THE EXPERT'S VOICE® IN ORACLE

IOUG
independent oracle users group
share your experience

SECOND EDITION

Expert Oracle Exadata

ORACLE'S HIGHEST PERFORMANCE WITH PETABYTE SCALABILITY

Martin Bach, Karl Arao, Andy Colvin, Frits Hoogland, Kerry Osborne, Randy Johnson, and Tanel Poder

Apress®

Expert Oracle Exadata

Second Edition

Martin Bach
Karl Arao
Andy Colvin
Frits Hoogland
Kerry Osborne
Randy Johnson
Tanel Poder

IOUG independent oracle users group

Apress®

Expert Oracle Exadata

ISBN-13 (pbk): 978-1-4302-6241-1

ISBN-13 (electronic): 978-1-4302-6242-8

Managing Director: Welmoed Spahr
Lead Editor: Jonathan Gennick
Development Editor: Douglas Pundick
Technical Reviewer: Frits Hoogland
Editorial Board: Steve Anglin, Louise Corrigan, Jim DeWolf, Jonathan Gennick, Robert Hutchinson, Michelle Lowman, James Markham, Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing, Steve Weiss
Coordinating Editor: Jill Balzano
Copy Editor: Ann Dickson
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 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. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@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/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com/9781430262411. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

About IOUG Press

IOUG Press *is a joint effort by the* ***Independent Oracle Users Group (the IOUG)*** *and* ***Apress*** *to deliver some of the highest-quality content possible on Oracle Database and related topics. The IOUG is the world's leading, independent organization for professional users of Oracle products. Apress is a leading, independent technical publisher known for developing high-quality, no-fluff content for serious technology professionals. The IOUG and Apress have joined forces in IOUG Press to provide the best content and publishing opportunities to working professionals who use Oracle products.*

Our shared goals include:

- Developing content with excellence
- Helping working professionals to succeed
- Providing authoring and reviewing opportunities
- Networking and raising the profiles of authors and readers

To learn more about Apress, visit our website at **www.apress.com**. Follow the link for IOUG Press to see the great content that is now available on a wide range of topics that matter to those in Oracle's technology sphere.

Visit **www.ioug.org** to learn more about the Independent Oracle Users Group and its mission. Consider joining if you haven't already. Review the many benefits at www.ioug.org/join. Become a member. Get involved with peers. Boost your career.

Contents at a Glance

About the Authors xxi

Acknowledgments xxiii

Introduction xxv

Chapter 1: What Is Exadata? 1

Chapter 2: Offloading / Smart Scan 21

Chapter 3: Hybrid Columnar Compression 67

Chapter 4: Storage Indexes 121

Chapter 5: Exadata Smart Flash Cache 141

Chapter 6: Exadata Parallel Operations 177

Chapter 7: Resource Management 209

Chapter 8: Configuring Exadata 251

Chapter 9: Recovering Exadata 303

Chapter 10: Exadata Wait Events 341

Chapter 11: Exadata Performance Metrics 371

Chapter 12: Monitoring Exadata Performance 423

Chapter 13: Migrating to Exadata 463

Chapter 14: Storage Layout 507

Chapter 15: Compute Node Layout 537

Chapter 16: Patching Exadata 547

Chapter 17: Unlearning Some Things We Thought We Knew 571

Appendix A: CELLCLI and DCLI 599

Appendix B: Online Exadata Resources 613

Appendix C: Diagnostic Scripts 617

Appendix D: exachk 621

Index 631

Contents

About the Authors xxi

Acknowledgments xxiii

Introduction xxv

Chapter 1: What Is Exadata? 1

An Overview of Exadata 1

History of Exadata 2

Alternative Views of What Exadata Is 4

Data Warehouse Appliance 4

OLTP Machine 5

Consolidation Platform 5

Configuration Options 6

Exadata Database Machine X5-2 6

Exadata Database Machine X4-8 7

Exadata Storage Expansion Rack X5-2 7

Hardware Components 9

Operating Systems 10

Database Servers 10

Storage Servers 10

InfiniBand 10

Flash Cache 11

Disks 11

Bits and Pieces 11

Software Components 11
Database Server Software 11
Storage Server Software 14
Software Architecture 16
Summary 20
Chapter 2: Offloading / Smart Scan 21
Why Offloading Is Important 21
What Offloading Includes 26
Column Projection 27
Predicate Filtering 31
Storage Indexes and Zone Maps 33
Simple Joins (Bloom Filters) 35
Function Offloading 38
Compression/Decompression 41
Encryption/Decryption 42
Virtual Columns 42
Support for LOB offloading 45
JSON Support and Offloading 46
Data Mining Model Scoring 47
Non-Smart Scan Offloading 48
Smart Scan Prerequisites 49
Full Scans 49
Direct Path Reads 50
Exadata Storage 53
Smart Scan Disablers 54
Simply Unavailable 54
Reverting to Block Shipping 55
Skipping Some Offloading 56
Skipping Offloading silently 56

How to Verify That Smart Scan Is Happening 57
10046 Trace 57
Session Performance Statistics 58
Offload Eligible Bytes 59
SQL Monitoring 63
Parameters 65
Summary 66
Chapter 3: Hybrid Columnar Compression 67
Oracle Storage Review 67
Disassembling the Oracle Block 70
Compression Mechanics 73
BASIC Compression 73
OLTP Compression 74
Hybrid Columnar Compression 76
HCC Internals 80
What Happens When You Create a HCC Compressed Table? 83
HCC Performance 86
Load Performance 86
Query Performance 87
DML Performance 90
Expected Compression Ratios 97
Compression Advisor 97
Real-World Examples 99
Restrictions/Challenges 105
Moving Data to a Non-Exadata Platform 105
Disabling Serial Direct Path Reads 106
Locking Issues 106
Single Row Access 110

Common Usage Scenarios 111
Automatic Data Optimization 112
Example Use Cases for ADO 114
Summary 120
Chapter 4: Storage Indexes 121
Structure 121
Monitoring Storage Indexes 122
Database Statistics 123
Tracing 124
Monitoring Wrap-Up 126
Controlling Storage Indexes 126
_kcfis_storageidx_disabled 127
_kcfis_storageidx_diag_mode 127
_cell_storidx_mode 127
_cell_storidx_minmax_enabled 128
Storage Software Parameters 128
Behavior 129
Performance 130
Special Optimization for Nulls 132
Physical Distribution of Values 133
Potential Issues 134
Incorrect Results 134
Moving Target 135
Partition Size 138
Incompatible Coding Techniques 138
Summary 139
Chapter 5: Exadata Smart Flash Cache 141
Hardware 142
Flash Memory in Exadata X4-2 Storage Servers 142
Flash Memory in Exadata X5-2 Storage Servers 144

Flash Cache vs. Flash Disk 145
Using Flash Memory as Cache 146
Mixed Workload and OLTP Optimizations 150
Using Flash Memory for Database Logging 151
Using Flash Memory to Accelerate Writes 153
Miscellaneous Other WBFC-related Optimizations 155
How ESFC and ESFL Are Created 156
Enabling the Write-back Flash Cache 158
Flash Cache Compression 162
Controlling ESFC Usage 163
Monitoring 164
At the Storage Layer 164
At the Database Layer 170
Summary 176
Chapter 6: Exadata Parallel Operations 177
Parameters 177
Parallelization at the Storage Tier 180
Auto DOP 180
Operation and Configuration 181
I/O Calibration 184
Auto DOP Wrap-Up 186
Parallel Statement Queueing 186
The Old Way 187
The New Way 187
Controlling Parallel Queueing 190
Parallel Statement Queueing Wrap-Up 197
In-Memory Parallel Execution 197
Troubleshooting Parallel Execution 206
Summary 208

Chapter 7: Resource Management 209

Consolidation 210

Types of Database Consolidation 210

Instance Caging 211

Configuring Instance Caging 212

Setting CPU_COUNT 213

Instance Caging Usage and Results 213

Instance Caging and Multitenancy 214

Over-Provisioning 214

Binding Instances to Specific CPUs Using Cgroups 215

Installation and Configuration of Cgroups 215

Oracle 12c THREADED_EXECUTION 217

Managing PGA Memory 218

Database Resource Manager 221

Creating a CDB Resource Plan 222

Creating a (Pluggable) Database Resource Plan 224

Using the Scheduler to Change the Resource Plan 227

The Wait Event: resmgr: cpu quantum 228

Where to Go from Here 228

Resource Mapping Priorities 229

Resource Limiting 229

Other Limiting Parameters 230

Consumer Group Mappings Using ORACLE_FUNCTION 231

Monitoring the Resource Manager 232

Resource Manager Views 233

I/O Resource Manager 234

IORM Methods 235

How IORM Works 236

IORM Architecture 236

IORM Objective 238

Configuring Interdatabase IORM 238

Category IORM 241
I/O Resource Manager and Pluggable Databases 243
I/O Resource Manager Profiles 243
Resource Management Directives Matrix 244
IORM Monitoring and Metrics 245
Summary 250
Chapter 8: Configuring Exadata 251
Exadata Network Components 251
The Management Network 252
The Client Access Network 252
The Private Network 252
About the Configuration Process 254
Configuring Exadata 256
Step 1: Gathering Installation Requirements 256
Step 2: Run Oracle Exadata Deployment Assistant 257
Step 3: Create Network VLANs and DNS Entries for Hostnames 285
Step 4: Run CheckIP to Verify Network Readiness 285
Step 5: Run Cables and Power to Exadata Racks 288
Step 6: Perform Hardware Installation 289
Step 7: Stage OneCommand Files and Oracle Software 289
Step 8: Configure the Operating System 291
Step 9: Run OneCommand 294
Upgrading Exadata 297
Creating a New RAC Cluster 298
Upgrading the Existing Cluster 299
Summary 301
Chapter 9: Recovering Exadata 303
Exadata Diagnostic Tools 303
Sun Diagnostics: sundiag.sh 304
Cell Alerts 307

Backing Up Exadata 308
Backing Up the Database Servers 308
Backing Up the Storage Cell 312
Backing Up the Database 316
Disk-Based Backups 316
Tape-Based Backups 317
Backup from Standby Database 318
Exadata Optimizations for RMAN 318
Recovering Exadata 319
Restoring the Database Server 320
Recovering the Storage Cell 323
Summary 339
Chapter 10: Exadata Wait Events 341
Events Specific to Exadata 342
The “cell” Events 343
Plan Steps That Trigger Events 344
Exadata Wait Events in the User I/O Class 346
cell smart table scan 346
cell smart index scan 350
cell single block physical read 352
cell multiblock physical read 354
cell list of blocks physical read 355
cell smart file creation 356
cell statistics gather 356
Minor Events in the User/IO Class 357
Exadata Wait Events in the System I/O Class 358
cell smart incremental backup 358
cell smart restore from backup 360
Exadata Wait Events in the Other and Idle Classes 361
cell smart flash unkeep 361
Event Meaning 362

Non-Exadata-Specific Events 363
direct path read 363
Enq: KO—fast object checkpoint 364
reliable message 365
Resource Manager Events 366
resmgr:become active 366
resmgr:cpu quantum 368
resmgr:pq queued 369
Summary 370
Chapter 11: Exadata Performance Metrics 371
Measuring Exadata's Performance Metrics 371
Revisiting the Prerequisites for Exadata Smart Scans 374
Exadata Smart Scan Performance 374
Understanding Exadata Smart Scan Metrics and Performance Counters 378
Exadata Dynamic Performance Counters 378
When and How to Use Performance Counters 379
The Meaning and Explanation of Exadata Performance Counters 383
Performance Counter Reference for a Selected Subset 386
Understanding SQL Statement Performance 411
Querying cellsrv Internal Processing Statistics 414
The V$CELL Family of Views 415
The cellsrvstat utility 419
Summary 421
Chapter 12: Monitoring Exadata Performance 423
A Systematic Approach 423
Monitoring SQL Statement Response Time 424
Monitoring SQL Statements with Real-Time SQL Monitoring Reports 425
Monitoring SQL Statements Using V$SQL and V$SQLSTATS 439

Monitoring the Storage Cell Layer 441
Accessing Cell Metrics in the Cell Layer Using CellCLI 442
Accessing Cell Metrics Using the Enterprise Manager Exadata Storage Server Plug-In 443
Which Cell Metrics to Use? 449
Monitoring Exadata Storage Cell OS-Level Metrics 450
Summary 461
Chapter 13: Migrating to Exadata 463
Migration Strategies 464
Logical Migration 465
Extract and Load 466
Copying Data over a Database Link 472
Replication-Based Migration 486
Logical Migration Wrap Up 492
Physical Migration 492
Backup and Restore 493
Full Backup and Restore 493
Incremental Backup 495
Transportable Tablespaces 497
Cross-Platform TTS with Incremental Backups 500
Physical Standby 503
Wrap Up Physical Migration Section 505
Summary 506
Chapter 14: Storage Layout 507
Exadata Disk Architecture 507
Failure Groups 509
Grid Disks 512
Storage Allocation 514
Creating Grid Disks 518
Creating Grid Disks 519
Grid Disk Sizing 520
Creating FlashDisk-Based Grid Disks 524

Storage Strategies 525
Configuration Options 525
Isolating Storage Cell Access 526
Cell Security 528
Cell Security Terminology 529
Cell Security Best Practices 529
Configuring ASM-Scoped Security 530
Configuring Database-Scoped Security 531
Removing Cell Security 534
Summary 536
Chapter 15: Compute Node Layout 537
Provisioning Considerations 538
Non-RAC Configuration 539
Split-Rack Clusters 541
Typical Exadata Configuration 543
Multi-Rack Clusters 544
Summary 546
Chapter 16: Patching Exadata 547
Types of Exadata Patches 548
Quarterly Database Patch for Exadata 549
Applying a QDPE in Place 550
Applying a QDPE by Cloning Homes 553
Exadata Storage Server Patches 556
Applying an Exadata Storage Server Patch 559
Upgrading Compute Nodes 565
Upgrading InfiniBand Switches 568
Applying Patches to Standby Systems 569
Summary 570

Chapter 17: Unlearning Some Things We Thought We Knew 571
A Tale of Two Systems 571
OLTP-Oriented Workloads 572
Exadata Smart Flash Cache (ESFC) 572
Scalability 573
Write-Intensive OLTP Workloads 573
DW-Oriented Workloads 574
Enabling Smart Scans 574
Things That Can Cripple Smart Scans 576
Other Things to Keep in Mind 583
Mixed Workloads 590
To Index or Not to Index? 591
The Optimizer Doesn't Know 594
Using Resource Manager 598
Summary 598
Appendix A: CELLCLI and DCLI 599
An Introduction to CellCLI 599
Invoking cellcli 600
Getting Familiar with cellcli 602
Sending Commands from the Operating System 607
Using cellcli XML Output in the Database 607
Configuring and Managing the Storage Cell 609
An Introduction to dcli 610
Summary 612
Appendix B: Online Exadata Resources 613
My Oracle Support Notes 613
The Authors' Blogs 615

Appendix C: Diagnostic Scripts 617

Appendix D: exachk 621

An Introduction to exachk 621

Running exachk 622

Saving Passwords for exachk 625

Automating exachk Executions 627

Summary 629

Index 631

About the Authors

Martin Bach is an Oracle consultant and overall technical enthusiast. He specialized in the Oracle DBMS in 2001, with his main interests in high availability and disaster recovery solutions for mission critical 24x7 systems. For a good few years now, Martin has had a lot of fun exploring many different types of Engineered Systems from an infrastructure and performance point of view. He is an Oracle Certified Master, Oracle Ace Director, and OakTable member. Previous publications include co-authoring *Pro Oracle Database RAC 11g on Linux* and *Expert Consolidation in Oracle Database 12c*. In addition, Martin maintains a weblog on `http://martincarstenbach.wordpress.com` where additional research about this book and other topics can be found. When he expresses his thoughts in tweets, he uses the twitter handle `@MartinDBA`.

Andy Colvin is an Oracle consultant who specializes in infrastructure management. He began working in IT in 1999 as a network and systems administrator, supporting several Oracle environments. Andy joined Enkitec in 2006 and began to focus on Oracle Engineered Systems in 2010. In 2012, Andy was awarded Oracle ACE status for his online contributions, mainly found at `http://oracle-ninja.com`. When not patching or configuring an Exadata, Andy still enjoys working with networks and various operating systems. When he has something worth saying in less than 140 characters, he tweets at `@acolvin`.

Frits Hoogland is an IT professional specializing in Oracle database performance and internals. Frits frequently presents on Oracle technical topics at conferences around the world. In 2009, he received an Oracle ACE award from the Oracle Technology Network and a year later became an Oracle ACE Director. In 2010, he joined the OakTable Network. In addition to developing his Oracle expertise, Frits investigates modern operating systems. Frits currently works at the Accenture Enkitec Group. Previous involvement with publications includes being the technical reviewer for *Expert Oracle Database Architecture, Expert Consolidation in Oracle Database 12c, Expert Oracle SQL, Expert Oracle Enterprise Manager*, and *Practical Oracle Database Appliance*. Frits keeps a weblog at `http://fritshoogland.wordpress.com` where additional research can be found.

Karl Arao currently works for Accenture Enkitec Group and has nine years of Oracle database consulting experience across a broad range of industries. He specializes in Performance, Resource Management, Capacity Planning, Consolidation, and Sizing. Prior to this, he was a Solutions Architect and an R&D guy. Karl is a proud member of OCP-DBA, RHCE, Oracle ACE, and the OakTable Network. He is a frequent speaker at Oracle conferences and shares his experiences, adventures, and discoveries in his blog (`karlarao.wordpress.com`), tweets at @karlarao, and owns a wiki (`karlarao.tiddlyspot.com`) where he shares his quick guides and documentations on technologies.

The foregoing are the authors who've prepared this second edition. Also having content in this book are the first-edition authors: Kerry Osborne, Randy Johnson, and Tanel Poder. While not contributing directly to this second edition, their support and guidance have been essential to keeping this work alive.

Acknowledgments

The book you are holding in your hands, be it in electronic or printed form, has been a fair bit of work for everyone involved. The agile development on the Exadata platform was in many ways a blessing and a curse—a blessing because you could appreciate the improvements introduced with every release, and a curse because the new features should be in the book, causing more work.... This project has been one of the longest I have been involved in, and I would like to thank my family (again!) for letting me spend a lot of time researching and writing for what turned out to be a long period of time. I'll try and make up for it, promise! Personally participating in the organization and writing were hugely rewarding as they allowed me to delve into the depths of the Exadata implementation. It is probably true that only in teaching and writing do you get the most comprehensive understanding of the subject you cover. How often did I think I knew what I was about to write, only to find out I had no clue. But, thankfully, I wasn't on my own. I wouldn't have been able to do this without the support from my colleagues and my friends, who proved inspirational (sometimes even unknowingly). There are simply too many to mention on this page—I'm sure you know whom I mean when you read this paragraph. A big "thank you" to you all.

—Martin Bach

First and foremost, I would like to thank the authors of the first edition for giving us great source material to work with. To Kerry, Randy, and Tanel—for all of the times that we have heard about how great the first edition was, I hope we did it justice. This has been a long journey to say the least. It has been great to work with Martin, Frits, and Karl throughout. As Frits and Martin mentioned, this took a significant amount of time away from other priorities, mainly my family. I truly appreciate their willingness to let me spend those long nights locked away, trying to get pen to paper and work out the thoughts in my brain. This has been a revealing experience, and I have learned a lot during the writing process. Keeping up with an ever-changing platform can make for plenty of rewrites during the life of the project! I enjoyed the time spent writing this, and I hope that you are able to read this book and learn something new.

—Andy Colvin

Being a writer for a book has been a learning cycle for me, as this is my first time for actually writing, instead of "just" commenting on the work of others. I started off doing one chapter, which would have been only a modest amount of work and time, but this one chapter eventually became three chapters. Of course, having been the technical reviewer for the previous edition, I served the technical reviewer of all the chapters I didn't write. Being both a writer and technical reviewer meant I spent a tremendous amount of time creating this book. I would like to thank my family for letting me spend the countless hours writing, reviewing, researching, testing, and so on. Exactly as Martin put it, a huge part of this book came into existence because of the collaboration of colleagues and friends, in all kinds of forms. Thank you.

—Frits Hoogland

First of all, I would like to thank my parents, Denis and Nenita, and my brother, Kevin. Without you, I wouldn't be striving to be the best that I can be. I love you. To the Arao and Agustin families, my friends, and loved ones—thank you for providing support and fun moments, while keeping me sane as I wrote my chapters. To Kerry Osborne, Veronica Stigers, and Martin Paynter—thank you for always believing in me and for all the interesting challenges and rare opportunities you have given. To Dinah Salonga and SQL*Wizard Family—thank you for all the mentorship and friendship and for exposing me to a lot of difficult customer situations that helped me become a solid DBA at a young age. I will never forget all the fun sleepless nights. And thanks especially to Jonathan Gennick and the Apress team for all your patience and support. Yes, we did it! Thanks to all who helped me on my research and your valuable input. Finally, I would like to express my appreciation for the great conversations I had with the like-minded people from the Oracle community, the conferences, oracle-l mailing list, the OakTable Network, and the Oracle ACE program. Thank you for all the inspiration, learning, shared ideas, friendship, and help. Great ideas are built on the ideas of great minds and great ideas of the past. Let's keep the community spirit high all the time.

—Karl Arao

Introduction

Thank you very much for buying the second edition of *Expert Oracle Exadata*. Us current authors have been standing on the shoulders of giants while putting this together. Whenever writing a second edition of a successful book, the authors face the pressure of creating at least as good, if not better, edition than the first edition was. And good it was, the first edition. We hope that we have been able to provide you, dear reader, with a suitable introduction to Exadata. In fact, our hope is to give you enough information to get started with Exadata. It is not uncommon to find database administrators in situations where they have been introduced to Exadata. only to ask the question, "Now what?" We have tried to structure the book to help you answer this question. You will read about what Exadata is before diving into the various optimizations that make it so unique in the world of Oracle database processing hardware. While some of the material, particularly in the earlier chapters, paints a broad picture, we gradually go into a lot of detail. Access to an Exadata development system can help you a lot in understanding the more advanced material. We have tried very hard to make it possible for you to follow along, but please bear in mind that the Exadata platform is not static at all; new releases in hardware and software can change the documented outcome of commands and SQL statements. We will try to address major differences on our web site, `http://www.expertoracleexadata.com/` and our personal blogs listed in Appendix B.

Note that we have used various undocumented underscore parameters and features to demonstrate how various pieces of the software work. Do not take this as a recommended approach for managing a production system! In fact, there is usually no reason to deviate from the defaults. Setting underscore parameters is allowed only with the explicit blessings from Oracle Support and as the result of a recommendation as part of a service request you raised. Remember that we have had access to a number of systems that we could tear apart with little worry about the consequences that resulted from our actions. This gave us a huge advantage in our investigations on how Exadata works across various hardware generations.

The Intended Audience

This book assumes that you are already familiar with Oracle. We do not go into a lot of detail explaining how Oracle works except as it relates to the Exadata platform. This means that we have made some assumptions about the readers' knowledge. We do not assume, for instance, that you are an expert in Oracle performance tuning, but we expect that you are proficient writing SQL statements and have a good understanding of the Oracle architecture. Since Exadata is a hardware and software platform, you will inevitably see references to Linux administration in some of the chapters more closely related to the hardware. Do not be intimidated—as an Exadata administrator, there are only a handful of commands that you need to know in day-to-day managing of the platform.

A Moving Target

We had this exact same section in the introduction of the first edition of this book, and the message is still the same, even after all these years. What keeps us amazed to this day is the pace of development of the Exadata platform. It is not only hardware that evolves and keeps up with the development of new technologies, but also the software that is constantly pushing the limits of what is possible. A new software release does not require you to upgrade the hardware. Except for the very first Exadata system, the current Exadata software version is compatible with every hardware generation.

The changes mentioned in the previous paragraph include substantial additions of new functionality, visible in Appendix A in the *Exadata Database Machine System Overview*. As you can imagine, trying to keep track of what Oracle released at a rapid pace was the most difficult part of the project. Every chapter had to go through multiple revisions when new hardware and software was released. The latest version we try to cover in this book is Oracle 12.1.0.2.2 RDBMS with cell software 12.1.2.1.x. Unlike the first edition of this book, which came out when Oracle 11.2.0.2 was current, there are quite a few releases now that Exadata supports technically. From an Oracle Support point of view, right now you should probably be in a migration phase to Oracle 12c. This is one of the reasons we gave the latest RDBMS release so much space in the book, even though many users are yet to migrate to it. Another consideration while writing this book was that we had to be quite careful to cite the correct version when a new feature was introduced. If you only have just started with Exadata, you might find the release numbers confusing; however, once you have your first few weeks of Exadata administration under your belt, you will find that quoting Exadata cell software releases becomes second nature.

The way Exadata evolves will undoubtedly make some of the book's contents obsolete, so if you observe differences between what is covered in this book and what you see it is probably due to version differences. Nevertheless, we welcome your feedback and will address any inconsistencies that you find.

Many Thanks to Everyone Who Helped!

We have had a great deal of support from a number of people on this project. Having our official technical reviewer take on writing a few chapters is almost an occurrence of history repeating itself. Writers and reviewers swapped roles to reply to the question, "Quis custodiet ipsos custodes?" We are also very grateful for everyone at Oracle who may have even known us from the first edition of this book and helped us overcome the stumbling blocks along the way. Finally, we want to give a big "thank you" to everyone at @Enkitec who helped keep the machines up and running, patched when a new release came out, and troubleshot when something seemed broken. The list of people is really long, so we won't be able to mention everyone by name. However, it is fair to say that if you worked at @Enkitec while this book was being written, you almost certainly contributed—thank you.

The first book helped generate interest in the second edition, and we have published some research that was too comprehensive on our personal blogs and web sites, prompting e-mail, twitter, and comments to start flying our way once an article went online. The same is true for the feedback we had with the Alpha Programme; without the community's feedback, this book would probably be less complete, and we would like to explicitly thank you for your comments.

And last, but not least, we would like to give a very special "thank you" to the authors of the first edition of the book, who allowed us to update what they wrote. Kerry, Tanel, and Randy have been instrumental in understanding the intended message of the chapters as well as chapter layout and tests. Without you, we wouldn't have been able to finish the chapters while maintaining the spirit of the first book.

Who Wrote What

Following the tradition set in the first edition, we would like to list which of us worked on each chapter. The authors of the second edition (in alphabetical order) are Karl Arao, Martin Bach, Andy Colvin, and Frits Hoogland. It really was a team effort between all of us involved, and we cannot even think about counting the hours of useful conversations and instant messages exchanged among all of us to bounce off ideas and make sure that we did not overlap contents in our chapters.

Karl: contributions to Chapters 5, 6, 7, 12

Andy: Chapters 1, 8, 9, 14, 15, 16, Appendix D

Martin: Chapters 2, 3, 5, 10, 11, 12, 13, 17, Appendices A, B, C

Frits: Chapters 4, 6, 7

Have Fun!

Writing the book was, for the most part, fun for all of us—especially when we knew about a complex problem, but had trouble reproducing a situation allowing us to research it. The moment the experiment came to a successful conclusion, the moment when we had all the output and steps to reproduce it recorded in our log files, was very often a moment of great joy and also relief. We hope his book provides a platform from which you can build your own knowledge. Although having spent a lot of time with both Exadata and Oracle Database 12c, there are still things we learn every day. Somehow it still feels we are only scratching the surface, still.

CHAPTER 1

What Is Exadata?

No doubt, you already have a pretty good idea what Exadata is or you wouldn't be holding this book in your hands. In our view, it is a preconfigured combination of hardware and software that provides a platform for running Oracle Database (either version 11g Release 2 or version 12c Release 1 as of this writing). Since the Exadata Database Machine includes a storage subsystem, different software has been developed to run at the storage layer. This has allowed Oracle product development to do some things that are just not possible on other platforms. In fact, Exadata really began its life as a storage system. If you talk to people involved in the development of the product, you will commonly hear them refer the storage component as Exadata or SAGE (Storage Appliance for the Grid Environment), which was the code name for the project.

Exadata was originally designed to address the most common bottleneck with very large databases—the inability to move sufficiently large volumes of data from the disk storage system to the database server(s). Oracle has built its business by providing very fast access to data, primarily through the use of intelligent caching technology. As the sizes of databases began to outstrip the ability to cache data effectively using these techniques, Oracle began to look at ways to eliminate the bottleneck between the storage tier and the database tier. The solution the developers came up with was a combination of hardware and software. If you think about it, there are two approaches to minimize this bottleneck. The first is to make the pipe between the database and storage bigger. While there are many components involved and it's a bit of an oversimplification, you can think of InfiniBand as that bigger pipe. The second way to minimize the bottleneck is to reduce the amount of data that needs to be transferred. This they did with Smart Scans. The combination of the two has provided a very successful solution to the problem. But make no mistake—reducing the volume of data flowing between the tiers via Smart Scan is the golden goose.

In this introductory chapter, we will review the components that make up Exadata, both hardware and software. We will also discuss how the parts fit together (the architecture). In addition, we will talk about how the database servers talk to the storage servers. This is handled very differently than on other platforms, so we will spend a fair amount of time covering that topic. We will also provide some historical context. By the end of the chapter, you should have a pretty good feel for how all the pieces fit together and a basic understanding of how Exadata works. The rest of the book will provide the details to fill out the skeleton that is built in this chapter.

An Overview of Exadata

A picture is worth a thousand words, or so the saying goes. Figure 1-1 shows a very high-level view of the parts that make up the Exadata Database Machine.

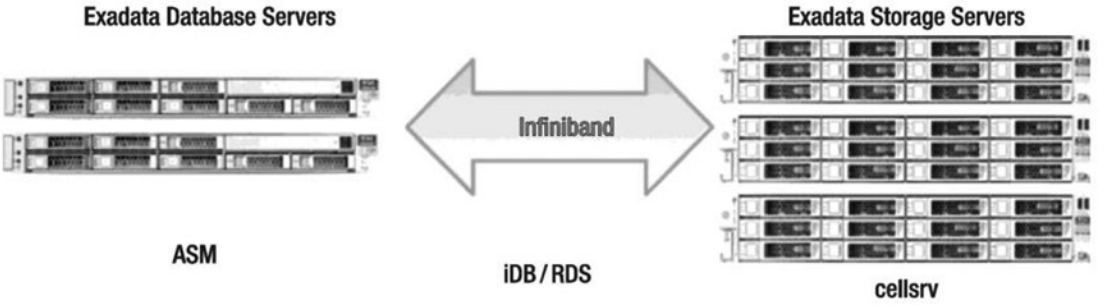

Figure 1-1. *High-level Exadata components*

When considering Exadata, it is helpful to divide the entire system mentally into two parts, the storage layer and the database layer. The layers are connected via an InfiniBand network. InfiniBand provides a low-latency, high-throughput switched fabric communications link. Redundancy is provided through multiple switches and links. The database layer is made up of multiple Sun servers running standard Oracle 11g or 12c software. The servers are generally configured in one or more Real Application Clusters (RAC), although RAC is not actually required. The database servers use Automatic Storage Management (ASM) to access the storage. ASM is required even if the databases are not configured to use RAC. The storage layer also consists of multiple Sun x86 servers. Each storage server contains 12 disk drives or 8 flash drives and runs the Oracle storage server software (`cellsrv`). Communication between the layers is accomplished via iDB, which is a network-based protocol that is implemented using InfiniBand. iDB is used to send requests for data along with metadata about the request (including predicates) to `cellsrv`. In certain situations, `cellsrv` is able to use the metadata to process the data before sending results back to the database layer. When `cellsrv` is able to do this, it is called a Smart Scan and generally results in a significant decrease in the volume of data that needs to be transmitted back to the database layer. When Smart Scans are not possible, `cellsrv` returns the entire Oracle block(s). Note that iDB uses the RDS protocol, which is a low-latency, InfiniBand-specific protocol. In certain cases, the Oracle software can set up remote direct memory access (RDMA) over RDS, which bypasses doing system calls to accomplish low-latency, process-to-process communication across the InfiniBand network.

History of Exadata

Exadata has undergone a number of significant changes since its initial release in late 2008. In fact, one of the more difficult parts of writing this book has been keeping up with the changes in the platform during the project. Following is a brief review of the product's lineage and how it has changed over time:

> **V1:** The first Exadata was released in late 2008. It was labeled as V1 and was a combination of HP hardware and Oracle software. The architecture was similar to the current X5 version, with the exception of Flash, which was added to the V2 version. Exadata V1 was marketed exclusively as a data warehouse platform. The product was interesting but not widely adopted. It also suffered from issues resulting from overheating. The commonly heard description was that you could fry eggs on top of the cabinet. Many of the original V1 customers replaced their V1s with V2s or X2-2s.

V2: The second version of Exadata was announced at Open World in 2009. This version resulted from a partnership between Sun and Oracle. By the time the announcement was made, Oracle was already in the process of attempting to acquire Sun Microsystems. Many of the components were upgraded to bigger or faster versions, but the biggest difference was the addition of a significant amount of solid state-based storage. The storage cells were enhanced with 384G of Exadata Smart Flash Cache. The software was also enhanced to take advantage of the new cache. This addition allowed Oracle to market the platform as more than a Data Warehouse platform, opening up a significantly larger market.

X2: The third edition of Exadata, announced at Oracle Open World in 2010, was named the X2. Actually, there were two distinct versions of the X2. The X2-2 followed the same basic blueprint as the V2, with up to eight dual-socket database servers. The CPUs were upgraded to hex-core models, where the V2s had used quad-core CPUs. The other X2 model was named the X2-8. It broke the small 1U database server model by introducing larger database servers with 8×8 core CPUs and a large 1TB memory footprint. The X2-8 was marketed as a more robust platform for large OLTP or mixed workload systems due primarily to the larger number of CPU cores and the larger memory footprint. In 2011, Oracle changed the hardware in the X2-8 to 8x10-core CPUs and 2TB of memory per node. For customers that needed additional storage, storage expansion racks (racks full of storage servers) were introduced. In January 2012, Oracle increased the size of the high-capacity disks from 2TB to 3TB.

X3: In 2012, Oracle announced the Exadata X3. The X3 was the natural progression of the hardware included in the X2 series. Compute node updates included eight-core Intel Sandy Bridge CPUs and increased memory, up to 256GB per server (although it originally was equipped with 128GB per server for a short time). Storage servers saw upgrades to CPUs and memory, and flash storage increased to 1.6TB per server. The X3-2 family also introduced a new size—the eighth rack. X3-8 racks saw the same improvements in the storage servers, but the compute nodes in X3-8 racks are the same as their X2-8 counterparts.

X4: Oracle released the Exadata X4 in 2013. It followed the traditional new features: processing increased to 2x12 core CPUs, the ability to upgrade to 512GB of memory in a compute node was added, and flash and disk storage increased. The X4-2 also saw a new model of high-capacity disk, trading out the 600GB, 15,000 RPM disks for 1.2TB, 10,000 RPM disks. These disks were a smaller form factor (2.5" vs 3.5"). The other notable change with the X4-2 was the introduction of an active/active InfiniBand network connection. On the X4-2, Oracle broke the bonded connection and utilized each InfiniBand port independently. This allowed for increased throughput across the InfiniBand fabric.

X5: In early 2015, Oracle announced the sixth generation of Exadata, the X5-2. The X5-2 was a dramatic change in the platform, removing the high-performance disk option in favor of an all-flash, NVMe (Non-Volatile Memory Express) model. High-capacity disk sizes were not changed, leaving them at 4TB per disk. Once again, the size of the flash cards doubled, this time to 6.4TB per storage server. Memory stayed consistent with a base of 256GB, upgradeable to 768GB, and the core count increased to 18 cores per socket. Finally, the requirement to purchase racks in predefined sizes was removed. The X5-2 rack could be purchased with any configuration required—a base rack begins with two compute nodes and

three storage servers. Beyond that, any combination of compute and storage servers can be used within the rack. This removed discussions around Exadata configurations being "balanced" based on the workload. As was seen by many deployments before the X5, every workload is a little bit different and has different needs for compute and storage.

Alternative Views of What Exadata Is

We have already given you a rather bland description of how we view Exadata. However, like the well-known tale of the blind men describing an elephant, there are many conflicting perceptions about the nature of Exadata. We will cover a few of the common descriptions in this section.

Data Warehouse Appliance

Occasionally, Exadata is described as a *data warehouse appliance (DW Appliance)*. While Oracle has attempted to keep Exadata from being pigeonholed into this category, the description is closer to the truth than you might initially think. It is, in fact, a tightly integrated stack of hardware and software that Oracle expects you to run without a lot of changes. This is directly in line with the common understanding of a DW Appliance. However, the very nature of the Oracle database means that it is extremely configurable. This flies in the face of the typical DW Appliance, which typically does not have a lot of knobs to turn. However, there are several common characteristics that are shared between DW Appliances and Exadata:

Exceptional Performance: The most recognizable characteristic of Exadata and DW Appliances in general is that they are optimized for data warehouse type queries.

Fast Deployment: DW Appliances and Exadata Database Machines can both be deployed very rapidly. Since Exadata comes preconfigured, it can generally be up and running within a week from the time you take delivery. This is in stark contrast to the normal Oracle clustered database deployment scenario, which generally takes several weeks.

Scalability: Both platforms have scalable architectures. With Exadata, upgrading is done in discrete steps. Upgrading from a half-rack configuration to a full rack increases the total disk throughput in lock step with the computing power available on the database servers.

Reduction in TCO: This one may seem a bit strange, since many people think the biggest drawback to Exadata is the high price tag. But the fact is that both DW Appliances and Exadata reduce the overall cost of ownership in many applications. Oddly enough, in Exadata's case, this is partially thanks to a reduction in the number of Oracle database licenses necessary to support a given workload. We have seen several situations where multiple hardware platforms were evaluated for running a company's Oracle application and have ended up costing less to implement and maintain on Exadata than on the other options evaluated.

High Availability: Most DW Appliances provide an architecture that supports at least some degree of *high availability (HA)*. Since Exadata runs standard Oracle 12c or 11g software, all the HA capabilities that Oracle has developed are available out of the box. The hardware is also designed to prevent any single point of failure.