Expert Oracle Practices

Oracle Database Administration from the Oak Table

Melanie Caffrey, Pete Finnigan, Randolf Geist, Alex Gorbachev, Tim Gorman, Connie Green, Charles Hooper, Jonathan Lewis, Niall Litchfield, Karen Morton, Robyn Sands, Jože Senegačnik, Uri Shaft, Riyaj Shamsudeen, Jeremiah Wilton, Graham Wood

Apress®

Expert Oracle Practices: Oracle Database Administration from the Oak Table

ISBN-13 (pbk): 978-1-4302-2668-0

ISBN-13 (electronic): 978-1-4302-2669-2

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

President and Publisher: Paul Manning
Lead Editor: Jonathan Gennick
Technical Reviewers: Melanie Caffrey, Arup Nanda, Peter Sharman
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Coordinating Editor: Fran Parnell
Copy Editors: Sharon Wilkey, James A. Compton
Compositor: Molly Sharp
Indexer: Brenda Miller
Artist: April Milne
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail info@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/info/bulksales.

The source code for this book is available to readers at www.apress.com. You will need to answer questions pertaining to this book in order to successfully download the code.

Contents at a Glance

Contents

Foreword

Like many red-blooded Americans, my friend—let's call him John—dabbles in carpentry whenever he can motivate himself to move from the computer to produce something more tangible. Recently he discovered that although motivation can be a great catalyst, it can never replace skills. In a grand testimony to that truth, he has never been able to produce anything more than a hole in a board or two pieces of wood out of one. His most recent story parallels the teachings of this book.

Embarrassed by the incessant boasting of neighbors and friends about their great accomplishments in the fine art of carpentry, John woke up one day determined to turn the tide—become skilled at carpentry. He even picked out a project—a garden shed to store all the mysterious but seemingly useful stuff that occupied the modest real-estate of his garage, forcing his cars to become refugees on the driveway. No way, no sir, he told the cars—pretty soon they would be able to go back to their promised homeland, just as soon as he banished the clutter to the garden shed, which he would build with his newly acquired skill. The cars seemed to honk in agreement, or perhaps he imagined it.

Charged with new passion, he rushed off to a store for homebuilders full of other trumped-up newbies like himself. He chose a book on do-it-yourself garden sheds. He bought the materials and the all-important tools. He did everything by the book. But two years have passed by, and all he has been able to produce are numerous cuts, bruises, scratches, and countless pieces of wood wasted as a result of not being cut properly—either cut too short or angled too acutely. In the course of these years, my friend added more tools to the collection—tools that supposedly make a carpenter out of anyone with the right inclination and bent of mind—and more wasted wood, drill bits, and saw dust. The shed, not even a remote resemblance of it, never saw the light of the day.

What went wrong? The missing ingredient was the right foundation. My friend was not a skilled carpenter. All the good intentions and excellent tools didn't do a thing to get him the right skills. What he should have invested in early on was time in learning the trade. He should have spent time in small but sure ways, learning how to use the tools he acquired. He should have learned how to measure pieces of wood before cutting them. No, he should have learned how to measure twice before cutting! He should have learned how to master the tools before actually using them on a project. But, above all, he should have realized that sophisticated tools and do-it-yourself books are not substitutes for a conceptual foundation on anything that requires expertise—the building of garden sheds included.

I tried to persuade my friend to give up on that project and hire a professional. But John doesn't give up easily. Instead, he began anew. This time he invested in learning the craft, in making small cuts and completing small projects to build up his skills. Once he was confident, he started on the shed. Finally, late one fall and after an embarrassing amount of time, the shed was done. John put it on concrete slabs and stored all his lawn and garden tools in it. For the first time in years, John's cars went inside his garage. The neighbors and I gathered. Everyone was smiling. John was beaming as he showed his handiwork to us all.

Then, the snow came.

You see, it was fall 2008, and the 2008–2009 winter turned into one of worst we had seen, with snowfalls going up to 13 inches or so every other day. The snow and ensuing cold spell left a huge pile of ice on John's beloved shed. One day, after days of enduring under the heavy accumulation, the shed buckled under the weight. We all watched helplessly as his blood and sweat just went up in a pile of wood, exposing everything inside it.

We neighbors tried to enlighten John on the futility of his attempts, but he won't budge. He spent hours and hours fixing the broken shed, but after every effort it would break. His attempts drew mixed emotions from the neighbors. Most people applauded his attempts at a positive attitude by making these efforts to fix the shed. But there were a few well-wishers who advised him to give up, to admit failure, and to have a prebuilt shed installed by professionals.

Where did John go wrong this second time? At least he started off in the right way, but he didn't have the expertise in the business of sustenance. Building a shed is only half the story; making it tough enough to withstand the elements is the other half—one that proverbially separates the men from the boys.

How does John's story parallel the teachings in the book? In many ways, John's experience paraphrases the activity of many professionals engaged in the technology trade—Oracle technology included. Even some so-called experts are not immune to the temptations of a shortcut. The dependence on tools is a foregone conclusion in many cases, with greater emphasis on seemingly sophisticated ones. What's worse, it is a common belief in the corporate world that a go-getter attitude coupled with the right tools enables one to do any job, and do it well. That belief often leads to a reinforcement of the pattern of behavior that rewards an effort—not necessarily the right effort.

The expert must have the right tools, but the sophistication of the tools is not as important as their right use. Using the handle end of a screwdriver to drive a nail when a hammer is lying right in front of you is not just idiotic but goes against the "no screwdrivers were harmed in the making of this project" theme. Using the right tool for the right job is what makes an expert an expert. Consider Chapter 3, in which Connie, Uri, and Graham show you how to use ADDM to resolve some thorny issues in performance. Many people I meet often balk at using a tool such as ADDM. Why? Their reasons are often a mixture of ignorance, skepticism, and masochistic viewpoints that a tool can't be better than their bare hands. But for a DBA to avoid using ADDM is like a nurse in a busy hospital wanting to throw away the automatic blood-pressure monitor. Sure, a nurse can get by with a manual cuff. So can an expert DBA get by without ADDM. But ADDM makes life easier and more efficient, and an expert will take advantage of ADDM in order to get those benefits.

Getting back to the story, Alex shows in Chapter 1 how the behavior of doing just *something*, even doing something completely ineffective, is often rewarded. Work that does not get results represents lost productivity and delayed execution. Alex painstakingly demonstrates how it hurts productivity to make assumptions rather than to take accurate measurements, and that a major impediment to the resolution of any problem is the presence of assumptions. Many chapters in this book try to drive home the point that the practice of measuring instead of assuming should be the first order of business in any resolution process. In Chapter 6, Karen shows how to instrument and measure code for hidden issues, and to rely on proper measurement techniques. In Chapters 8 and 9, Charles and Randolf describe how to develop a system that measures performance and practically puts the culprit of a performance problem right in front of you. And Robyn shows in Chapter 13 how to use statistics to find the proverbial "red rock" representing a performance problem buried under a mountain of collapsed rubble.

About eight years ago, I was delivering a technical session on buffer busy waits, how to diagnose them, and how resolve them. At the end of the session, during the question-and-answer period, an attendee asked me a very interesting question: SQL Server didn't suffer conditions such as buffer busy waits, so did that mean that SQL Server was better than Oracle? I resorted to an allegory as a response. I said that I took my car for oil changes, but not my bicycle; so did it mean that my bicycle is technologically superior to my car? The response, as expected, drew a chuckle from the audience and drove home the point about the futility of comparison between those two databases. In the audience was Tim Gorman, one of the authors of this book, who also responded to the question from the attendee. Tim asked the attendee how he knew that SQL Server did not indeed have the condition known as buffer busy waits. Perhaps SQL Server did suffer from those, but he just didn't know how to measure them and resolve them.

While the audience was mulling over the response, I realized the enormous universal truth behind Tim's response. Most of the audience, even seasoned DBAs, weren't even aware of the buffer busy waits

in Oracle, or were quite hazy in their understanding of them. The point was not about the specific condition or what it is called. The point was about the universal acceptance of the absence of something that has not been characterized yet. The questioning of accepted knowledge is the key to getting closer to the truth. But it is not just the act of skepticism, but the examination of the knowledge that matters. Centuries ago, many believed the world was flat. Had Newton not discovered gravity, the concept of a round earth might have defied logic and been relegated to the category of myth. Instead, the ability to understand gravity and correlate that understanding with other observations about the world led to widespread acceptance that the world was, in fact, round.

In Chapter 12, Riyaj explains the little known and often misunderstood concept of latch contention. Lack of understanding of latch contention leads to design of systems that do not scale well, and to issues that remain unresolved for quite long periods of time. Riyaj's chapter should also educate readers enough to explain some of the other mysteries that have plagued them. The concept of deeper understanding to dispel myths is reinforced in Jonathan's Chapter 11 on statistics. These chapters highlight the need to build a system for scalability, which was the precise cause of the failure in the case of the garden shed made by my friend John.

Many of the authors, such as Jonathan and Jože, portray themselves not as consultants or DBAs extraordinaire, but merely as researchers. This self-attribution is a testimony to the fact that they are still learning and will continue to do so. They have epitomized the belief that knowledge is never a destination; it's a journey. It is not supposed to end. Any pretense of accomplishing the end is exactly that—a pretense. The other authors hold different titles, but all of them espouse the same principle: they never seek the end of knowledge; they create it, encourage others to create ever more, and then they absorb it all like a sponge. As a famous contemporary author once wrote, the worst part of knowledge is that the more you know, the more you learn how much you don't know. The best part is that this realization results in the creation of knowledge exponentially. The enlightened group of authors behind this book have attempted to kick off that very process. By filling in some voids in your knowledge, they hopefully tickle your imagination and inspire you to explore more, to add to the entire gamut of knowledge as a whole.

Going back to the original discussion, what makes an expert? Is it winning some titles or passing some certification tests? Many of these authors have been honored for their excellence in some form or other. Some are formally recognized by Oracle (as Administrator Certified Expert and ACE Directors, and as an Author of the Year, for example). Others are recognized by by formal certification authorities (as Oracle Certified Professionals and Oracle Certified Masters). All are recognized by the strictest body of all—the OakTable Network itself. But all these recognitions pale in comparison to the biggest accomplishment of all—their recognition by their peer group: the user community. Ultimately, what makes the difference is the acceptance of their excellence by you, the reader and the user of Oracle technology. All of these authors participate heavily in the user community, give knowledge back to the community—internationally or in their local spheres—and shape the progress of accumulation of collective knowledge. When they come together to pen their thoughts in the format of a book, the result can be nothing short of sheer pleasure.

An expert also must be well rounded, albeit is expected to be an expert in a subset of the technologies. The dissemination of knowledge is not complete without a deeper understanding of some of the supporting technologies. In Chapter 5, Niall shows how to effectively manage Oracle on Windows through a thorough explanation of the subtle nuances of Windows and how to get around them. In Chapter 7, Jože shows how to exploit a little known but powerful feature of the cost-based optimizer called the *extensible optimizer*. In Chapters 14 and 15, world-renowned security expert Pete shows how to deviate from performance aspects and bulletproof your database from security threats—a goal that attained considerable importance lately. In Chapter 10, another renowned expert named Tim takes a detour to the world of superlatives—big databases, longer load times, and so on, to educate readers in the fine art of managing beasts of databases.

Technology changes faster than you can spell it, and most of us are stuck in a loop of understanding deeply and then becoming obsolete. However, there is no substitute to solid foundation in design, as Melanie describes in Chapter 4. Changes are inevitable, and keeping up with the pace of change often poses challenges for experts in any domain. Pace of change is one of the factors leading to the pollution of knowledge, as performance-tuning guru Jonathan explains in Chapter 11 in relation to the gathering of statistics. And Jeremiah's Chapter 2 is the result of change; it explains the newest phenomenon to hit the experts—Oracle in the cloud.

As a part of my extra-curricular activities, I often travel outside my native United States to other countries to provide seminars, trainings, sessions, and to participate in discussion forums. In fact, I am writing this foreword in the middle of a two-week, four-country tour across Europe. While on the plane from Estonia to Germany, I was reflecting on the lesser-appreciated aspects of knowledge processing, the ones too subtle to notice but too important to ignore. Cultural aspects play a huge role in the collection and dissemination of knowledge. Mankind is far from being homogeneous; it's submerged in plurality—languages, cultures, customs, and expressions shaped by traditions. Ignoring this plethora of differences is not only detrimental to synergy, but also to the very essence of the knowledge-begets-knowledge belief. Not understanding cultural and customary differences can only lead to the inefficiency of the process, if not to a more dangerous reversal and creation of ignorance puddles. A vital characteristic of this cast of authors is that they represent many parts of the globe. The ideas they propound, the mannerisms they convey, are shaped and influenced—some more than the other—by the customs and traditions they uphold. A profound example is the United States, a melting pot of many cultures in which diversity gives strength to the society and is celebrated. It's diversity of the authors that gives this book an all-round perspective, something that is literally world-class.

Finally, it's the supporting cast that makes the last run of the technical aspect of the book a huge success. The authors didn't just get a free rein in whatever they wanted to write; their work was thoroughly scrutinized by a team of technical reviewers—also members of the OakTable Network and respected professionals in their own right. Even the editor—Jonathan Gennick—is an Oakie as well, making it an Oakie production end to end.

While I have been honored to write the foreword, I stoop under the weight of the huge expectation of summarizing the essence of these chapters from the stalwarts of the Oracle user community into a few pages. In conclusion, I will reiterate the message that has been delivered by many authors in many shapes or forms—excellence is not an end or even a journey. It is not about gathering as much knowledge as possible; it's about building a very strong foundation on whatever you know, expanding the horizon but not at the cost of a weakened foundation. It is about refining your foundation, and reinforcing it by constantly subjecting it to assessment and analysis, and finally providing evidence (quantifiable or otherwise), not opinions based on perceptions.

Happy reading!

Arup Nanda
Chief database architect, Starwood Hotels and Resorts
OakTable Network member
Oracle ACE Director

About the Authors

Melanie Caffrey, Pete Finnigan, Randolf Geist, Alex Gorbachev, Tim Gorman, Connie Green, Charles Hooper, Jonathan Lewis, Niall Litchfield, Karen Morton, Robyn Sands, Jože Senegačnik, Uri Shaft, Riyaj Shamsudeen, Jeremiah Wilton, Graham Wood

Melanie Caffrey

Melanie Caffrey is a senior development manager for Oracle Corporation, providing front-end and back-end Oracle solutions for the business needs of various clients. She is co-author of several technical publications, including *Oracle Web Application Programming for PL/SQL Developers*, *Oracle DBA Interactive Workbook*, and *Oracle Database Administration: The Complete Video Course*, all published by Prentice Hall. She has instructed students in Columbia University's Computer Technology and Applications program in New York City, teaching advanced Oracle database administration and PL/SQL development. She is a frequent Oracle conference speaker.

Pete Finnigan

Pete is a world-renowned expert in the area of Oracle security, providing consultancy, design expertise, security audits, and training, all in the area of Oracle security. Pete is a member of the OakTable Network. He has spoken regularly all over the world at various conferences such as those of the United Kingdom Oracle Users Group (UKOUG), PSOUG, Black Hat, and Risk. Pete is a published author on Oracle security, and researches and writes about the subject regularly. Pete also runs his website, `www.petefinnigan.com`, dedicated to Oracle security and providing help and free tools to the community.

Randolf Geist

Randolf Geist has been working with Oracle software for 15 years now. Since 2000 he has operated as a freelance database consultant focusing primarily on performance-related issues, and in particular helping people to understand and unleash the power of the Oracle cost-based optimizer (CBO). He is writing on his blog about CBO-related issues and is also regularly contributing to the official Oracle Technology Network (OTN) forums. Randolf is a member of the OakTable Network, the Oracle ACE program, and is an Oracle Certified Professional DBA for Oracle Versions 8i, 9i, and 10g. He also maintains SQLTools++, an open source Oracle GUI for Windows.

Alex Gorbachev

Alex Gorbachev is a respected figure in the Oracle world, and a sought-after leader and speaker at Oracle conferences around the globe. He is an OakTable Network member, and has been recognized as an Oracle ACE Director for his contributions to the community and unmatched set of skills. He is the founder of the *Battle Against Any Guess* movement, promoting scientific troubleshooting techniques. He is currently the chief technology officer at The Pythian Group.

Alex has worked for The Pythian Group in several roles. He began by leading a team of database experts in Ottawa. He then moved to Australia to build the company's presence in the East Asia Pacific region. Now he is back in Ottawa as The Pythian Group's chief technology officer. In all his work, Alex continues to work toward bridging the gap between business and technology. The search for the perfect fit between technology, engineering talents, and business process is what keeps him up at night.

Tim Gorman

Tim Gorman began his IT career in 1984 as a C programmer on Unix and VMS systems, working on medical and financial systems as an application developer, systems programmer, and systems administrator. He joined Oracle Corporation in 1990 as a consultant, became an independent consultant in 1998, and has worked for SageLogix since 2000. Gorman is the co-author of *Essential Oracle8i Data* Warehousing and *Oracle8 Data Warehousing*. He specializes in performance-tuning applications, databases, and systems, as well as data warehouse design and implementation, backup and recovery, architecture and infrastructure, and database administration. Gorman still considers himself a pretty good coder, although the market for C programs has dried up somewhat lately.

Connie Green

Connie Green has 20 years of experience with Oracle products, the last 12 years working for Oracle U.S. Server Development. Connie's expertise is in database performance tuning. She designed and developed Statspack from its initial release. Over the years Connie has been privileged to work with many talented kernel developers, performance engineers, and support engineers, having been involved with library cache development, buffer cache development, and the query optimizer.

Charles Hooper

Charles Hooper is the IT manager and an Oracle database administrator at K&M Machine Fabricating, where he has been employed since 2000. His role in the company extends well beyond Oracle database administration responsibilities, providing opportunities for database performance tuning, network administration, programming, hardware/software troubleshooting, mentoring fellow IT staff, and end-user training for the Infor ERP Visual system as well as other custom-developed applications. Charles is well known among the user community of Infor Visual Enterprise because of his years of participation in various user forums answering technical questions, providing the only Oracle presentation at Infor's 2008 user's conference, and providing presentations to regional user groups. Prior to joining K&M, he was a computer/technology instructor and Novell NetWare administrator. He holds a bachelor of science in mathematics and computer science. Much has changed in the computer industry since his first home computer in 1981, which offered 5KB of memory, a surprisingly small amount today as manufacturers offer home computers with as much as 24GB of memory.

Jonathan Lewis

Jonathan Lewis has been working in the IT industry for nearly 25 years, and has been using the Oracle RDBMS for more than 20. For the last 16 years he has been working as a freelance consultant, often spending only one or two days at a time with any client to address critical performance problems. He also advises on design and implementation problems, and on how to make best use of the most appropriate Oracle features for a given project.

Jonathan is renowned throughout the world (having visited 42 countries at last count) for his tutorials and seminars about the Oracle database engine and how to make best use of it. His exceptional ability has earned him an O-1 visa from the United States, allowing him to do consultancy and lecture work there.

Jonathan has written two books about Oracle (*Practical Oracle8i*, Addison-Wesley, 2000; *Cost-Based Oracle Fundamentals*, Apress, 2005) as well as contributing to two others (*Oracle Insights*, Apress, 2004; *Oracle Database 10g New Features*, McGraw-Hill, 2004). He also writes regularly for the UKOUG magazine, and occasionally for other publications around the world. In the limited amount of time he has left over, Jonathan also publishes high-tech Oracle articles on his blog at `http://jonathanlewis.wordpress.com`.

Niall Litchfield

Niall Litchfield is a DBA of 15 years standing, with extensive experience of running databases on various x86(64) platforms, and with a particular bias toward Microsoft Windows. Niall began professional life as a "bean-counter" for KPMG; consequently he is a DBA who respects auditors, not least because they have to stand in unheated warehouses in the middle of winter watching other people count things, something he never wants to do again. His dad was a *real* engineer crafting genuinely useful objects with fine tolerances and that, coupled with an early experience in which he showed that the two competing macro-economic theories of the time both explained actual behavior equally well, and that neither explained it adequately, rather explains his evidence-based approach to databases in general and performance tuning in particular.

Karen Morton

Karen Morton is a consultant and educator specializing in application optimization in both shoulder-to-shoulder consulting engagements and classroom settings. She is a senior principal database engineer for Agilex Technologies (`http://agilex.com`), a Chantilly, Virginia company founded by Bob LaRose and Jay Nussbaum. Agilex offers customers innovative thinking on leveraging advanced technologies within the healthcare, federal, intelligence, commercial, and public sectors to fully realize the value of information.

For over 20 years, Karen has worked in information technology, starting out as a mainframe programmer, developer, DBA, data architect, and now as a researcher, educator, and consultant. Having used Oracle since the early 90s, she began teaching others how to use Oracle over a decade ago.

She is a frequent speaker at conferences and user groups, an Oracle ACE, and a member of the OakTable network. She blogs at `http://karenmorton.blogspot.com`.

Robyn Sands

Robyn Anderson Sands is a software engineer for Cisco Systems. In a previous incarnation, she worked in industrial engineering, manufacturing development, and quality engineering at Lockheed Martin, supporting the P-7, F-22, and C-130J programs. Robyn has been working with databases and Oracle software since around 1996. She began her work with Oracle by developing tools for scheduling, composite fabrication capacity modeling, and engineering workflow, and progressing to the implementation and administration of data warehouse, PeopleSoft, and SAP systems. Current projects include "architecting" and managing the development of embedded database systems for Cisco customers, and searching for new ways to design and develop database systems with consistent performance and minimal maintenance requirements. She has been a speaker at UKOUG, Miracle conferences, Oracle OpenWorld, and the Hotsos Symposium. She occasionally posts random blog entries at `adhdocddba.blogspot.com`.

Jože Senegačnik

Jože Senegačnik has more than 20 years of experience in working with Oracle products. He began in 1988 with Oracle Database version 4 while working for the City of Ljubljana, Slovenia, where he had charge over the city's municipal and geographic information systems. From 1993 to 2003, he worked in developing GIS systems for the Surveying and Mapping Authority of the Republic of Slovenia, and in the development of applications for other governmental institutions, all based on the Oracle database. More recently, he has specialized in performance optimization, having developed his own toolset for monitoring performance and analyzing trace files.

Jože is an internationally recognized speaker, and a member of the highly respected OakTable Network (`oaktable.net`). He is a regular speaker at user-group conferences, especially those put on by the Slovenian Oracle Users Group (SIOUG), the Independent Oracle Users Group (IOUG), and the United Kingdom Oracle Users Group (UKOUG). He also speaks routinely at the Hotsos Symposium and Oracle OpenWorld. In addition to sharing his knowledge through conference talks, Jože conducts technical seminars organized either by Oracle University or himself. He was awarded Oracle ACE membership for his long record of positive contributions to the Oracle community.

Uri Shaft

Uri Shaft received a Ph.D. from the University of Wisconsin-Madison, specializing in database systems. He was the lead developer of the innovative QUIQ hybrid database and information retrieval system. Currently, Uri works for Oracle USA in the server manageability group. He is the developer in charge of the Automatic Database Diagnostic Monitor (ADDM), and of components that analyze the Active Session History (ASH) sampled data. Uri's areas of expertise include multimedia support in database systems, multidimensional indexing theory (and practice), and database performance diagnosis and tuning.

Riyaj Shamsudeen

Riyaj Shamsudeen is the principal DBA and president of OraInternals (`www.orainternals.com`), a performance/recovery/EBS11i consulting company. He specializes in RAC, performance tuning, and database internals. He also frequently blogs about these technology areas in his blog, `http://orainternals.wordpress.com`. He is a regular presenter in many international conferences such as Hotsos, COLLABORATE, RMOUG, SIOUG, and UKOUG. He is a proud member of OakTable network. He has more than 16 years of experience using Oracle technology products and more than 15 years as an Oracle DBA/Oracle Applications DBA.

Jeremiah Wilton

Jeremiah Wilton has worked with Oracle technology since 1994. His main claim to fame is having been Amazon.com's first database administrator, back in the pre-IPO days. For seven years, he helped Amazon.com survive exponential scaling and a wide variety of nearly catastrophic technology failures. Jeremiah owned and ran ORA-600 Consulting for a number of years, until it was acquired by Blue Gecko, a global provider of remote administration for Oracle, MySQL, and E-Business Suite. Jeremiah also teaches the Oracle certificate program for the University of Washington. Jeremiah is an Oracle Certified Master, a member of the OakTable, and a frequent presenter at industry conferences and user groups. His publications and whitepapers can be found at `www.bluegecko.net`.

Graham Wood

Graham Wood is an architect in the database development group at Oracle. Most of his 20 years at Oracle have been spent in performance-related areas, including designing and tuning large high-performance systems, building monitoring tools such as Statspack, and in architecting performance and scalability features into the database engine itself. More recently Graham was the architect of the Oracle Manageability team tasked with simplifying the process of tuning the operation of the database, which resulted in the development of AWR, ASH, and ADDM to provide automatic tuning.

About the Technical Reviewers

Melanie Caffrey

Melanie Caffrey is a senior development manager for Oracle Corporation, providing front-end and back-end Oracle solutions for the business needs of various clients. She is co-author of several technical publications, including *Oracle Web Application Programming for PL/SQL Developers*, the *Oracle DBA Interactive Workbook*, and *Oracle Database Administration: The Complete Video Course*, all published by Prentice Hall. She has instructed students in Columbia University's Computer Technology and Applications program in New York City, teaching advanced Oracle database administration and PL/SQL development. She is a frequent Oracle conference speaker.

Arup Nanda

Arup Nanda has been an Oracle DBA for more than 16 years—touching all aspects of database management and architecture—from modeling to performance tuning and disaster recovery. He has written more than 300 articles, co-authored 4 books, spoken at 150 technical conferences, and delivered a number of full-length training sessions. In 2003, he was awarded DBA of the Year by Oracle. He is an Oracle Certified Professional DBA, an OTN ACE Director, and a member of the OakTable Network. He lives in Connecticut with his wife Anu and son Anish.

Peter Sharman

Peter Sharman is a curriculum developer with Server Technologies Curriculum Development at Oracle Corporation. He has over 20 years of Oracle experience, and has been a speaker at several Oracle OpenWorld, Miracle Database Forum, and RMOUG Training Days conferences. Pete has multiple OCP and OCM certifications, and is currently responsible for developing training on the Oracle Enterprise Manager product family.

CHAPTER 1

Battle Against Any Guess

by Alex Gorbachev

During my experience with Oracle, I have become very engaged in the user community. I've been a frequent visitor on the Oracle Technology Network forums and the Oracle-L list and have become a regular participant and contributor at user group conferences and other events. My experience started with seeking help and gradually shifted towards helping others with their issues. My growth in Oracle has correlated with the booming popularity of the Internet, over which it becomes very easy to both seek and give advice.

While the Internet increases community participation, it also causes some dysfunction that can lower the quality of the information. Many times I have seen online discussions branch into controversial arguments in which the "combatants" are going by guesswork. It is surprising how few people will stop to test what really happens, and instead will battle endlessly over what might happen or what they believe ought to happen.

While my contributions to the community have been usually rather technical, this chapter is more generic and rather motivational reading. My first attempt at a motivational contribution was creating BattleAgainstAnyGuess.com, or the BAAG Party for short, in June 2007. This is where the title of the chapter comes from. The trigger to establish the BAAG Party was coming across yet another quest for guess-based solutions on the Internet; and I wanted something generic to refer to every time I see such symptoms. Thus, I want to start this chapter by showing some examples of guess-provoking questions.

Guess Hunting

The way you ask a question is crucial. A badly formed inquiry is almost guaranteed to attract guess-based solutions. Here is one example of seeking a quick solution from the Oracle-L list:

> *"I'm also having performance issues with 10g. Why would my dictionary queries take a long time to return? ... In 9i they used to take seconds, now they take minutes or they just never come back..."*

When reading this question, it is difficult to divine precisely what the problem is that the writer is experiencing. Without a clear understanding of the problem, the proposed solutions were all over the map. Here are some of the suggestions that turned up quickly:

> *"You might need to analyze the system tables."*

"There are a few known bugs with DD queries in 10g. Few of them involved the CDEF$ table so you might want to do a search for that and/or for the particular views you're having trouble with. The solution was to delete statistics from the involved tables and then lock the stats."

"Remove any initialization parameters set for Oracle 9i."

"Apply application vendor suggestions (like _optimizer_cost_based_transformation=false, NLS_LENGTH_SEMANTICS=CHAR, _gby_hash_aggregation_enabled=false)."

"Disable sub-query unnesting (_UNNEST_SUBQUERY = FALSE)."

"Don't use FIRST_ROWS optimizer goal."

All these might be absolutely valid solutions for different people's own problems. One could very well be the solution the original poster needs, but we don't know which one. A couple of these solutions actually contradict each other (collect vs. delete statistics). These recommendations are based on the previous experience (often quite extensive) of those who proposed them, and they might match well the symptom observed, that "dictionary queries take a long time to return." However, there is one common problem to all of the proposed solutions: the analysis phase is missing. No one has done any analysis or testing to verify the problem, or to verify that their proposed solution even addresses the problem. Everyone is, in essence, guessing.

To show you the magnitude of guesswork and where it leads, here is another example, this time from the OTN Forums:

"My database running on AIX 5.3, oracle getting the version 9.2.0.5.0, after migration it is getting very slow. Kindly provide a solution to tune the database and increase performance."

This is probably an extreme example of ignorance and/or laziness that pushes the author to search for a quick fix solution. Now let's see how this plea for help is being followed up. First of all, a number of people asked for clarification on what is actually running slowly, and for more details about the environment—fair enough. However, there was also a shower of well-meaning advice.

One well-intended bit of advice was:

"You can delete and re-gather dbms stats for your application schemas after the upgrade."

And these were the results after the original poster tried implementing the advice:

"getting same problem
continuously database have lock and the
dbcache hit ratio is 60% only.
total sga size is 20GB
db_cache_size 13gb"

This next proposed solution is a bit better. It's hinting towards actually analyzing the problem:

"From OEM you can view the performance and the SQL statements which are being fired every moment and then find out about missing indexes or tune the SQL."

Then follows advice from someone who is seemingly a guru in Oracle database performance tuning. That advice comes in the form of 11 bullet points. Applied, each of them could fix certain problems, or make performance worse in this particular case. Making 11 changes and hoping that one of them fixes a problem is not an optimal approach.

Following is yet another suggestion on configuring asynchronous input/output. It could be a valid path in certain cases, but is it valid in this case?

"Have you got asynch I/O configured on AIX?"

The original poster did enable ASYNC I/O and, of course, it didn't help.

The list of randomly proposed solutions went on and on, and the discussion ended up sidetracked far from solving the original problem. Did the original poster ever get the help s/he was after? I don't know. It didn't appear so.

Why Do We Guess?

The most trivial factor leading to guess-based solutions is laziness, a natural human quality. Why embark on a path of investigation when there is a chance that one of the proposed solutions just fixes the problem? The reason is that while a random fix might actually work for some commonplace issues, it introduces a significant chance of fixing the wrong problem, making things worse, or simply hiding the symptoms. Fortunately, in my professional life I've met very few DBAs who are too lazy to analyze a problem. Most prefer to troubleshoot and fix a problem once and for all.

■ **Note** One can almost argue that taking the time to analyze a problem and implement a true solution is also a way of being lazy, because you save all the work of randomly guessing and trying. But it's a "good" lazy.

Unfortunately, there are other factors besides laziness. Often, companies are not investing enough to provide their support engineers with the right tools and knowledge. Spending money on a performance-tuning tool or on the Diagnostic Pack option will pay back in spades when it comes to, say, troubleshooting an issue that causes online orders to time out, or that causes the factory floor to be idle. The same goes with investing in education and hiring qualified performance consultants or services. The investment pays back the next time a problem is experienced with a business-critical function, and such problems are solved many times more quickly than otherwise.

An Example from Cycling

by Jonathan Gennick

The idea of "knowing" something rather than "guessing" it is near and dear to my heart. The concept applies beyond just Oracle. I do an increasing amount of bicycle maintenance for myself and for friends. A common problem that I encounter is the bent derailleur hanger. To diagnose that problem reliably, one