS

Visit our web site at: www.itconvergence.com

You should be ensuring that each task worked on by a DBA has a corresponding ticket—no exceptions!

Corporations will always eventually do what's in the best interest of the shareholders. We need to embrace technology to continue adding higher value. As long as we do that, we will always have a job.

What is your experience with the quality of offshore personnel?

Overall, offshore personnel in the DBA area tend to be not as experienced as the quality of talent we find in the U.S. and require more training. But that is understandable because countries like India have been catapulted from an environment that had very little IT to the forefront of being the IT custodians for some of the world's largest organizations. Companies that have successfully utilized offshore resources have invested significantly in training, building standards, and improving communications. The salary difference is still enormous and the time differences make it ideal to leverage offshore locations for follow-the-sun support models. However, the thing that really hurts our company is employee turnover in India. Sometimes this starts even before people start on the job—they accept an offer, request a start date that is a month away (in India, people often are required to give a month's notice) and then, they don't show up on the day they are supposed to—no calls or explanations come forth!

India especially is going through a boom that we went through in the late '90s where a DBA could walk across the street and get a 30–40% raise. Once overseas HR policies strengthen and the demand settles and salaries level out, I expect this will be less of a problem in the next 3–4 years. Either that, or increasing levels of individual greed will cause India to lose its advantage to other countries, like China and Romania, that are further behind as far as IT salaries go.

What career advice do you have for database administrators and wannabe database administrators in this country? (Should we all switch to bee-keeping and other jobs that cannot be outsourced to cheap offshore locations?)

Not at all. On the contrary, I would say there is a shortage of good DBAs in the U.S. One of the reasons we use offshore resources is because we can't find enough good people here. More people should get into DBA work with both guns firing! Most importantly though, they need to be innovative and business-savvy. The DBAs here often have the luxury of being on-site with the business users. Spend time with them to understand how you can use your knowledge of the database to better serve their current and future needs.

Focus more on being database architects rather than database administrators. Be willing to embrace newer technology and automation. Be the technology champion in your organization by segregating workload into mundane and non-mundane. Don't be worried about losing

your job if the mundane work gets outsourced to firms that specialize in that and can bring economies of scale to bear. Instead, figure out how not to drop the ball with the non-mundane work. Read and understand the new features that your DBMS product gives you; play with those features in a lab or development environment, and work on projects that utilize those features to enhance your environment.

There is a shortage of good DBAs in the U.S.

Go beyond the command line and the GUI to understand the concept behind the technology. Don't worry that if you don't remember the commands, they will become antiquated. On the contrary, with your institutional knowledge, proximity to your users, deeper product knowledge, willingness to embrace outsourcing and automation, and your newly nurtured architectural capabilities, you can add more value to your company than ever before.

Finally, what kind of advice would you have for managers in our field?

Beyond asking those puzzling interview questions to screen your new hires, you should be doing your bit to make day-to-day DBA work a science in your environment. You should be ensuring that each task worked on by a DBA has a corresponding ticket—no exceptions! You should be analyzing the tickets on a weekly basis. The tickets should describe the exact problem worked on (e.g., it should read “rebuilt index NDX_tblname_02” rather than “did maintenance on the DB”) and the amount of time the person worked on it. If any ticket pertains to a recurring problem, you should expect your seniormost DBA to build a standard operating procedure for it that everyone can follow to produce consistent results when dealing with that issue. That procedure will also be the blueprint for any downstream automation.

You should attempt to ensure that once such standard operating procedures and automation routines are built, they are being used by every DBA in the team. You should attempt to ensure that workload is distributed evenly and average resolution time is brought down. When you do quarterly comparisons, you should see a reduction in time for tasks that took up the bulk of DBA time the previous quarter. Periodically, you should have a trustworthy consultant come in to do a “discovery” to ensure you are not missing any opportunities for further optimization. If you run a shop based on these ground rules, you can rest assured that your team members will have high morale, you can get a lot more done, and your senior management will value you—without having to worry about the optimal database-to-DBA ratio. Head count is less relevant; process is the king! ▲

Transportable Table Partitions

by Caleb Small

Introduction

Transportable Tablespaces are the most efficient way to move large amounts of data from one Oracle database to another. When combined with a properly designed table-partitioning scheme, this feature provides an ideal method of refreshing a data warehouse or archiving transactional data with minimum service disruptions.

To review, the basic premise of Transportable Tablespace is copying the actual Oracle datafile or datafiles from one database to another, then exporting metadata from the source database and importing it into the target database, thus “plugging in” a tablespace. The metadata export includes descriptions of the tables and other objects contained in the tablespace set, but no actual data, making it much faster than a conventional export. The source database requires the Enterprise Edition license.

In this tutorial, we simulate a transactional banking system, which is required to keep three years of data online and store older data in another database. We’ll start by creating a transaction table, range-partitioned by the transaction date. There will be one partition for each year’s worth of business. Furthermore, we’ll store each partition in its own tablespace to facilitate easy transport when we’re ready to transport a partition. The data will be indexed to enforce the primary key constraint. For efficiency, the indexes will be LOCAL PARTITIONED, meaning that each table partition will have its own corresponding index partition in its own tablespace. Corresponding table and index partitions can then be treated as one unit. To make the tutorial more interesting, the source database will be on the Linux platform, the target database will be on the Windows platform, and both will use Oracle 10g.

PART 1—Design and create the source table

Start by creating five tablespaces for five years’ worth of data, and five more for the indexes. Note the small initial size and the use of AUTOEXTEND with MAXSIZE. We don’t want to transport files that are larger than necessary.

```
CREATE TABLESPACE ttsdat1 DATAFILE '/u01/app/oracle/oratrans/ttsdat1.dbf'
  SIZE 1M AUTOEXTEND ON MAXSIZE 50M;

CREATE TABLESPACE ttsidx1 DATAFILE '/u01/app/oracle/oratrans/ttsidx1.dbf'
```

```
  SIZE 1M AUTOEXTEND ON MAXSIZE 50M;
...
SELECT tablespace_name, status, plugged_in FROM
dba_tablespaces;
save ts
```

Now, create the range-partitioned table using the transaction date as the partitioning key. Storage parameters, such as the tablespace name, can be different for each partition.

```
CREATE TABLE txns (
  trans_id NUMBER(12),
  trans_dt DATE,
  from_acct CHAR(10),
  to_acct CHAR(10),
  amount NUMBER(12,2)
  PARTITION BY RANGE (trans_dt)
    ( PARTITION fy2001 VALUES LESS THAN (to_date
('2002-01-01','yyyy-mm-dd') )
      TABLESPACE ttsdat1,
      PARTITION fy2002 VALUES LESS THAN (to_date
('2003-01-01','yyyy-mm-dd') )
      TABLESPACE ttsdat2,
      PARTITION fy2003 VALUES LESS THAN (to_date
('2004-01-01','yyyy-mm-dd') )
      TABLESPACE ttsdat3,
      PARTITION fy2004 VALUES LESS THAN (to_date
('2005-01-01','yyyy-mm-dd') )
      TABLESPACE ttsdat4,
      PARTITION fy2005 VALUES LESS THAN (to_date
('2006-01-01','yyyy-mm-dd') )
      TABLESPACE ttsdat5 );
```

Create a unique index to enforce the primary-key constraint. For manageability, we should use a LOCAL PARTITIONED index so that removing or adding a partition will not require an index rebuild in either source or target databases. However, this requires that the partitioning key be included in the index, which incurs additional storage and processing overhead. Again, each index partition has its own storage parameters.

```
CREATE UNIQUE INDEX txns_pk_idx ON txns(trans_id,trans_dt) LOCAL
  ( PARTITION fy2001 TABLESPACE ttsidx1,
  PARTITION fy2002 TABLESPACE ttsidx2,
  PARTITION fy2003 TABLESPACE ttsidx3,
  PARTITION fy2004 TABLESPACE ttsidx4,
  PARTITION fy2005 TABLESPACE ttsidx5 );
```


Once the index is created, we add the primary key constraint to the whole table. Remember that this will also add a NOT NULL constraint to the two indexed columns.

```
ALTER TABLE txns ADD CONSTRAINT txns_pk PRIMARY
KEY (trans_id, trans_dt)
  USING INDEX LOCAL
    ( PARTITION fy2001,
      PARTITION fy2002,
      PARTITION fy2003,
      PARTITION fy2004,
      PARTITION fy2005 );
descr txns
```

Now it's time to create some test data. The following procedure will create a bunch of rows that look like bank transaction data, and will spread it evenly over the last four years starting from today. Select a few rows from the table to convince yourself.

```
CREATE SEQUENCE trans_id_seq;
BEGIN
FOR i IN 1..25000 LOOP
  INSERT INTO txns SELECT
    trans_id_seq.nextval,
    SYSDATE-ABS(dbms_random.random)/
power(2,31)*365*4,
SUBSTR(TO_CHAR(dbms_random.random,'000000000000'),-10,10),
SUBSTR(TO_CHAR(dbms_random.random,'000000000000'),-10,10),
TO_CHAR(ABS(dbms_random.random)/
100000,'99999999.00') FROM dual;
  COMMIT;
END LOOP;
END;

SELECT * FROM txns WHERE rownum < 25;
SELECT trunc(trans_dt,'YEAR'), count(*)
FROM txns GROUP BY trunc(trans_dt,'YEAR');
```

Check some data dictionary views to find out what we've got. Save these queries for later use as we make changes to the table.

```
ANALYZE TABLE txns COMPUTE STATISTICS;
ANALYZE INDEX txns_pk_idx COMPUTE STATISTICS;

-- Number of rows in each table partition
SELECT partition_name, num_rows FROM user_tab_partitions;
save tabpartrows

-- Number of rows in each index partition
SELECT partition_name, num_rows, status FROM user_ind_partitions;
save indpartrows

-- Number of rows in the entire table
SELECT table_name, num_rows FROM user_tables;
save tabrows

-- Size of each segment
SELECT segment_type, segment_name, partition_name, tablespace_name, blocks, extents
FROM user_segments ORDER BY segment_Type, segment_name, partition_name;
save segsize
```

PART 2—Unplug the oldest partition

Our “production” table is complete and contains at least one partition that's old enough to archive. The first thing to do is to create a discrete table that is identical to the partition we want to remove. It must have the same columns, data types, constraints, and indexes. It must also exist in the same tablespace as the partition. The index must also exist in the corresponding index tablespace.

```
CREATE TABLE txns_fy2001 TABLESPACE ttsdat1 AS
SELECT * FROM txns WHERE 1=0;

ALTER TABLE txns_fy2001 ADD CONSTRAINT txns_fy2001_pk PRIMARY KEY (trans_id, trans_dt)
  USING INDEX TABLESPACE ttsidx1;
```

The new table contains no rows, and the index is empty. When we EXCHANGE a partition with a discrete table, we literally exchange data segments between the two. That's why they must match in structure and location. No data is actually moved; only internal pointers are changed.

```
ALTER TABLE txns EXCHANGE PARTITION fy2001
WITH TABLE txns_fy2001 INCLUDING INDEXES;
```

Count the rows in the new table and in the partitioned table before and after the EXCHANGE to convince yourself that the data did actually “move.” Check the size of the data segments, and the status of the indexes.

```
SELECT COUNT(*) FROM txns PARTITION (fy2001);
SELECT COUNT(*) FROM txns_fy2001;
@segsize
```

We no longer need the old partition because it is now empty, so we can drop it. If your goal was to simply transport this partition without removing it, such as in a data warehouse scenario, you could exchange the table back into the partition after transporting the tablespace.

```
ALTER TABLE txns DROP PARTITION fy2001;
```

To get ready, make the tablespaces we are transporting (data and index) READ ONLY to ensure there are no active transactions, and that none are initiated while the data is being transported. The rest of the partitions are unaffected.

```
ALTER TABLESPACE ttsdat1 READ ONLY;
ALTER TABLESPACE ttsidx1 READ ONLY;
```

Use the DBMS_TTS utility to verify that the tablespace set is self-contained; that is, there are no objects within the set that reference any object outside of the set (foreign keys, constraint indexes, and so on). The tablespace set can be composed of one or more tablespaces that will be transported together.

```
EXECUTE sys.DBMS_TTS.TRANSPORT_SET_
CHECK('ttsdat1,ttsidx1', TRUE);
SELECT * FROM sys.TRANSPORT_SET_VIOLATIONS;
```

We're going to use the Oracle data pump to perform the export, so make sure a named directory exists within the database and that the appropriate users have the necessary permissions to use it. Unlike the old export and import utilities, the data pump cannot arbitrarily read or write to any directory.

```
CREATE OR REPLACE DIRECTORY trans_dir AS '/u01/app/oracle/oratrans';
GRANT READ, WRITE ON DIRECTORY trans_dir TO ...;
```

Perform the metadata export using the data pump. This will run quickly and consistently regardless of the partition size, as it contains data descriptions only, not actual data.

```
cd /u01/app/oracle/oratrans
expdp bank/bank@FRED \
  DUMPFILE=ttsfy1.dmp \
  DIRECTORY=trans_dir \
  TRANSPORT_TABLESPACES = ttsdat1,ttsidx1
ls -al
```

The second part of transportation involves copying the datafiles for the tablespaces we are moving. If there is more than one datafile per tablespace, remember to get them all. Remember to include all tablespaces that make up the set to be transported. If you are moving to a similar operating system platform, copying can be performed by simple operating system commands. If you are using ASM or if "endian conversion" is necessary, use RMAN to perform the copy (and parallelize it!). You can list supported plat-

forms and the corresponding Endian formats by querying V\$TRANSPORTABLE_PLATFORM.

```
rman target fred
CONVERT TABLESPACE ttsdat1, ttsidx1
  TO PLATFORM 'Microsoft Windows IA (32-bit)'
  FORMAT='$ORACLE_BASE/oratrans/%N_win.dbf';
exit
ls -al
```

After transport is complete, we can drop the source tablespaces. This is optional based on your business requirement. You could go back to the source database and simply put them back into READ WRITE mode and continue business as usual. Discrete tables can be easily EXCHANGED back into the original partition they came from.

```
DROP TABLESPACE ttsdat1 INCLUDING CONTENTS;
DROP TABLESPACE ttsidx1 INCLUDING CONTENTS;
```

In preparation for the next year's business data, we need to create new data and index partitions, in their own tablespaces as usual, to facilitate future transport.

```
CREATE TABLESPACE ttsdat6 DATAFILE '/u01/app/oracle/oratrans/ttsdat6.dbf'
  SIZE 1M AUTOEXTEND ON MAXSIZE 50M;
CREATE TABLESPACE ttsidx6 DATAFILE '/u01/app/oracle/oratrans/ttsidx6.dbf'
  SIZE 1M AUTOEXTEND ON MAXSIZE 50M;

ALTER TABLE txns ADD PARTITION fy2006
  VALUES LESS THAN (to_date('2007-01-01','yyyy-mm-dd'))
  TABLESPACE ttsdat6
```

Quovera

Oracle Consulting Solutions Specialists

With over 10 years of Oracle consulting excellence, Quovera has performed more than 300 engagements for over 200 clients. Quovera's specialized Oracle consulting services include:

- ◆ 11i E-Business Suite implementations and upgrades
- ◆ Full lifecycle custom and web development and data warehousing
- ◆ Business intelligence, reporting and business performance management

Quovera also provides expert mentoring and training on JDeveloper, Developer and Designer with its team of renowned Oracle Press Authors and Certified Masters.

Quovera brings the experience of its Oracle consulting staff to work with you to improve your Information Technology processes and achieve your business goals.

Quovera, Inc.

800 West El Camino Real, Suite 100
Mountain View, CA 94040
Tel. (650) 962-6319 · www.quovera.com



Pioneering the Field of Capacity Optimized Storage

The Data Domain DD200 Restorer:

A disk-based recovery storage appliance designed for Oracle database backups:

- ▶ Works with leading backup software vendors and RMAN
- ▶ Patented Capacity Optimized Storage technology that enables long term data protection at low cost
- ▶ Data Invulnerability Architecture ensures recoverability of backup data
- ▶ Rapid disaster recovery via cost effective site-to-site data replication



Data Domain®

Next Generation Data Protection

www.datadomain.com
1.877.622.2587

```
UPDATE INDEXES (txns_pk_idx (PARTITION fy2006
TABLESPACE ttsidx6));
@indpartrows
```

PART 3—Plug into the archive database

Copy the metadata export and the datafiles comprising the data and index tablespaces to the target system. Connect to the target system and verify that the database character sets match those of the source database. This is a very important restriction on Transportable Tablespaces!

```
CONNECT arch/arch@BING
SELECT * FROM v$nls_parameters WHERE
parameter IN ('NLS_CHARACTERSET', 'NLS_NCHAR_
CHARACTERSET');
```

Ensure that an Oracle directory with necessary permissions exists for the use of the data pump.

```
CREATE OR REPLACE DIRECTORY trans_dir AS 'd:
/oradata/oratrans';
GRANT READ, WRITE ON DIRECTORY trans_dir TO
...;
```

Import the metadata into the target system. Again this is fast because we only import data definitions; the data already exists in the named datafiles. Note the use of REMAP_SCHEMA when a different user will own the objects in the target database.

```
impdp arch/arch@BING parfile=ttsfy1.par
DUMPFIL=ttsfy1.dmp
DIRECTORY=trans_dir
REMAP_SCHEMA=bank:arch
TRANSPORT_DATAFILES='d:\oradata\oratrans\
TTSDAT1_WIN.dbf',
'd:\oradata\oratrans\TTSIDX1_WIN.dbf'
```

By default, the new tablespaces will be READ ONLY. In order to create the table partition and EXCHANGE it with the existing discrete table, we will have to make them READ WRITE.

```
ALTER TABLESPACE ttsdat1 READ WRITE;
ALTER TABLESPACE ttsidx1 READ WRITE;
```

Because this is the first time we are transporting a partition, we must create the partitioned table with its first partition. Remember, it must exactly match the source table and exist in the same tablespace.

```
CREATE TABLE arch
PARTITION BY RANGE (trans_dt)
( PARTITION fy2001 VALUES LESS THAN (to_date
('2002-01-01', 'yyyy-mm-dd') )
TABLESPACE ttsdat1 )
AS SELECT * FROM txns_fy2001 WHERE 1=0;
```

Again, the new partition must exactly match the discrete table, including indexes and constraints. Note the use of the tablespace clause to ensure that the index is located in the same tablespace.

```
CREATE UNIQUE INDEX arch_pk_idx ON arch(trans_
id,trans_dt) LOCAL
( PARTITION fy2001 TABLESPACE ttsidx1 );
ALTER TABLE arch ADD CONSTRAINT arch_pk PRIMARY
KEY (trans_id, trans_dt)
USING INDEX LOCAL ( PARTITION fy2001 );
descr arch
```

We now have a partitioned table (with one empty partition), and our discrete table with data that was transported. To complete the process, simply EXCHANGE data segments between the empty partition and the discrete table. Then, count the number of rows and check the size of data segments and the status of indexes.

```
ALTER TABLE arch EXCHANGE PARTITION fy2001
WITH TABLE txns_fy2001 INCLUDING INDEXES;
@segsiz
@indpartrows
```

When you are convinced that the exchange has occurred properly and the partitioned table is functional, drop the now-empty discrete table and you are done for this year.

```
DROP TABLE txns_fy2001 PURGE;
```

In subsequent years, transport of partitions would follow an almost identical process. The only difference is that the partitioned table already exists in the target database; therefore, we simply ADD another partition and EXCHANGE it with the discrete table, as in the following example.

```
ALTER TABLE arch ADD PARTITION fy2002
VALUES LESS THAN (to_date('2003-01-01', 'yyyy-mm-dd') )
TABLESPACE ttsdat2
UPDATE INDEXES (arch_pk_idx (PARTITION fy2002
TABLESPACE ttsidx2));
ALTER TABLE arch EXCHANGE PARTITION fy2002
WITH TABLE txns_fy2002 INCLUDING INDEXES;
DROP TABLE txns_fy2002 PURGE;
```

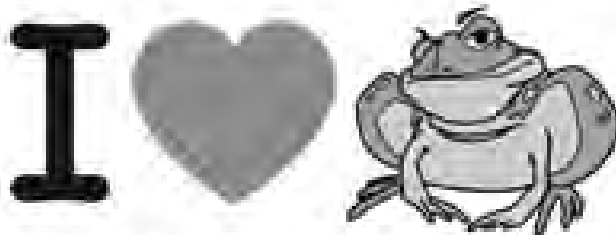
Summary

Both Transportable Tablespaces and Table Partitioning are complex features of the Oracle database. They can greatly enhance the performance and manageability of a production system. However, it is imperative to spend some time setting up a test bed and evaluating your specific situation. There are many other caveats and restrictions that you might encounter. This tutorial only discusses one simple scenario, and there is no replacement for reading and understanding the Oracle documentation.

More information on Transportable Tablespaces is available on my website at www.caleb.com/dba. ▲

Caleb Small is an independent consultant currently teaching evening courses on Oracle technology at Camosun College in Victoria and the University of Washington in Seattle. He can be contacted at caleb@caleb.com.

Copyright © 2005, Caleb.com



We Know you Love Toad

Now fall in love with our other Oracle solutions.

Whatever you need for Oracle — from development to database management — Quest Software has an industry-leading solution for you. Build better code-faster. Diagnose and resolve performance problems before they impact your end users. And administer change and ensure availability.

So if you love Toad, we know you will adore our other Oracle solutions too.

To learn more about all of our Oracle Solutions,
please visit www.quest.com/heart2

Application Management | Database Management | Infrastructure Management



Hello, World!— An Introduction to Python

by Catherine Devlin

*Hello, world, here's a song that we're singin',
c'mon get happy
A whole lotta lovin' is what we'll be bringin',
we'll make you happy . . .*

—The Partridge Family theme song



Catherine Devlin

A "Hello, World!" program is a simple program that displays the phrase "Hello, World!" It is used as a way to introduce a new programming language to programmers. The first "Hello World!" program was found in *The C Programming Language*, by Kernighan and Ritchie.

In this series of articles, expert programmers teach us new ways to access the Oracle database. The assignment each time is simply to select the phrase "Hello, World!" from the DUAL table into a program variable and then display the contents of the variable. Each expert will also take the opportunity to describe the program's environment and the mechanics of the Oracle connection.

Our first expert is Catherine Devlin, who describes an exotically named programming language called Python.

Introduction to Python

“A gile” or “dynamic” languages have received increasing attention in recent years for their ease of use and high productivity. These include Perl, PHP, Ruby, and my favorite by far, Python. Python is a high-level, object-oriented, open-source, all-purpose programming language. It is available for virtually all platforms, including Linux, Unix, Windows, and Mac OS X.

Python was introduced in 1991 by Guido van Rossum, who named it after the Monty Python television show. That playful spirit continues among Python users, who refer to Guido as the BDFL—the Benevolent Dictator For Life—of the community. Development of Python has proceeded vigorously yet carefully, producing a language that is extremely elegant and simple at its core yet provides powerful libraries and packages for virtually any purpose.

Python continues to grow in popularity; currently it is the #6 choice for SourceForge projects, and it is #8 in the TIOBE rankings. Python is completely free of charge and is supported in classic open-source fashion by its friendly user community. A variety of consultants are available to

support development efforts, and vendor-supported IDEs like WingIDE and Komodo are available in addition to the community-created ones.

Why Python? (Why Not PL/SQL?)

You'll find Python most useful when:

- ▶ You want to use object-oriented programming. PL/SQL does have some object-oriented capabilities, but the syntax is clumsy and has painful limitations. Python's object orientation is clean and powerful.
- ▶ You need to access data that is not contained in the database such as operating system files, operating system settings, Internet data, values in the Windows registry, and so on. Some of this data can be accessed in PL/SQL, but it is almost always easier in Python.

PL/SQL, on the other hand, will likely be more useful when:

- ▶ Your code consists primarily of SQL statements and relatively uncomplicated logic.
- ▶ The code needs to be invoked by a database trigger.

The Python Architecture

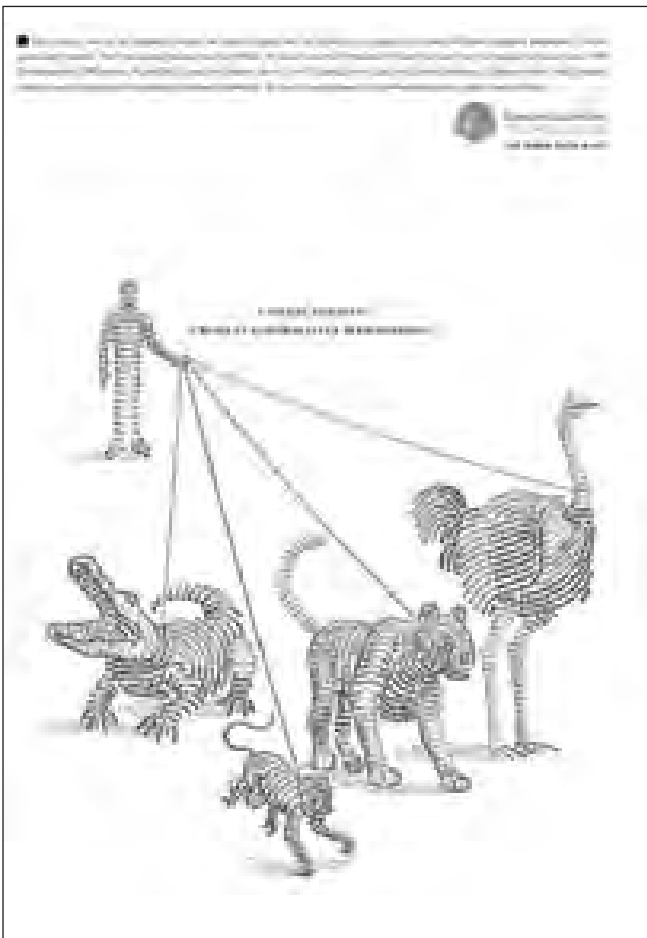
Python code is executed by an interpreter. Like PL/SQL, and unlike Java or the C language, Python requires no separate compile or link steps. A Python program file (e.g., “myProgramName.py”) is invoked simply by entering “python myProgramName.py” at the command prompt.

For access to Oracle, a special module not contained in Python’s standard library must be downloaded and installed. You have a few choices. ODBC modules are available for Python. In this case, you’ll need to install an ODBC driver for Oracle and create an ODBC “data source” on your client machine using operating system tools.

Alternatively, you can use a “DB-API 2.0” module to give Python direct access to the Oracle OCI. DB-API 2.0 is a document specifying uniform syntax and capabilities for third-party Python database access modules. Two of these, “DCOracle” and “cx_Oracle”, are available for Oracle; we’ll use cx_Oracle in our programming example, since it is particularly easy to install.

Getting Started with Python

If you use Linux or Unix, you probably already have Python; type “python” and see what happens. Otherwise, you can download it without charge from www.python.org. For Windows, download the MSI installer, which will install a precompiled binary. Of course, you may also download and compile the C language source code; in that case, refer to the Python website for instructions specific to your operating system.



To access Oracle, download the cx_Oracle module from www.computronix.com/utilities.shtml. Here you’ll find MSIs for Windows binaries, RPMs for Linux, and source code. To install from source, just run “python setup.py install” (these are the magic words to install almost any Python module). You’ll also need at least a basic Oracle client installed, so that cx_Oracle can use the Oracle Call Interface.

The First Program

Entering “python” at the command prompt will start Python in interactive mode. Here’s your first Python program:

```
>>> print "Hello, World!"
Hello, World!
```

This interactive mode is great for quick experimentation. To retrieve the phrase “Hello, World!” from the database, we’ll need to do a little more work, so let’s use a text editor to write a program called “helloWorld.py”. For clarity, we provide the whole script followed by line-by-line explanations.

```
import cx_Oracle
connectString='scott/tiger@orcl'
try:
    conn = cx_Oracle.Connection(connectString)
    curs = conn.cursor()
    bindVar = 'X'
    curs.execute("SELECT 'Hello, World!' FROM dual
WHERE dummy = :bv", bv=bindVar)
    line = curs.fetchone()
    result = line[0]
    print result
    curs.close()
    conn.close()
except (RuntimeError, DatabaseError),
errorDetails:
    print 'Problem accessing database %s: %s' %
(connectString, errorDetails)
```

1. Because cx_Oracle is not part of Python’s core, we use the “import” statement to make its components available to this script. Notice that we didn’t use a semicolon to end the command, because all Python commands end at the end of the line anyway. Python simply adheres to the human style of reading and writing instead of forcing you to use one convention for human eyes and a second convention for the computer. In any case, good programming style calls for one command per line.
2. Notice that we did not have to declare the type of the connectString variable before assigning a value to it. This feature is called “dynamic typing.” It’s the Python interpreter’s job, not yours, to figure out that connectString is a string variable.
3. Any errors that arise in the “try block” will be handled by the “except block” that follows it. This is similar to “BEGIN” and “EXCEPTION” in PL/SQL.
4. Note that this line is indented. This is how Python knows where your block begins and ends—the block

ends when you stop indenting text, without the need for an explicit “END” or curly brackets. This is another example of human-friendly Python syntax. This line creates an instance of the Connection object using the cx_Oracle module. “cx_Oracle.Connection(connectString)” invokes Connection’s “constructor method,” with connectString as the parameter. If object-oriented jargon makes you dizzy, all you need to understand is that this is how we establish a database connection!

5. Since “conn” is an object, various “methods”—functions or procedures associated with the object and acting directly on its data—are available for it. One of these methods, “cursor”, creates an SQL cursor.
6. This line should be self-explanatory for most readers!
7. Python recognizes either single or double quotes; a string that begins with a single quote must end with a single quote, and vice versa. This is handy for strings that themselves include quotes—simply use the other type of quote to enclose the string, and you won’t need to use escape characters for the quotes inside the string.
8. One the query has been executed by Oracle, “fetchone()” will bring back a row of data in the form of a list—Python’s version of an array.
9. We need the first (and only) value in the list. Notice that the elements of the list are numbered beginning with the number 0, not the number 1.
10. This is similar to DBMS_OUTPUT.PUT_LINE (without the limitations).
11. Explicitly closing the cursor is not strictly necessary—Python will clean up when execution terminates—but it would be important if this were part of a longer script.
12. Explicitly closing the connection is not strictly necessary either.
13. Like the EXCEPTION block of a PL/SQL program, the following block will be executed only if an error were encountered in the try block. This exception handler only traps errors of the two specified types—RuntimeError, a core Python exception type, and DatabaseError, which is defined by the cx_Oracle module. Specifics about the error that was encountered will be made available in errorDetails.
14. The “%s” placeholders are replaced with values supplied in the parenthesized comma-separated list.

Additional notes for the Oracle literate:

- Python is case-sensitive!
- In Python, “=” assigns, “==” compares (like “:=” and “=”, respectively, in PL/SQL.)
- In PL/SQL, if a function or procedure does not have

any parameters, you may omit the parentheses; i.e., “myOraFunc” may be used in place of “myOraFunc()”. In Python, you must always use the parentheses when invoking a function; “myPyFunc” is simply a “reference” to the function; you can do several wonderful things with it, but not invoke the function!

Compiling, Linking, and Executing

As we have previously mentioned, Python code is executed by an interpreter. Like PL/SQL, and unlike Java or C, Python requires no separate compile or link steps.

A Python program is invoked by entering “python myProgramName.py” at the command prompt, and OS settings can also be configured to invoke Python automatically whenever a “.py” file is selected.

Further Reading

I hope you’ve become intrigued by Python’s ease and elegance. What you haven’t seen yet is Python’s power—the huge range of programming tasks it can help you with. If you’re ready to read more, here are some places to start.

- www.python.org/—Official Python site. Downloads, tutorials, etc.
- www.python.org/topics/database/—Central point for Python database work, with links for database modules.
- www.ioung.org/Python.pdf—Like this article, but longer.
- www.pythonware.com/daily/—Aggregated Python blogs. New Python goodies appear constantly, and blogs can help you keep up with the latest Python news.

Object-oriented devotees often want object-relational mapping tools (like Oracle’s TopLink), which hide SQL from the programmer and expose the database in object form. Some Oracle-compatible O/R tools for Python can be found at the following addresses:

- www.modeling.sourceforge.net/
- www.opensource.theopalgrou.com/
- www.soiland.no/software/forgetsq
- www.skunkweb.sourceforge.net/pydo.html
- [www.livinglogic.de/Python/orasql/ ▲](http://www.livinglogic.de/Python/orasql/)

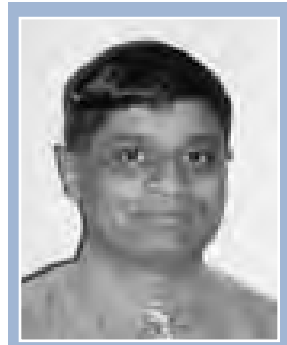
Catherine Devlin is an Oracle Certified Professional with six years of experience as an all-purpose DBA, PL/SQL developer, distributed systems architect, and web application developer for small-scale Oracle OLTP systems. She got her start at URS Corp. in Sacramento, and today works for IntelliTech Systems in Dayton, Ohio. She blogs at catherinedevlin.blogspot.com/ and can be reached at catherine.devlin@gmail.com.

It Was a Dark and Stormy Night...

It was a dark and stormy night; the rain fell in torrents—except at occasional intervals, when it was checked by a violent gust of wind which swept up the streets (for it is in London that our scene lies), rattling along the housetops, and fiercely agitating the scanty flame of the lamps that struggled against the darkness.

—The first sentence of the novel *Paul Clifford* by the nineteenth-century author Edward George Bulwer-Lytton

A Story about Deadlocks



Iggy Fernandez

Introduction

Edward George Bulwer-Lytton was a nineteenth-century English politician and author who coined the phrase, “The pen is mightier than the sword.” He is also famous for the suggestive opening phrase “It was a dark and stormy night” that appears in so many Peanuts comic strips by Charles Schulz.¹

In this essay, we first tell a silly story about deadlocks in the flowery style of Bulwer-Lytton. The story illustrates that there are many different opinions about how to handle deadlocks. We then take a more serious look at the theory behind deadlocks.

A Silly Story about Deadlocks (in the Flowery Style of Bulwer-Lytton)

It was a dark and stormy night; a coven of database administrators was gathered at the local watering hole, trading “big fish” stories (interlaced, like a concoction of couscous sprinkled with parsley, with the occasional “How I Trashed the Database” story) and quaffing the generous libations poured by the master of the house.

Suddenly the frantic shriek of a cell phone filled the air with a discordant cacophony that seemed to rend the very heavens.

The youngest database administrator pulled out his cell phone and read the message faster than any gunslinger from the Wild West ever drew his revolver and gunned down a jackdaw sitting atop a pine. “Why are they calling me about deadlocks?” he screamed at the dark and stormy heavens.

“Don’t sit there whining while the entire database is deadlocked!” scolded a grizzled old graybeard. “Watch that database like a hawk! When I was a young feller, I single-handedly managed many thousands of databases! You live in a country club by comparison!”

He tipped his head back and poured the rest of his beer down his throat. Then his face fell to the table and he began to snore.

“True! True! Well said!” chimed the cheery database administrator on his left. “At my previous company, we treated deadlocks like vermin and immediately killed the

¹ San Jose State University conducts an annual “bad writing competition,” in which contestants have to “compose the opening sentence to the worst of all possible novels.” The entire “Lyttony” of winners since the competition was instituted in 1982 can be found at www.bulwer-lytton.com/lyttony.htm.

transaction that was causing the deadlock!”

“If deadlocks occur frequently, there is no need for cell phone messages,” said the serious database administrator on his right. “It would be sufficient to send out an email message whenever a deadlock occurs.”

“Bah! Humbug!” said the youngest database administrator. Then, remembering his manners and his low ranking, he hastened to add, “God bless us, every one!”

The hour was getting rather late, so they shook the grizzled old graybeard awake and went their different ways, the youngest database administrator still grumbling under his breath.

By this time, the night was no longer dark or stormy, and a full moon filled the heavens with a softly glowing silver light that helped them fit their keys into the keyholes of their cars.

We trust that they all got home safely. God bless them, every one!

Seriously Speaking

There are many different opinions about how to handle deadlocks. To add to the confusion, Oracle reports the error in the “alert log” and creates a “trace file,” a treatment usually reserved for more serious errors such as “Cannot Extend Segment.” Here is the classic deadlock scenario:

1. At Time T1, some transaction (let’s call it Transaction A) acquires an exclusive lock on some object (let’s call it Object O1) by changing it in some way or by reserving it for a future update (“SELECT FOR UPDATE”).
2. At Time T2, another transaction (let’s call it Transaction B) acquires an exclusive lock on some other object (let’s call it Object O2) by changing it in some way or by reserving it for a future update (“SELECT FOR UPDATE”).
3. At Time T3, Transaction A then tries to acquire an exclusive lock on Object O2 and is blocked by Transaction B. Transaction A now has to wait until Transaction B completes its transaction (“COMMIT”) or abandons it (“ROLLBACK”).
4. At Time T3, Transaction B then tries to acquire an exclusive lock on Object O1 and is blocked by Transaction A. Transaction B now has to wait until Transaction A completes its transaction (“COMMIT”) or abandons it (“ROLLBACK”).

Now, Transaction A and Transaction B are each waiting for the other to complete or abandon its transaction and are said to be “deadlocked.” Fortunately, Oracle automatically detects this scenario, picks one of the transactions as a victim and automatically rolls back its changes, thus allowing the more fortunate transaction to proceed.

It should be emphasized that Oracle has already resolved the deadlock, and there is nothing the database administrator needs to do after Oracle reports the problem. *Every other database engine, including SQL Server and DB2/UDB, also exhibits this automatic behavior.* This is unlike a plain-vanilla blocking lock that has created a logjam in the database and is forcing other transactions to queue behind it—a scenario that might require the intervention of a database administrator. Plain-vanilla locks are very common in database engines that force readers to block writers, but they are much rarer in the Oracle world because of “Read Consistency.” Be that as it may, our article only concerns itself with “deadlocks,” not “plain-vanilla blocking locks.”

What does a transaction have to do if it is picked as a victim in a deadlock scenario? The transaction will receive the error “ORA-00060 deadlock detected while waiting for resource,” indicating that it was picked as a victim and that *all of its work has been rolled back.* The transaction is free to try again, and there is a more-than-considerable chance that it will succeed the second time around.

It is easy to conceive of a deadlock involving three or more transactions (e.g., Transaction A is waiting on Transaction B, which in turn is waiting on Transaction C, which in turn is waiting for Transaction A). Once again, Oracle can detect the problem and will pick one transaction as a victim. In mathematical language, a deadlock is a “cycle” in a “directed graph.”

Deadlock is a fairly common error in other database engines such as SQL Server and DB2/UDB for several reasons:

- ▶ Other database engines use in-memory “lock managers”; i.e., locks are maintained in memory instead of in the affected block (as in the case of Oracle).
- ▶ Other database engines use page-level locking; i.e., in order to conserve memory space in the lock manager, a whole page of records is automatically locked when a transaction intends to lock only a single record within the page.
- ▶ Other database engines use “lock escalation”; i.e., the lock manager promotes a transaction’s row or page locks to a table-level lock in order to conserve memory space if the transaction exceeds a predefined quota of locks.

In-Memory Lock Managers and Page-Level Locking

All database engines except Oracle enforce row locks using an in-memory lock manager. (Oracle, by contrast, enforces a row lock by updating the data page itself.) In order to conserve memory space, older versions of those database engines actually locked the entire data page, even if the transaction needed to lock only one row. This predisposes applications to deadlocks as illustrated by the following scenario:

1. At Time T1, Transaction A requests an exclusive lock

on a single row but an entire data page is locked instead, potentially locking many more rows than are actually required.

2. At Time T2, Transaction B requests an exclusive lock on a different row on a different page and, as in the case of Transaction A, receives a lock on the entire data page in question, potentially locking many more rows than are actually required.
3. At Time T3, Transaction A then requests an exclusive lock on a row in the data page locked by Transaction B.
4. At Time T4, Transaction B then requests an exclusive lock on a row in the data page locked by Transaction A.

There is one scenario in which Oracle locks more rows than are required by a transaction. This occurs in the case of bit-mapped indexes because each entry in a bit-mapped index represents *all* rows that have the same value of the indexed column. Beware of bit-mapped indexes if the table in question is regularly updated.

In-Memory Lock Managers and Lock Escalations

The in-memory lock managers used by database engines other than Oracle are constrained by memory space. Once a transaction exceeds a pre-configured number of row or page locks on a single table, a *table lock* is acquired and the row or page locks are released. This is referred to as a “lock escalation.” This predisposes applications to deadlocks as illustrated by the following scenario:

1. Transaction A requests a whole bunch of exclusive row locks on a single table.
2. In parallel with Transaction A, Transaction B also requests a whole bunch of exclusive row locks on the same table.
3. Transaction A then requests one lock too many on the same table. The lock manager then attempts to acquire a *table lock* on behalf of Transaction A and is blocked by Transaction B.
4. Similarly, Transaction B requests one lock too many on the same table. The lock manager then attempts to acquire a *table lock* on behalf of Transaction B and is blocked by Transaction A.

And So We End the Ditty

Deadlock is very rare in the Oracle architecture (except perhaps in the case of bit-mapped indexes). Oracle therefore sees fit to treat it specially and record it in the database alert log. However, Oracle automatically resolves deadlocks by picking a victim and rolling back its changes. Therefore, there is nothing left for the database administrator to do when the error is recorded in the alert log. The transaction is free to try again if appropriate. If frequent errors (a subjective term) are reported in the alert log, long-term fixes such as changes to the application design or physical database design may be necessary. ▲

Iggy Fernandez is a Senior Database Administrator and possesses certifications for Oracle 8i, Oracle 9i, and Oracle 10g. Feedback and comments on this article may be sent to iggy_fernandez@hotmail.com.

Oracle Insights: Tales of the Oak Table

Review by Brian Hitchcock

Our regular columnist Brian Hitchcock takes an irreverent and thought-provoking look at a fascinating new book, Oracle Insights: Tales of the Oak Table, and highlights some of the equally irreverent and thought-provoking comments made in the book such as, "We have many world-renowned experts whose tuning methods lead us nowhere near the true problem" and "Sometimes the cleverest people made the most fundamental errors."

Title: Oracle Insights: Tales of the Oak Table

By: Mogens Nørgaard and 10 others

Published by: Apress, 2004

Summary

Overall review: Some of the material is worthwhile, but overall, I think your time and money could be better spent on other books.

Target audience: Oracle DBAs.

Would you recommend to others? Again, there is some good material, but not enough to pay for.

Who will get the most from this book? Readers who want to spend the time to learn more of the lore of Oracle.

Is this book platform specific? No.

Why did I obtain this book? I didn't; I was asked to review it and it was given to me.

Opening Remarks

I think it is important that you understand where I'm coming from before you read this review. While I don't own any oak furniture (all my furniture comes from Wal-Mart: it works, and it's a good value), these observations represent my perspective:

- You are the only one who knows your systems.
- You should use anything that works, including ratios, SQL trace data, OS level statistics, and whatever tools and techniques get you and your customers to a good outcome.
- Talking to people is the most important technique you have. Communicating with the people (and they are

people) who support all the other parts of the overall application system does more to solve complex performance issues than any single expert can do.

- Proprietary has seen its best days. Proprietary Unix and proprietary RDBMS are on the way out. They won't be gone tomorrow, but the various open-source operating systems and RDBMS products are growing. The database itself is moving away from the center of the action to a supporting role. I think you should use the time you have to learn about open-source RDBMS and the many other components of modern application systems. It is these other components that are coming up more and more in performance engagements.

Overall Review

This book is a collection of material from its various authors. As such, the content and quality of the chapters vary widely. While there are pieces of information that are worth knowing, much of the material is widely known, commonsense, or both. There is little here that is new or unique. I don't think you will learn very much that you can immediately apply to your current projects.

This book reflects a lack of editing. You don't know how good your editor is until you see how other publications handle the task. There is no consistency between the chapters. One will be very high level, and the next will have very specific technical detail, with nothing tying them together.

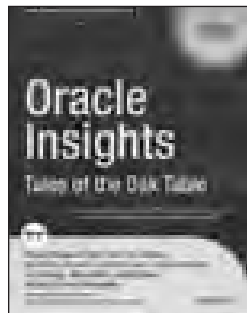
Far too much space is taken up with the authors telling us how impressed they are with each other. Reading 32 pages of this is tedious. And do you really need to pay to read about these experts' bathroom and sleep habits?

Some of the material is very old. There are many contradictions from chapter to chapter. One expert explains just how bad something is, and in the next chapter another expert explains how that same thing is a valuable solution. Some of the material is repetitive. How many times do we need to be told that using "ratios" for performance tuning is a bad idea?

Given the vast differences in quality, tone, and theme of each chapter, I decided to review each chapter separately.

The chapters are:

1. A Brief History of Oracle—Dave Ensor
2. You Probably Don't Tune Right—Mogens Nørgaard
3. Waste Not, Want Not—Connor McDonald
4. Why I Invented YAPP—Anjo Kolk



5. Extended SQL Trace Data—Cary Millsap
 6. Direct Memory Access—Kyle Hailey
 7. Compulsive Tuning Disorder—
Gaja Krishna Vaidyanatha
 8. New Releases and Big Projects—James Morle
 9. Testing and Risk Management—David Ruthven
 10. Design Disasters—Jonathan Lewis
 11. Bad CaRMa—Tim Gorman
- Appendix: Join the BAARF Party (or Not)—
James Morle and Mogens Nørgaard

Chapter 1: A Brief History of Oracle—Dave Ensor

I enjoyed the brief history, and it helps explain how some “features” came to be. It isn’t always great science. I also recommend the section on Oracle Parallel Server (OPS) and Real Application Clusters (RAC). The author gives a review of OPS and RAC and offers the opinion that overall it isn’t much better than a well-configured SMP machine. Once you really grasp the point that you need to “partition” your database to get good scalability, you realize that the less you really use RAC, the better it performs. I also found it interesting that the author refers to Flashback features as a “complete farce,” and the point is made that most of the Flashback features aren’t really useful for most systems.

Chapter 2: You Probably Don’t Tune Right—Mogens Nørgaard

This chapter has the best insights of the whole book, but it really makes the following chapters seem less worthwhile. The author explains how Oracle is becoming a commodity and that proprietary Unix is dying. These observations need to be heard and the implications considered. However, this leaves an important question unanswered. Having stated that Oracle is becoming a commodity and proprietary Unix is dying, the author doesn’t answer the logical next question, namely, is there any reason proprietary RDBMS, such as Oracle, won’t die as well? The implications of the end of the era of proprietary OS, RDBMS, and other pieces of our application environments need to be understood. If proprietary Unix is being replaced by Linux, why not include some insights on how Oracle is being replaced by open-source RDBMS?

The author also explains that you need fully instrumented code for all pieces of an application, and that without this you can’t know what is happening to your app. This is a very important and reasonable observation. However, much of the rest of the book ignores this and covers nothing but the Oracle database. How many current business systems consist of only an Oracle database? I would like to know what insights the author could offer us about application systems that are more complex.

Chapter 3: Waste Not, Want Not—Connor McDonald

This chapter discusses very detailed solutions to very specific performance issues. While the author’s description of these complicated tuning solutions is thorough, what isn’t discussed is how this system performed over the long term. Once a system is highly tuned, it often needs even more tuning in the future, making complicated tuning solutions

harder to maintain. Also, in Chapter 7 we are told that we shouldn’t tune too much. The author doesn’t discuss the issue of how much tuning is too much. The author suggests Flashback as a cure to one of his problems, but he doesn’t address the issue that the author of Chapter 1 refers to Flashback as a “complete farce.”

As with other chapters, there are interesting things here. The discussion of parsing issues and the Hidden DUAL

You need fully instrumented code for all pieces of an application, and that without this you can’t know what is happening to your app.

Access interested me because it was something I had never thought of. Some of the material here covers Forms 6 on Oracle 8.1.6. Most readers would probably prefer hearing about the author’s experience with something more current.

Chapter 4: Why I Invented YAPP—Anjo Kolk

The most interesting thing in this chapter is the discussion of just how disorganized the implementation of the Wait Interface really was. It is popular to view the Wait Interface as something that was fully designed and covers everything. It was a one-off solution to a pressing problem and, to quote the author, “it wasn’t really planned” and “it was kind of sneaked in at the last minute.” Keep this in mind when you hear how the Wait Interface can solve all performance issues.

It is also pointed out that the Oracle software often doesn’t work the way the manuals (and former Oracle employees) say it does. Keep this in mind when things just don’t work the way you were told they would in the Oracle training class.

The author’s main story covers the many problems he solved for a system using OPS, but the value of using OPS is not discussed. Didn’t we read in Chapter 1 that OPS and RAC really aren’t that useful despite all the marketing?

As with Chapter 2, the issue is brought out that you must have performance data for all the components of your application system before you can make any progress, but we are not offered any practical solutions to this issue. I think this is a very important insight that doesn’t get much attention, because it goes against the status quo. Oracle experts don’t want to spend too much time pointing out that you may not need an Oracle expert at all.

Chapter 5: Extended SQL Trace Data—Cary Millsap

You will not be surprised to find that this chapter is simply selling extended SQL tracing as the cure to all performance issues. When the dominant performance issue is in the Oracle database, SQL tracing information is very valuable. But this isn’t always the case, and in my experience it isn’t even *often* the case. This chapter contradicts what is stated in Chapters 2 and 4, and again in Chapter 7, that you must have performance data from all parts of your system, not just within the Oracle database. The author tells

us that “performance problems cannot hide from Oracle’s extended SQL trace data.” If the performance issue is in the application server, it most certainly can and will hide from SQL trace data. Clearly, I don’t agree that extended SQL tracing is all you need to solve real-world performance issues, so I found this chapter to be lacking in insight. I think it encourages people to waste time looking at very specific details of the Oracle database before they have any evidence that the issue is even database related. We are told that relying on the ratios (such as the much-maligned Buffer Cache Hit Ratio) is a very bad thing. Perhaps relying on SQL trace data is no better. There are times when either of these can help, and there are times when both are useless.

Chapter 6: Direct Memory Access—Kyle Hailey

The author describes his effort to write code that directly accesses the Oracle SGA and how this was used to help him resolve various problems. As with other chapters, this material describes an effort from 1994 using Oracle 7. Using direct access of the Oracle SGA can be dangerous, and even if we assume it was a good idea in 1994, the author doesn’t provide any information on how this technique could or should be used on current systems. The author describes all the ways this can crash your database. Further, the author tells us that his code is “quite difficult to maintain.”

Chapter 7: Compulsive Tuning Disorder—Gaja Krishna Vaidyanatha

In this chapter, the author covers the many aspects of too much tuning, wherein DBAs go off to tune things that don’t matter, have no idea if they have

“We have many world-renowned experts whose tuning methods lead us nowhere near the true problem.”

made any real progress, and don’t know when they should stop tuning. There are many things that I found interesting in this chapter. In the author’s own words, “We have many world-renowned experts whose tuning methods lead us nowhere near the true problem.” Keep this in mind whenever you read or hear anything from tuning experts.

We are told that “ratios are for losers.” However, the same chapter asserts, “This is not to say that all ratios are completely useless.” I’ve had situations in which a ratio did help resolve a performance issue. I conclude that this makes me a “not completely useless loser.” My charming spouse finds this very amusing. I think she laughed a little too much, but I digress.

More seriously, the author brings up a very valuable topic, namely, the need to ask “what is the ROI of a tuning effort?” This is almost always ignored because it is hard and will often point us to something we don’t want to work on, such as a performance issue outside the database. You can’t know that SQL tracing, or a ratio, or anything else is worth looking at until you know why you are tuning and how you



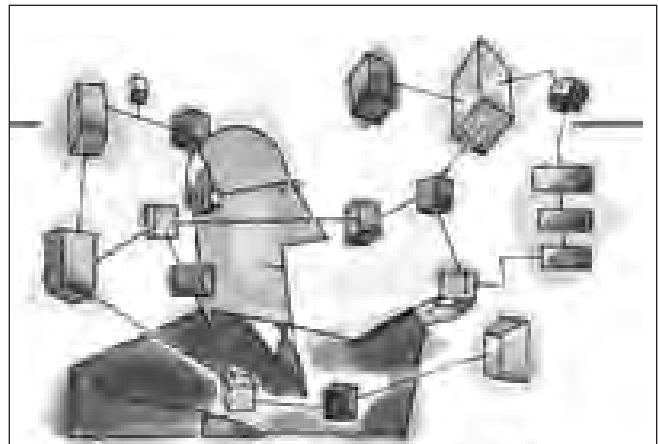
There's No Substitute for Experience

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

We offer a free consultation to discuss:

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

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



**You've got the skill
 now get the tool**

DBFlash™ for Oracle helps you:
Gain visibility of Oracle Wait-Events
Accelerate Oracle performance by more than 100%
Cut costs and find problems before they find you.

www.confio.com
 info@confio.com
 303.938.8282



© 2006 CONFIO

will know when you have tuned enough. This again points to the need to have performance data for all the pieces of your application. The author also tells us that performance tuning is “all about identifying the root cause of an application-level response time issue.” Having made this statement, the author turns to discussing nothing but Oracle database performance issues. I would like to hear about the author’s experiences in tuning application systems where the problems were more complex than a single database on a single machine. This is the type of performance issue I see every day.

Assuming that the performance issue is in the database, the author explains why we need both database- and OS-level performance data to “define the complete performance problem.” In Chapter 5 we were told that SQL trace data is “usually the only data collection needed.” Perhaps this depends on what you think “usually” means. Further, the author states that “any resource consumption and contention that is generated outside the realm of an Oracle database cannot be revealed by the Oracle Wait Interface.” The same is true for SQL tracing. Sources of response time that are outside the database will not show up in a SQL trace.

The author tells us that every tuning effort must have a stated tuning requirement. Unfortunately, the real world doesn’t work this way. I get many requests to see if we can make it faster and if we can add more databases to the consolidated database server. These are performance issues, but they don’t have a stated numerical requirement.

I found the coverage of new features in 10g to be too detailed. I don’t get much out of looking at listings of view definitions. On the other hand, the coverage of exactly how to get various OS-level metrics is very good. This is useful information for anyone tracking down performance issues on a single server.

Chapter 8: New Releases and Big Projects— James Morle

This chapter stands out for two reasons. First, it is simply the best writing in the whole book. This person can tell a story. Second, one of the insights left me speechless, but we’ll get to that later. We are told (again) that OPS doesn’t really scale as well as we have been told it will. We are also told that we need to have extensive testing at all stages of application development. This is a valid observation, but at the same time, does anyone really believe that most organizations do extensive testing at all stages of application development and maintenance? It simply costs too much, and the application itself changes too fast. The story involves the details of multiple performance issues for Oracle 7.0.13 and MTS. As with several other chapters, I’d prefer to hear about the author’s experiences with more recent versions of Oracle.

I enjoyed the part of the story that covers character set issues, since I have been there and done that. If you are facing a character set conversion, this is a good story to read, and I think this issue is a lot more common than most DBAs realize. The discussion of the problems with RAID is very clear and makes the point very well, and the discussion of Functional Partitioning is very good. It provides detail on why you need to use RAC less to make it work better.

Now for the part that left me speechless: The author offers

“Insight 4,” where we are told that application systems should have manual processes in case the application isn’t available. And we are also told that the “vast majority” of businesses can exist for short periods with manual procedures.

Since the specific performance problem that is described is from the early 1990s (when exactly was Oracle 7.0.13 the latest release?), perhaps a business could have a manual process for some part of its mission critical applications back then. The author states that his customer did have a manual process that could handle the “front office” aspects of the application when necessary. This may have

We are told the value of Flashback again, so we have two experts who like Flashback, and one who doesn’t.

been true at that time, but imagine Amazon or eBay having a manual process to display thousands of items to thou-

sands of customers, let alone to gather auction bids, payment information, and so on. The author doesn’t explain how the vast majority of modern businesses could even contemplate a manual process for their applications.

Chapter 9: Testing and Risk Management— David Ruthven

This chapter could have been left out with no impact. The author covers, in great detail, all of the aspects required for an ideal application development process. Topics covered include the need for software reliability, designing for scalability, the benefits of accurate software design, the fact that no parameters should be hard coded, the need for instrumentation, and the many kinds of testing that you should perform. This is all good information, but how many organizations can really follow this idealized approach? Further, there is no mention of how this applies to Oracle or applications that use Oracle software. This chapter may well be very good . . . for a book on software design and the testing needed during software development. I’d like to hear how the author has achieved this ideal development process for a complex, real-world application.

The author does provide a good story about customer support, though. A customer was very upset, and instead of creating a scene, the customer took the author to dinner. This made the author feel so bad, he made sure the customer’s problem got fixed very quickly.

Chapter 10: Design Disasters—Jonathan Lewis

This is another chapter of very detailed database performance tuning stories. The discussion of how the optimizer gets it wrong is good. We are told the value of Flashback again, so we have two experts who like Flashback, and one who doesn’t, and yes, I’m keeping score.

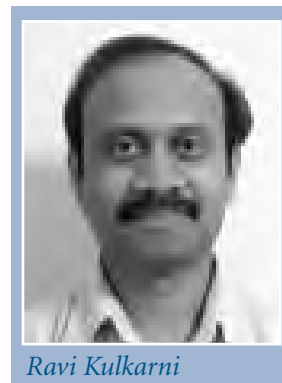
Chapter 11: Bad CaRMa—Tim Gorman

Included here are more detailed stories about performance issues with Oracle version 7. Perhaps this book

(continued on page 26)

Is That Your Final Price?

by Ravi Kulkarni



Ravi Kulkarni

Our intrepid gumshoe trolls the highways and the byways of the world-wide web looking for newsworthy information to bring to our readers.

Eye-Oogh and Oh-Ugh, Oh-Rug and Dee-Mug

Three large user groups—the Independent Oracle Users Group (IOUG, formerly the International Oracle Users Group), the Oracle Applications Users Group (OAUG), and Quest International (the JD Edwards users group)—will join forces at Collaborate '06. It is promised to be “an event like you’ve never seen before.” The Oracle Retail Users Group (ORUG, formerly the Retek User Group) and the PeopleSoft Distributors’ & Manufacturers’ User Group (DMUG) will also join the fun.

Lots of happy songs will probably be sung at Collaborate '06 because it's going to be held in none other than Nashville, Tennessee (a.k.a. Music City), the home of the Grand Ole Opry.

www.oaug.org/conferences/2006/collaborate06/

Is that your final price?

When dual-core CPUs were first announced in the last quarter of 2004, Oracle insisted that a dual-core CPU would require two licenses even though Microsoft decided against requiring double licenses for its database engine. At the time, Jacqueline Woods, Vice President of Global Pricing and Licensing Strategy, compared CPU cores to fruit:

“If you go to a restaurant and order two apples, it doesn't matter how the server delivers the apples to you. The apples could come on one plate or two plates. Either way, you will consume two apples. Processor licensing works the same way. Customers pay by the number of processors they use, whether they are delivered on one chip or two.”

www.oracle.com/corporate/pressroom/jwoods_multicore.html

Facing competitive pressure and customer resistance, Oracle soon offered a concession. In July 2005, it decided that each core would count as 0.75 of a CPU. This time, Jacqueline Woods adopted a more conciliatory tone:

“The goal of Oracle's pricing and licensing strategy is to provide simple and flexible licensing models to meet

our customers' needs. For the past several months, we have been working closely with our customers to address the recent advancements in multi-core processors. To meet these needs, Oracle is updating its licensing policy regarding multi-core processors.”

www.oracle.com/corporate/press/2005_jul_multi-coreprocessorpricingpolicy.html

Presumably the resistance continued and Oracle offered yet another concession in December 2005. The multiplying factor would depend on the vendor. Each core in a Sun CPU would require 0.25 of a license. Each core in an AMD or Intel CPU would require 0.50 of a license, eliminating the price advantage held by Microsoft. All other CPUs would require 0.75 of a license per core. Once again, it was Jacqueline Woods who delivered the welcome news:

“As technology evolves, we have adapted our licensing models to accommodate those changes. In the same way that the shift from mainframes to client/server and client/server to multitiered architectures required new licensing metrics, advancements in multicore chip technology represents the same. These new pricing policies will enable our customers to leverage the advancements in multicore chip technology and derive even more value from their Oracle technology software.”

www.oracle.com/corporate/press/2005_dec_multicoreupdate_dec2005.html

Free pass!

If you you are an 8i Certified Oracle DBA and despair at having to pass two tough exams to get to the 10g level, here is some good news.

In recognition of the time and dedication it takes to stay current with technology, the Oracle Certification Program has made it easier for DBA OCPs to upgrade their credential and stay current. By taking a bridge release exam, candidates can skip one certification release and follow a more streamlined path to upgrade their credential to the latest Oracle DBA certification.

education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=56#2

You can also go directly from Oracle 7 to Oracle 9i. ▲

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, Darrin Swan, at darrin.swan@quest.com. ▲

Long-term event sponsorship:

LOCKHEED MARTIN

CHEVRON

ORACLE CORP.

Thank you! Year 2006 Gold Vendors:

- Confio Software
- Data Domain
- Database Specialists, Inc.
- Embarcadero Technologies
- GoldenGate Software, Inc.
- IT Convergence
- Quest Software
- Quovera, Inc.

For information about our Gold Vendor Program, contact the NoCOUG vendor coordinator via email at: vendor_



TREASURER'S REPORT

Randy Samberg, *Treasurer*

Beginning Balance
October 1, 2005

\$ 48,362.58

Revenue

Membership Dues	2,050.00
Meeting Fees	980.00
Vendor Receipts	6,500.00
Training Day	---
Advertising	---
Interest	56.39
Miscellaneous	7.00

Total Revenue

\$ 9,593.39

Expenses

Regional Meeting	13,111.65
Journal	6,679.28
Membership	---
Administration	2,112.67
Website	---
Board Meeting	622.53
Training Day	---
Marketing	118.97
PayPal	38.92
Miscellaneous	90.00
IRS	---
FTB	331.00
Insurance	507.00
IOUG-Rep	---
Adjustment	500.00

Total Expenses

\$ 24,112.02

NoCOUG Winter Conference

Tuesday, February 14, 2006

Session Descriptions

For more detailed descriptions and up-to-date information, see www.nocoug.org.

Track 1:

Lessons Learned from a Very Large RAC Benchmark

Hanan Hit, *Transparency Software*

In this presentation we will review a very large benchmark that was conducted at Hewlett Packard's benchmark facility in Houston, Texas. The benchmark was developed in order to simulate and measure the performance of a merger of mobile telephone companies. The system described was required to support 64 million customers running a mix of interactive applications and batch processes. The system was required to emulate 34,000 concurrent users by using 12,000 concurrent dedicated Oracle connections along with 500 concurrent batch jobs. In building this benchmark we successfully demonstrated the capacity of this "very large cluster" as well as recommended best practices that maximized its efficiencies, based on systematic measurements of the performance gains that these practices provided.

Some of the Oracle RAC issues we will highlight in this presentation include:

- ▶ The usage of Hyper Messaging Protocol (HMP), and its "scaling" capabilities in such a large environment
- ▶ Bind picking and the issues imposed while running in multiple nodes
- ▶ The effect of tuning of UDP connectivity on the performance and scalability of systems this size
- ▶ Deadlocks and their influence on the overall stability of the system
- ▶ Some suggestions about hidden `init.ora` (`spfile`) parameter usage
- ▶ RAC load balancing and failover testing; suggestions for the method of defining the connection
- ▶ The usage of multiple block sizes
- ▶ Some suggestions on how to reduce the internode communication

Advanced Research Techniques in Oracle—Part I

Tanel Pöder, *Integrid*

Have you ever wondered whether there are diagnostics and performance data-gathering tools available in addition to conventional 10046 tracing and V\$ views in Oracle? Yes, there are—and there are many. This session concentrates on some of those additional tracing and diagnostics mecha-

nisms, which allow us to gather detailed information about Oracle behavior when conventional methods aren't sufficient or some specific problem needs to be solved.

The topics include:

- ▶ Overview of limitations with V\$ views and `sql_trace`
- ▶ Real-time manipulation of text trace output using a pipe and external programs
- ▶ Advanced tracing (events 10200, 10201, 10005, 10704, 10706, 10134, etc.)
- ▶ Overview of how event-checking works internally
- ▶ Tracing of OS operations; `truss` and `strace`
- ▶ Tracing of dynamic library calls
- ▶ Dynamic and transparent OCI call tracing
- ▶ Peeking at variables and memory locations in Oracle
- ▶ An overview of how X\$ tables work internally

Advanced Research Techniques in Oracle—Part II

Tanel Pöder, *Integrid*

This advanced-level session is intended to be a useful source of information on Oracle internals. Some key low-level structures and operations are discussed, such as SGA memory layout and how to debug running Oracle processes to find out exactly what is going on. Usage instructions for various useful, but not very widely known, tracing parameters and events are also provided.

The topics include:

- ▶ Mapping X\$ table columns with memory
- ▶ Using in-memory tracing facilities and X\$TRACE
- ▶ Debugging Oracle processes
- ▶ Interpreting stack traces and binary code
- ▶ Watching code segments and memory regions
- ▶ Invoking an external debugger or executable when an event occurs
- ▶ Obtaining diagnostics in extreme circumstances
- ▶ Invoking Oracle kernel functions with a debugger
- ▶ Some useful scripts for researching Oracle internals

Access and Identity Management for Enterprise Portals

Michael Mesaros, *Oracle Corporation*

Many organizations today are moving toward enterprise portals as gateways to all of their enterprise applications. Enterprise portals offer a number of benefits. They can improve an organization's ability to predict and respond to market dynamics, enhance the productivity of application users, and simplify the management of information technology environments. However, to ensure portal manageability and usability, implementers need to give special consideration to the identity management infrastructure. For example, during portal deployment, many organizations realize the need to consolidate user identities from a variety of directories, databases, and applications. With this need for identity consolidation comes the need to deploy the administrative tools required to support user self-service and delegated administration. If heterogeneous applications

are to be consolidated behind the enterprise portal, a single sign-on solution that can support a variety of application environments is required. Finally, if these portals are going to be made available to users in other companies or agencies, they need to provide the ability to support modern, XML-based standards for identity federation, such as SAML and Liberty.

This presentation describes how organizations deploying enterprise portals can leverage Oracle Identity Management in their environments. We will begin by looking at a typical Oracle Portal implementation. Next, we will give a brief overview of Oracle Identity Management. Finally, we will look at how various features and functionalities of Oracle Identity Management can be applied to the portal environment. While our examples will focus on an Oracle Portal implementation, it is important to note that the benefits described in this presentation may be leveraged in home-grown and third-party portal deployment environments as well.

[Track 2:](#)

Active Session History: A New Performance Monitoring Paradigm

Kyle Hailey, *Embarcadero Technologies*

Oracle version 10g introduces a new performance table called “v\$active_session_history,” also known as ASH. ASH heralds a new era of statistical sampling for performance monitoring as opposed to “exact” statistical collection. Statistical sampling can be more powerful, effective, and efficient than older methods. Using ASH, system performance can be monitored, alerts can be generated, and bottlenecks can be pinpointed in seconds using a few SQL statements with low resource overhead. Learn also how ASH can be simulated on any release of Oracle from version 7 through 10. These same tuning methods can be applied to operating systems and other databases.

Web Services, SOA and BPEL—A Primer

Basheer Khan, *Innowave Technology*

With Web Services and Service-Oriented Architecture (SOA) gaining widespread adoption and Business Process Execution Language (BPEL) becoming the latest industry buzzword, many enterprises are wondering how best to embrace the new software-architecture standard that is slated to become the prevailing software-engineering practice by 2008. This educative presentation will introduce you to Web Services, SOA, and BPEL. The presentation will also include a short demonstration of using the power of BPEL to integrate applications and discuss how to leverage it to jumpstart SOA adoption within your enterprise.

Testing PL/SQL with Ounit

Arnold Weinstein, *Lawrence Livermore National Laboratory*

Software testing is a process used to identify the correctness, completeness and quality of developed computer software. Actually, testing can never establish the cor-

rectness of computer software, as this can only be done by formal verification. Testing can only find defects, not prove that there are none. There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following rote procedure. Ounit is a utility that helps Oracle developers unit-test their PL/SQL code faster, more easily, and more comprehensively than ever before. Ounit offers a powerful graphical interface to utPLSQL, the open-source unit-testing framework for the Oracle PL/SQL language. With Ounit, you can simply point and click your way through testing sessions, and instantly see the outcomes. Because testing is easier and faster, you will test more frequently and more thoroughly.

Hibernate: A Persistence Layer for Java and Oracle

Joel Thompson, *Rhino Systems Inc.*

What are persistence layers and what can they do for you as a developer? What are the functionalities and features of a persistence layer? In this presentation we will sample a popular technology called Hibernate. Hibernate is an open-source persistence layer that you use by specifying an XML file that maps plain old Java beans to the actual database tables. In the XML file you can specify complex relationships, like One-to-Many and Many-to-Many. In addition to the Hibernate technology, we'll discuss some common design patterns. Hibernate is not a panacea for your database I/O, and it does have some drawbacks, which will be discussed.

[Track 3:](#)

Introduction to ADF Faces in JDeveloper 10g—Is It Oracle Forms Developer Yet?

Peter Koletzke, *Quovera*

Oracle has long offered tools that assist with designing, developing, and deploying applications; its current focus is with the Java IDE, JDeveloper. JDeveloper 10g (Release 10.1.3) offers a new set of features, libraries, methods, and Application Development Framework (ADF) Faces, which promise to simplify Java development tasks. The tools that JDeveloper provides to work with ADF Faces allow development of J2EE web applications in an environment that is closer to Oracle Forms than anything previously released. In addition to standard editors and wizards, ADF in JDeveloper simplifies the formerly daunting task of connecting front-end Java code to database elements—something Oracle Forms developers have long taken for granted.

This presentation describes ADF and ADF Faces and how they can assist with development of a Java-based application. It explains how ADF fits into the concept of a Java framework and how the JavaServer Faces (JSF) foundation of ADF Faces is more developer-friendly than previous Java technologies. It also describes the various tools in JDeveloper that support the “ADF development method” and how they ease the task of connecting Java

(continued on page 26)

Session Descriptions

(continued from page 25)

front-end code to the database. Finally, the presentation discusses the type of development shop that will benefit the most from ADF (and ADF Faces) and whether it really is possible to achieve the productivity of an Oracle Forms environment by using this release of JDeveloper.

Raptor: Introducing Oracle's New Graphical Database Development Tool

Kris Rice, *Oracle Corporation*

As a DBA or Database Developer, do you use a variety of tools to browse database objects, create and run SQL, and edit and debug PL/SQL? Raptor is Oracle's newest graphical alternative to SQL*Plus, allowing the Database Developer a convenient way to perform basic tasks. With Raptor you can manage objects in Oracle databases. You can browse, create, edit, and delete (drop) database objects; create, edit, and debug SQL statements and PL/SQL code; manipulate data; export and import database objects; and create reports. You can connect to any target schema using standard Oracle database authentication. In this session we introduce Raptor and demonstrate the latest functionality.

SQL Tuning in Oracle Database 10g: The Do's and Don'ts

Mughees A. Minhas, *Oracle Corporation*

As part of Oracle's ongoing strategy to make the database more self-managing, Oracle 10g has introduced two new solutions, SQL Tuning Advisor and SQL Access Advisor, which together can eliminate the need for manual tuning. Users can now let the Oracle database tune their enterprise applications for them. This paper discusses the two new advisors, describing the internals of how they work, best practices for their usage, their limitations, and their benefits.

Performance Diagnosis Usage Model in Oracle Database 10g

Prabhaker Gongloor, *Oracle Corporation*

Oracle Database 10g has a rich set of capabilities that make the identification of the root cause of performance problems quick and easy. This presentation discusses the usage model, various diagnostic solutions that are available, when to use which solution, and their most effective use. It covers functionality introduced in Oracle 10.2 that allows DBAs to analyze transient performance problems, and compare current and historical performance to baselines (preserved snapshot sets) to detect changes in configuration and workload. ▲

Book Review

(continued from page 21)

should be called "Oracle 7 Insights"? The higher-level story of the "data-driven" application is good because it points out the problems with trying to write a completely generic application. The author offers the following observation: "Sometimes the cleverest people made the most fundamental errors." This concept has come up in several chapters, and it bears repeating. Be careful when dealing with experts. Sometimes you really are better off with solutions that are less exciting and have been proven over time. The story related in this chapter is a good example of an organization buying into the vision of an expert and paying an incredible price for it.

ADVERTISING RATES

Contact: **Nora Rosingana**

325 Camaritas Way · Danville, CA 94526

Ph: (925) 820-1589

The NoCOUG Journal is published quarterly.

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

Personnel recruitment ads are not accepted.

Appendix: Join the BAARF Party (or Not)— James Morle and Mogens Nørgaard

Since this issue has been discussed for many years, I would be surprised if the conclusions presented are news to anyone.

Conclusion

While there is interesting material presented, the real value of this collection of authors was missed. If the various topics and themes had been identified, and if each of the authors had discussed each topic and either had come to a consensus or had explained in detail their divergent views, I think the book would have been much more valuable and would have come a lot closer to the stated goal of providing "Oracle insights." As examples, I'd really like to hear the various authors explain how they do or don't need OS-level statistics for resolving performance issues, whether OPS/RAC really is or isn't mainly marketing hype, and whether Flashback is useless or not.

I think this book had great potential, but by isolating the authors in their own private worlds, the potential was missed. This is a collection of material that lacks the power that would have come from interaction among these authors.

About the Authors

The book cover says that the authors are eleven leading authorities on Oracle's database engine who have written this book to share unique insights on how best to exploit the software. ▲

Optimizing Oracle—Performance by Design

by Jonathan Lewis

Author of *Cost-Based Oracle Fundamentals*

May 15-17, 2006

San Ramon Marriott

2600 Bishop Drive, San Ramon, CA 94583

Sign up
now! Seating
is limited!

Management Summary:

Why should you send your DBAs and senior developers on this course? Because you want to make the best use of their time, and you want to install systems that don't exceed budget, waste time, and perform badly. If you've paid the Oracle license fees, you need people who understand how it works so that they can avoid the worst design errors, and get the best value for money from your investment.

There are only a few special mechanisms that are critical to building, or maintaining, an efficient Oracle system. Once your DBAs understand them, and have the tools to help them investigate what's going on, they will be able to solve problems more quickly, and be in a much better position to identify safe and cost-effective opportunities for improving your systems.

The course contains a lot of answers to current questions—but more importantly, it shows people how they can rapidly work out answers to the new questions that keep appearing.

Day 1—Infrastructure

- Parser and Optimizer costs
- Undo
- Redo
- Latches

Day 2—Application Structure

- Locking and Queuing
- Application Overheads

- Troubleshooting—Why are we waiting?
- Troubleshooting—How to avoid work

Day 3—Optimizing SQL

- Cost-based Optimization—How the arithmetic works
- Cost-based Optimization—How to use and abuse it
- Understanding EXPLAIN PLAN
- Open Forum and round-up

Registration Cost Includes:

- Three full-day presentations
- A copy of Jonathan's latest book *Cost-Based Oracle Fundamentals*
- Course handouts
- CD containing test scripts
- Light breakfast and buffet lunch on all three days

How to register:

- The cost is \$1,250 for NoCOUG members and \$1,500 for non-members.
- Reserve your place by sending e-mail to jonathan@jlcomp.demon.co.uk.
- Payment will be through PayPal and must be received by May 8.
- The maximum class size is 32.

For more information, please visit:

www.jlcomp.demon.co.uk/seminar.html

or send e-mail to jonathan@jlcomp.demon.co.uk.

It's Registration Time Again!

I know the year just zoomed by. I'm sure it's due to the great 2005 NoCOUG program. The board has another stellar year planned for you. Please register early to continue getting this "award winning" Journal and streamlining your conference registration. For our 2005 members who have not yet registered for 2006, this issue of the NoCOUG Journal is complimentary. We know you plan to register.

Check the address label on the back of the Journal. If there is an asterisk (*) affixed to your name, you are already registered for 2006. Keep in mind that the mailing list was pulled on 1/22/06. The (*) status indicator will only be used on this issue.

All 2005 members should have received an email or U.S. mail reminder by now. That will include your NoCOUG ID for online registration at www.nocoug.org. Contact me with any questions at membership@nocoug.org.

Hope to see you at the coming conferences!

Joel Rosingana
NoCOUG Membership Director

NoCOUG Winter Conference Schedule

February 14, 2006, at the Oracle Conference Center, Redwood Shores, CA

Please visit www.nocoug.org for updates, directions, and to submit your RSVP.

Cost: \$40 admission fee for nonmembers. Members free. Includes lunch voucher.

- 8–9AM **Registration and Continental Breakfast**—Refreshments served
- 9–9:30 **General Session and Welcome**—Darrin Swan, NoCOUG President
- 9:30–10:15 **Keynote:** *Oracle Fusion Middleware Value Proposition for Oracle, Non-Oracle, and Custom Applications* by Vijay Tella, Chief Strategy Officer, Oracle Applications Server, Oracle Corporation
- 10:15–10:45 **Break**
- 10:45–11:45 **Parallel Sessions #1**
- Track 1:** *Lessons Learned from a Very Large RAC Benchmark* by Hanan Hit, Transparency Software
- Track 2:** *Active Session History: A New Performance Monitoring Paradigm* by Kyle Hailey, Embarcadero Technologies
- Track 3:** *Introduction to ADF Faces in JDeveloper 10g—Is It Oracle Forms Developer Yet?* by Peter Koletzke, Quovera
- 11:45AM–1PM **Lunch**
- 1–2 **Parallel Sessions #2**
- Track 1:** *Advanced Research Techniques in Oracle—Part I* by Tanel Pöder, Integrid
- Track 2:** *Web Services, SOA and BPEL—A Primer* by Basheer Khan, Innwave Technology
- Track 3:** *Raptor: Introducing Oracle's New Graphical Database Development Tool* by Kris Rice, Oracle Corporation
- 2–2:15 **Break**
- 2:15–3:15 **Parallel Sessions #3**
- Track 1:** *Advanced Research Techniques in Oracle—Part II* by Tanel Pöder, Integrid
- Track 2:** *Testing PL/SQL with Ounit* by Arnold Weinstein, Lawrence Livermore National Laboratory
- Track 3:** *SQL Tuning in Oracle Database 10g: The Do's and Don'ts* by Mughees A. Minhas, Oracle Corporation
- 3:15–3:45 **Raffle and Refreshments**
- 3:45–4:45 **Parallel Sessions #4**
- Track 1:** *Access and Identity Management for Enterprise Portals* by Michael Mesaros, Oracle Corporation
- Track 2:** *Hibernate: A Persistence Layer for Java and Oracle* by Joel Thompson, Rhino Systems Inc.

Session descriptions appear on page 24.

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

NoCOUG
P.O. Box 3282
Danville, CA 94526

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

Is this your last issue?
See page 27.