

*Comprehensive, End-to-End Insight into
Oracle's Flagship Database Machine*



Oracle Exadata

EXPERT'S HANDBOOK

Tariq Farooq | Charles Kim | Nitin Vengurlekar
Sridhar Avantsa | Guy Harrison | Syed Jaffar Hussain

Covers
versions **11g**
and **12c**

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Oracle Exadata Expert's Handbook

This page intentionally left blank



Oracle Exadata Expert's Handbook

Tariq Farooq
Charles Kim
Nitin Vengurlekar
Sridhar Avantsa
Guy Harrison
Syed Jaffar Hussain

◆◆ Addison-Wesley

New York • Boston • Indianapolis • San Francisco
Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2015935451

Copyright © 2015 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, 200 Old Tappan Road, Old Tappan, New Jersey 07675, or you may fax your request to (201) 236-3290.

Various screen shots and illustrations of Oracle products are used with permission. Copyright © 1995–2014 Oracle and/or its affiliates. All rights reserved.

ISBN-13: 978-0-321-99260-4

ISBN-10: 0-321-99260-1

Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana

First printing, June 2015



Contents

Preface	xvii
Acknowledgments	xix
About the Authors	xxiii
About the Technical Reviewers and Contributors	xxvii
Chapter 1 360-Degree Overview of Exadata	1
An Exadata Synopsis	2
An Engineered Database Machine	3
How Exadata Changes Your Job Role	4
Oracle Enterprise Manager 12c	4
Hardware Architecture	5
Server Layer—Compute Nodes	5
Shared Storage—Storage Cells	6
Networking Fabric—InfiniBand	6
Power Distribution Units (PDUs)	7
Cisco Switch	7
2u Custom Network Switch Space	7

Software Architecture	7
Real Application Clusters (RAC)	7
Automatic Storage Management (ASM)	8
DB Compute Nodes	9
Storage Cell Software	9
Models and Configuration Options	9
Historical Synopsis	10
The Evolution of Exadata	10
Exadata SuperCluster T4-4	25
Exadata SuperCluster T5-8	27
Exadata SuperCluster M6-32	29
Exadata Storage Expansion Racks	31
Exadata Storage Cells	33
Hardware Progression	33
Examining an Exadata Machine	35
Summary	37
Chapter 2 Real Application Clusters (RAC) in Exadata	39
The Significance of RAC in Exadata	40
An Overview of RAC	41
A Quick Primer on RAC in Exadata	42
How RAC Affects DBAs	42
Setting Up RAC Clusters in Exadata	43
Operational Best Practices	45
Maximum Availability Architecture (MAA)	45
Optimal and Efficient Databases in RAC	46
Managing RAC with OEM 12c	49
Common Utilities and Commands	50
Troubleshooting and Tuning RAC	55
Start with ORAchk	55
Employ the TFA Collector Utility	56
Use the Automatic Diagnostic Repository	56
Check the Alert and Trace Log Files	56
Employ the Three As	56

Check the Private Cluster Interconnect	57
Enable Tracing and Inspect the Trace Logs	57
Cluster Health Monitor	57
Employ Oracle Enterprise Manager 12c	57
Miscellaneous Tools and Utilities	58
Useful Oracle Support Resources	58
Summary	59
Chapter 3 The Secret Sauce: Exadata Storage Cells	61
An Overview of Exadata Storage Server	61
Storage Server Architecture	63
Cell Software Components and Management	64
Configuring Mail Server for Alert Notifications	68
Displaying Cell Server Details	68
Cell Metrics and Alert History	69
Querying Cell Alert History	70
Querying GV\$ Views	71
Storage Architecture and Formulation	72
Disk Architecture in Non-Exadata	74
Disk Architecture in Exadata	74
System Users for Cell Administration	77
Listing Disk Levels	77
Configuring Cell Disks	81
Creating Grid Disks	81
Configuring Flash Grid Disks	81
Creating an ASM Disk Group	82
Managing the Cell Server	82
Troubleshooting the Cell Server	83
SunDiag	83
ExaWatcher	84
Exachk	85
CheckHWnFWProfile	85
Storage Cell Startup and Shutdown	85
Solving Disk Problems	87

	Enforcing Cell Security	89
	Configuring ASM-Scoped Security	90
	Configuring Database-Scoped Security	91
	Exempting Cell Security	92
	Summary	92
Chapter 4	Flash Cache, Smart Scans, and Cell Offloading	93
	Concepts of Exadata Flash Cache	93
	Why Flash Cache Is Necessary	93
	Evolution of Flash Cache in Exadata	94
	Storage Server and Flash Cache	95
	The Exadata Smart Flash Cache Feature	96
	Populating the Flash Cache	97
	Exadata Smart Flash Logging	98
	The Database and Flash Cache	99
	Smart Scans and Cell Offloading	101
	Storage Indexes	107
	Caching Data in the Flash Cache	115
	Summary	120
Chapter 5	Exadata Compression: HCC Demystified	121
	Columnar Storage Models	122
	The PAX Model	124
	Fractured Mirrors	124
	Fine-Grained Hybrids	124
	Oracle Implementation of DSM—Hybrid Columnar Compression	125
	Compression within Oracle Databases	125
	The Concepts of HCC	125
	Compression Ratios	127
	Compression Types and Compression Units	129
	HCC and Performance	131
	Bulk Load Operations	132
	Bulk Read I/O Operations	135
	Small I/O Operations	137

	HCC and DML	140
	HCC and Locking	144
	Practical Uses of HCC	147
	Summary	148
Chapter 6	Oracle Database 12c and Exadata	149
	12c Partitioning Features	149
	Partial Indexes	149
	Partition Index Maintenance	153
	Partition Move	155
	New 12c Optimizer Features	157
	Adaptive Plans	157
	Automatic Re-optimization	159
	Dynamic Adaptive Statistics	159
	Information Lifecycle Management	164
	Application Continuity	167
	Multitenant Architecture	168
	Overview	169
	PDB: A New Consolidation Model	169
	Unplug/Plug Operations	177
	RAC and PDB	178
	Exadata Software Updates	183
	Summary	183
Chapter 7	Exadata Networking: Management and Administration	185
	Exadata Network Components	185
	The Role of the InfiniBand Network	186
	Network Architecture	187
	Network Setup Requirements	188
	Troubleshooting Tools and Utilities	190
	Physical Link Monitoring	190
	Log Files Collection	194
	Integrated Lights Out Manager	195
	OEM Cloud Control 12c	197
	Summary	199

Chapter 8	Backup and Recovery and Data Guard	201
	RMAN Disk-to-Disk Backups	202
	Settings for RMAN Backups on the Exadata	203
	<code>rman2disk</code> Shell Script	204
	<code>rman2disk</code> Template Files	206
	Using <code>rman2disk</code>	206
	Creating RMAN Backups	209
	RMAN Backup Schedule	213
	Container and Pluggable Databases	215
	Data Guard	216
	Patches	217
	Session Data Unit	217
	Bandwidth-Delay Product	218
	Network Queue Size	220
	Disabling TCP Nagle Algorithm	221
	Enabling Network Time Protocol	221
	Block Change Tracking	222
	Fast Recovery Area	222
	Automatic Archive Switch	223
	Parallel Execution Message Size	223
	Database Cache Size	224
	Standby Redo Logs	224
	Force Logging	226
	Flashback Logging	227
	Real-Time Apply	227
	Timeout and Reopen Options	228
	Archive Generation Rate	229
	Standby File Management	231
	Data Guard Standby-First Patching	231
	Active Data Guard	232
	Far Sync	233
	Archive Log Retention Policy	233
	Data Corruptions	234
	Data Guard Instantiation	235

Configuring Data Guard Broker	239
OEM Cloud Control 12c	241
Switchover Considerations	242
Switchover Tracing	243
Guaranteed Restore Point	244
Summary	244
Chapter 9 Managing Exadata with OEM 12c	245
Exadata Targets Discovery	246
Exadata Monitoring Architecture	246
Oracle Exadata Plugins	248
Prerequisite Checks	249
Manual Deployment	249
Exadata Database Machine Discovery	250
Prerequisite Checks	250
Launching Exadata Discovery	250
Post-Discovery Procedure	260
Exadata Components	260
Monitoring and Management	261
Administration	262
Summary	265
Chapter 10 Migrating to Exadata	267
Exadata Implementation Lifecycle	267
Phase I: Architectural Strategy	268
Sizing the Specific Exadata Solution	272
Phase II: Planning and Design	277
Custom versus Third-Party Applications	278
Choosing Exadata Features to Implement	279
Accounting for the Paradigm Change	279
Determining Migration Strategies	280
Phase III: Migration Testing	287
Backup and Recovery Strategy	288
Exadata Monitoring and Alerting	289
Exadata Patching	289

Exadata Migration Best Practices	290
Summary	291
Chapter 11 Upgrading and Patching Exadata and ZFS Storage	
Appliance	293
Planning an Exadata and ZFS Upgrade	294
Patch Release Cycle	296
Quarterly Full Stack Download	297
Patching Tools and Processes	297
OPatch	298
patchmgr	299
OPlan	300
Oracle Patch Types	302
Patch Set Updates	303
Critical Patch Updates and Security Patch Updates	304
Oracle Patching Standard	304
One-Off Patches	304
Exadata High Availability Upgrades	305
Reviewing Settings with Exachk	306
Exadata Full Stack Upgrade	307
Exadata Upgrade Path	307
Downloading Patches for Exadata and ZFS	311
Upgrading the Cell Nodes	312
Updating the Compute Nodes	315
Updating InfiniBand Switches	319
Updating Grid Home	319
Upgrading Ethernet Switches	323
Upgrading the KVM Switch	331
Upgrading PDUs	332
ZFS Upgrade	333
ZFSSA Configuration and Upgrade	333
ZFS Update Stage 1	334
ZFS Update Stage 2	334
Updating ZFS BIOS	335
Summary	336

Chapter 12	ZFS Storage Appliance for Exadata	337
	ZFS Family Line	338
	Increased Storage Capacity	340
	Reclaiming Resources and Space from DBFS	341
	Information Lifecycle Management	342
	ZFSSA Browser User Interface	342
	Creating NFS Shares	343
	Preparing Exadata for Direct NFS	345
	Configuring and Mounting the NFS Share	348
	Snapshots	349
	Clones	351
	Snapshots and Clones with Data Guard	352
	Best-Practice Settings on ZFS Share	353
	Other Industry Use Cases	355
	Learning on the Simulator	355
	Summary	356
Chapter 13	Exadata Performance Tuning	357
	Oracle Performance Tuning	357
	Systematic Oracle Performance Tuning	358
	Oracle Performance Troubleshooting	359
	Application Design for Exadata	362
	Database Design for Exadata	364
	Storage Indexes	365
	Offloading	365
	Exadata Smart Flash Cache and Indexes	366
	Index Design for New Applications	367
	Indexing Strategy for Existing Applications	368
	Choosing Compression Levels	372
	SQL Tuning for Exadata	372
	Exadata RAC Tuning	374
	Global Cache Basics	374
	RAC Tuning Principles	375
	Cluster Overhead	376

Reducing Global Cache Latency	378
LMS Latency	381
Balancing an Exadata RAC Database	383
Balancing Workloads with IORM and DBRM	385
Optimizing Exadata I/O	386
Leveraging Flash More Effectively	387
Configuring the Write-Back Facility	387
Configuring ASM	387
Changing the Block Size	388
Summary	388
Chapter 14 Database Consolidation on Exadata	389
Database Consolidation Models	389
Exadata Consolidation Planning	390
Grouping Applications	391
Server Pools	391
Chargeback	392
Evaluating Sizing Requirements	393
Setting Up Exadata for Consolidation	394
Storage and I/O Settings	394
Memory Settings	398
CPU Settings	399
Isolation Management	405
Fault Isolation in Database Consolidation	406
Fault Isolation in Schema Consolidation	406
Operational Isolation in Database Consolidation	406
Operational Isolation in Schema Consolidation	407
Resource Isolation in Database Consolidation	408
Resource Isolation in Schema Consolidation	408
Security Isolation in Database Consolidation	409
Security Isolation in Schema Consolidation	409
12c Pluggable Database	410
Summary	411

Chapter 15	Exadata Smart Flash Cache in Depth	413
	Solid-State Disk Technology	413
	Limitations of Disk Technology	413
	The Rise of Solid-State Flash Disks	415
	Flash SSD Architecture and Performance	417
	The Oracle Database Flash Cache	421
	Exadata Flash Hardware	422
	Exadata Smart Flash Cache	423
	Exadata Smart Flash Cache Architecture	423
	What the Exadata Smart Flash Cache Stores	425
	Flash Cache Compression	425
	CELL_FLASH_CACHE Storage Clause	426
	Flash Cache KEEP Expiration	426
	Monitoring Exadata Smart Flash Cache	427
	Exadata Smart Flash Cache Performance	429
	Exadata Smart Flash Logging	433
	Controlling and Monitoring Smart Flash Logging	436
	Testing Exadata Smart Flash Logging	437
	Smart Flash Cache WriteBack	439
	Data File Write I/O Bottlenecks	440
	Write-Back Cache Architecture	441
	Enabling and Disabling the Write-Back Cache	442
	Write-Back Cache Performance	442
	Summary	443
Chapter 16	Advanced Exadata Flash Configuration	445
	Using Flash as Grid Disks	445
	Grid Disks, Cell Disks, and the Flash Cache	446
	Creating a Flash-Based ASM Disk Group	448
	Flash Tablespace versus Flash Cache	451
	Index Fetch Performance	451
	Scan Performance	452
	Creating a Flash Temporary Tablespace	453
	Using Flash for Redo Logs	456

Storage Tiering Solutions	458
Using Partitions to Tier Data	459
12c ILM and ADO	462
Summary	463
Chapter 17 Exadata Tools and Utilities	465
Exadata Diagnostic Tools	465
SunDiag	466
Exachk: Exadata Health Check	467
InfiniBand Network Diagnostic Tools	469
Verifying InfiniBand Topology	472
infinicheck	473
Other Useful Exadata Commands	475
imageinfo and imagehistory	475
InfiniBand Network-Related Commands	476
Monitoring Exadata Storage Cells	484
Dell Software Tools for Exadata	484
Monitoring the Cell with Enterprise Manager	487
Summary	491
Index	493



Preface

Blazingly fast, Exadata is Oracle's complete database machine—with unparalleled performance brought about by engineering hardware and software technologies from Oracle and Sun. Exadata has been widely embraced by enterprise users worldwide, including government, military, and corporate entities.

Authored by a world-renowned veteran author team of Oracle ACEs/ACE directors with a proven track record of multiple bestselling books and an active presence on the Oracle speaking circuit, this book is a blend of real-world, hands-on operations guide and expert handbook for Exadata Database Machine administrators (DMAs).

Targeted for Oracle Exadata DBAs and DMAs, this expert's handbook is intended to serve as a practical, technical, go-to reference for performing administration operations and tasks for Oracle's Exadata Database Machine. This book is a

- Practical, technical guide for performing setup and administration of Oracle's Exadata Database Machine
- Expert, pro-level Exadata handbook
- Real-world, hands-on Exadata operations guide
- Expert deployment, management, administration, support, and monitoring guide and handbook
- Practical, best-practices advice from real-life Exadata DMAs

The authors have written this handbook for an audience of intermediate-level, power, and expert users of the Exadata Database Machine.

This book covers both 11*g* and 12*c* versions of the underlying Exadata software.



Acknowledgments

Tariq Farooq

To begin with, I would like to express my endless gratitude and thanks for anything and everything in my life to the Almighty ALLAH, the lord of the worlds, the most gracious, the most merciful.

I dedicate this book to my parents, Mr. and Mrs. Abdullah Farooq; my amazing wife, Ambreen; and my wonderful kids, Sumaiya, Hafsa, Fatima, and Muhammad-Talha; my nephews, Muhammad-Hamza, Muhammad Saad, Abdul-Karim, and Ibrahim. Without all of their perpetual support, this book would not have come to fruition. My endless thanks to them as I dedicated more than two years of my spare time to this book, most of which was on airplanes and in late nights and weekends at home.

My heartfelt gratitude to my friends at the Oracle Technology Network (OTN), colleagues in the Oracle ACE fellowship, my co-workers and everyone else within the Oracle community, as well as my workplace for standing behind me in my quest to bring this project to completion, especially Jim Czuprynski, Mike Ault, Bert Scalzo, Sandesh Rao, Bjoern Rost, Karen Clark, Vikas Chauhan, Suri Gurram, John Casale, and my buddy Dave Vitalo.

Considering that Exadata was the hot new kid on the Oracle block, I had been contemplating and reaching out to a lot of folks about writing a book on Exadata for over a year, before the stars got aligned and we started working on this project. From inception to writing to technical review to production, authoring a book is a complex, labor-intensive, lengthy, and at times painful process; this book would

not have been possible without the endless help and guidance of the awesome Addison-Wesley team. A very special thank-you to Greg Doench, the executive editor, and all the other folks at Addison-Wesley, who stood like a rock behind this project. Kudos to the book review and editorial team at Addison-Wesley for a job well done. A special thanks to Nabil Nawaz for contributing and helping out with the authoring process.

Finally, many appreciative thanks to my buddies and co-authors—Charles, Nitin, Sridhar, Syed, and Guy—for the amazing team effort that allowed us to bring this book to you, my dear reader. My sincerest hope is that you will learn from this book and that you will enjoy reading it as much as we did researching and authoring it.

Charles Kim

I dedicate this book to my father, who passed away to be with the Lord earlier this year. I thank my wonderful wife, Melissa, who always supported my career aspirations no matter how crazy they seemed. Last, I would like to thank my three precious sons, Isaiah, Jeremiah, and Noah, for always making me smile.

Nitin Vengurlekar

I would like to thank my family, Nisha, Ishan, Priya, and Penny, and especially my mother, father, and Marlie.

Sridhar Avantsa

Any success that I have had or will have is primarily due to the support, encouragement, and guidance I have received from people I have had the distinct honor, pleasure, and good luck to have in my life. There are a lot more folks who have had a profound impact on my life, but the individuals listed here have been my Rocks of Gibraltar over the years. I dedicate this book to these individuals as a small token of my appreciation and thanks. I am forever indebted to them for the love, encouragement, and support they have provided over the years.

To my parents, Mr. Avantsa Krishna and Mrs. Avantsa Manga, who for years denied themselves any pleasures or luxuries to ensure that they provided me with the education, facilities, and opportunities to build a solid foundation for success in the future. To my grandfather, Mr. A. V. Rayudu, whose unflinching faith in his grandchildren helped us to really believe in ourselves.

To my elder brother, Srinivas Avantsa, who understood and knew me better than I ever did, for guiding and supporting me through my formative years. If not for him, I might never have made it to college. To my cousin, Nand Kishore Avantsa, who introduced me to the fascinating world of computer science.

To my wife of the last 19 years, Gita Avantsa, the love of my life and the very best mother my children could ask for. She has stood by me and supported me over the years, through thick and thin, with a smile that always reminds me that “everything is going to be all right.” Without her support and understanding, balancing work with writing a book would have been impossible.

To my sons, eight-year-old Nikhil and seven-year-old Tarun, who have put up with me on a daily basis. Your innocence, intelligence, and resiliency never cease to amaze me.

Last, but not least, I want to thank my coauthors for including me in this journey, and my team at Rolta AdvizeX for helping me—among them Rich Niemiec, Robert Yingst, and Michael Messina stand out.

Thank you ever so much, everybody.

Guy Harrison

I dedicate this work to the memory of Catherine Maree Arnold (1981–2010).

Thanks as always to Jenni, Chris, Kate, Mike, and William Harrison who give my life meaning and happiness. Thanks Tariq and Greg for giving me the opportunity to work with Sridhar, Syed, Charles, Sahid, Bert, Michael, Nitin, Nabil, and Rahaman.

Syed Jaffar Hussain

I would like to dedicate this book to my parents, Mr. and Mrs. Saifulla; my wife, Ayesha; my three little champs, Ashfaq, Arfan, and Aahil; and the entire Oracle community.

First and the foremost, I thank the Almighty for giving me everything in life and my parents for giving me wonderful life and making me what I am today. Also, I owe a very big thank-you to my family for allowing me to concentrate on writing assignments and helping me complete the project on time. Beyond a doubt, the project wouldn't have been possible without the tremendous moral support and encouragement of my wife, friends, and colleagues. Thank you, everyone, once again.

I would like to thank my management (Khalid Al-Qathany, Hussain Mobarak AlKalifah, Majed Saleh AlShuaibi), my dearest friends (Khusro Khan, Mohammed Farooqui, Naresh Kumar, Shaukat Ali, Chand Basha, Gaffar Baig, Hakeem, Mohsin, Inam Ullah Bukhari, Rizwan Siddiqui, Asad Khan), my brother Syed Shafiullah, fellow colleagues, well-wishers, supporters, nears, and dears for their immense support and constant encouragement. I can't forget thanking Mr. Vishnusivathej, Nassyam Basha, YV Ravi Kumar, Aman Sharma, and Karan and Asad Khan for helping me while writing this book.

I am also thankful from the bottom of my heart to the official technical reviewers (Sandesh Rao and Javed) for taking some valuable time from their busy schedules to review our book and for providing great input. I can't conclude the list without mentioning the members of the Addison-Wesley team who put this project together.

Nabil Nawaz

I would like to first thank my wife Rabia and the kids for being patient while I was busy contributing to this book and away from family for several weekends—this project ended up taking more time than I expected. I am very lucky to have an understanding family that supported me on my first book!

I am grateful to Charles Kim for inviting me to be part of this amazing book; he also spent time revising the contributions I made and I really appreciate his guidance and all of his help. Charles has been an excellent mentor and is always willing to help anyone learn about technology.

Thank you also to Bane Radulovic from Oracle Support for all of his time to discuss and review the Exadata Stack upgrade process in detail. Without him I would have never been able to contribute to the upgrade chapter.



About the Authors



Tariq Farooq is an Oracle Technologist/Architect/Problem-Solver and has been working with various Oracle Technologies for more than 24 years in very complex environments at some of the world's largest organizations. Having presented at almost every major Oracle conference/event all over the world, he is an award-winning speaker, community leader/organizer, author, forum contributor, and tech blogger. He is the founding president of the IOUG Virtualization & Cloud Computing Special Interest Group and the BrainSurface social network for the various Oracle communities. Tariq founded, organized, and chaired various Oracle conferences including, among others, the OTN Middle East and North Africa (MENA) Tour, the OTN Europe Middle East and Africa (EMEA) tour, VirtaThon (the largest online-only conference for the various Oracle domains), the CloudaThon and RACaThon series of conferences, and the first ever Oracle-centric conference at the Massachusetts Institute of Technology (MIT) in 2011. He was the founder and anchor/show-host of the VirtaThon Internet Radio series program. Tariq is an Oracle RAC Certified Expert and holds a total of 14 professional Oracle Certifications. Having authored over 100 articles, whitepapers, and other publications, Tariq is the coauthor of the *Expert Oracle RAC 12c*, *Building DB Clouds in Oracle 12c*, *Oracle Problem-Solving*, and *Troubleshooting Oracle* books. Tariq has been awarded the Oracle ACE and ACE Director awards from 2010–2015.



Charles Kim is an architect in Hadoop/Big Data, Linux infrastructure, cloud, virtualization, and Oracle Clustering technologies. He holds certifications in Oracle, VMware, Red Hat Linux, and Microsoft and has over 23 years of IT experience on mission- and business-critical systems. Charles presents regularly at Oracle OpenWorld, VMWorld, IOUG, and various local/regional user group conferences. He is an Oracle ACE director, VMware vExpert, Oracle Certified DBA, Certified Exadata Specialist, and a RAC Certified Expert. His books include *Oracle Database 11g: New Features for DBAs and Developers*, *Linux Recipes for Oracle DBAs*, *Oracle Data Guard 11g Handbook*, *Virtualize Oracle Business Critical Databases*, *Oracle ASM 12c Pocket Reference Guide*, and *Virtualizing Hadoop*. Charles is the current president of the Cloud Computing (and Virtualization) SIG for the Independent Oracle User Group and blogs regularly at DBAExpert.com/blog.



Nitin Vengurlekar is the cofounder and CTO of Viscosity North America where he is responsible for partner relationships and end-to-end solution deployment. Prior to joining Viscosity, he worked for Oracle for more than 17 years, mostly in the RAC engineering group and in RAC product management. He spent his last three years at Oracle as database cloud architect/evangelist in Oracle's Cloud Strategy Group in charge of private database cloud messaging. Nitin is a well-known speaker in the areas of Oracle storage, high availability, Oracle RAC, and private database cloud. He has written or contributed to *Database Cloud Storage*, *Oracle Automatic Storage Management*, and *Oracle Data Guard 11g Handbook*, and has written many papers and contributed to Oracle documentation as well as Oracle educational material. With more than 28 years of IT experience, Nitin is a seasoned systems architect who has successfully assisted numerous customers in deploying highly available Oracle systems.



Sridhar Avantsa started his career with Oracle in 1991 as a developer. Over the years he progressed to become a DBA and an architect. Currently he runs the National Oracle Database Infrastructure Consulting Practice for Rolta AdvizeX (formerly known as TUSC), which he joined in 2006 as a technical management consultant. His specific areas of interest and expertise include infrastructure architecture, database performance tuning, high availability/disaster recovery and business continuity planning, Oracle RAC and Clustering, and the Oracle engineering systems. Sridhar has been an active member of the Oracle community as a presenter and as a member of Oracle Expert Panels at conferences.



Guy Harrison is an Oracle ACE and Executive Director of research and development at Dell Software. He is the author of *Oracle Performance Survival Guide* and (with Steven Feuerstein) *MySQL Stored Procedure Programming* as well as other books, articles, and presentations on database technology. He also writes a monthly column for *Database Trends and Applications* (www.dbta.com).



Syed Jaffar Hussain is an Oracle Database expert with more than 20 years of IT experience. In the past 15 years he has been involved with several local and large-scale international banks where he implemented and managed highly complex cluster and non-cluster environments with hundreds of business-critical databases. Oracle awarded him the prestigious “Best DBA of the Year” and Oracle ACE director status in 2011. He also acquired industry-best Oracle credentials, Oracle Certified Master (OCM), Oracle RAC Expert, OCP DBA 8i,9i,10g, and 11g in addition to ITIL expertise. Syed is an active Oracle speaker who regularly presents technical sessions and webinars at many Oracle events. You can visit his technical blog at <http://jaffardba.blogspot.com>. In addition to being part of the core technical review committee for Oracle technology oriented books, he also coauthored *Oracle 11g R1/R2 Real Application Clusters Essentials* and *Oracle Expert RAC*.

This page intentionally left blank



About the Technical Reviewers and Contributors



Dr. Bert Scalzo is a world-renowned database expert, Oracle ACE, author, Chief Architect at HGST, and formerly a member of Dell Software's TOAD dev team. With three decades of Oracle database experience to draw on, Bert's webcasts garner high attendance and participation rates. His work history includes time at both Oracle Education and Oracle Consulting. Bert holds several Oracle Masters certifications and has an extensive academic background that includes a BS, MS, and Ph.D. in computer science, as well as an MBA, and insurance industry designations.



Javid Ur Rahaman has more than 15 years of experience with various Oracle technologies working in the APAC, USA, and African regions. He currently works as a Practice Lead—Oracle Managed Services at Rapidflow Apps Inc., a California-based VCP Specialized Oracle Partner. He contributes to various seminars on Oracle technologies at different forums. Javid's areas of focus include large-scale national and international implementations of Oracle Exadata, Exalogic, Exalytics, ODA, RAC, OBIEE, SOA, OTM, Web Center, Oracle Demantra, Cloud Integration with EBS, HCM Cloud Implementation, and EBS among other Oracle technologies. Javid can be followed on his blog <http://oraclesynapse.com>, on Twitter @jrahaman7.



Nabil Nawaz started his career with Oracle in 1997 and currently works as a senior consultant at Viscosity North America. He has more than 17 years' experience working as an Oracle DBA starting with version 7.1.6; he is an OCP, is Exadata certified; and is also an Oracle ACE associate. His background is quite vast with Oracle and he has had the opportunity to work as a consultant in many large Fortune 500 companies focusing on architecting high availability solutions such as RAC, Dataguard, and most recently Oracle Exadata and Virtualization technologies. Nabil is a native of Dallas, Texas, and resides there with his wife Rabia and three children. Nabil regularly speaks at Oracle Users groups and can be followed at his blog <http://nnawaz.blogspot.com/> and on Twitter @Nabil_Nawaz.



Sandesh Rao is an Oracle technologist and evangelist, solving critical customer escalations, and has been part of the Oracle RAC Development organization for the past 8 years, developing several products to help Oracle customers. Sandesh regularly speaks at Oracle OpenWorld, COLLABORATE, and webinars/seminars as part of the Oracle RAC Special Interest Group on products related to Oracle high availability like RAC, ASM, and Grid Infrastructure. Involved with working on Engineered systems and best practices for the same through tools like exachk, he is also responsible for Diagnosability across the Oracle Database product stack and development of tools in this space. Sandesh has been working with customers, architecting, and designing solutions and solving critical problems and leading four different teams across different products like Database, Enterprise Manager, Middleware, and now the Cloud space for customer private and public clouds. Learn more about Sandesh at <http://goo.gl/t6XVAQ>.

This page intentionally left blank



The Secret Sauce: Exadata Storage Cells

The core focus of this chapter is to examine the **Exadata Storage Server**, popularly known as **Exadata Cell**, in more detail. Storage Cells are arguably the “secret sauce” or critical element that makes Exadata scale and perform so well. While the previous chapters explained the features and advantages of Exadata in 360-degree view, in this chapter we’ll complete the picture by focusing on the highly specialized storage.

DBAs have almost universally acknowledged that historically the Achilles’ heel for database performance was disk I/O. The faster the database and/or the more spindles handling database requests, the better the performance. Of course, newer storage technologies such as Flash disk or SSD (solid-state disk) don’t fit very nicely into that generalization; however, they are still new enough and remain relatively expensive such that the bulk of storage remains on magnetic disk. Hence I/O performance remains an issue that Exadata storage or cell server attempts to address. This chapter focuses more on the abstract concepts of *what* the cell storages are and *how* they operate in the setup.

An Overview of Exadata Storage Server

An Exadata Database Machine is typically shipped with three preconfigured hardware components: Compute (database) Nodes, cell storage, and ultrafast InfiniBand storage network. The Exadata Storage Server is not simply another storage server.

It is capable of delivering unique features and provides more functionality than any third-party traditional storage server. It plays a significant role in the Exadata Database Machine by provisioning storage capacity and the very unique Exadata features.

Exadata Storage Server is not just another typical storage area network (SAN) storage device or black box that facilitates and fulfills the storage requirements. It has the intelligent Exadata Storage Software that provides the capabilities of Cell Offload processing (Smart Scan), Storage Indexes, Hybrid Columnar Compression (HCC or EHCC), I/O Resource Management (IORM), Smart Flash Cache, fast file creation, and so on. The Exadata software features are covered in detail in other chapters.

In general, each Exadata machine comes in various configurations. The number of database and cell servers purely depends on the Exadata rack capacity you choose. It comes in Eighth, Quarter, Half, and Full Racks. Depending on your choice of configuration, the cell server range can be three, seven, or 14. You have the flexibility to choose the size that best suits your needs, and you can scale up as the demand rises in the future. Figure 3.1 represents the cell server count and capacity of different Exadata racks:

- A Quarter Rack comes with two Compute (DB) Nodes and three storage servers.
- A Half Rack comes with four Compute (DB) Nodes and seven storage servers.
- A Full Rack comes with eight Compute (DB) Nodes and 14 storage servers.



Figure 3.1 The storage server count is based on rack size.

For instance, an Exadata X4 Storage Server has the following software and hardware capacity:

- A mandatory preconfigured Oracle Enterprise Linux (OEL) operating system
- Three default system user configurations: root, celladmin, and cellmonitor
- Intelligent Exadata Storage Server Software
- Two six-core Intel Xeon E5-2630 v2 processors (2.6GHz)
- 12 x 1.2TB SAS disks with High Performance (HP) and 10,000 RPM or 12 x 4TB disks with High Capacity (HC) and 7200 RPM
- 4 x 800GB Sun Flash Accelerator 480 PCI cards
- Dual-port, 2 x InfiniBand, and 4 x QDR (40GB) active-active connectivity
- 96GB memory per server
- 3.2TB Exadata Smart Flash Cache
- CELLSRV, Restart Server (RS), and Management Server (MS) background services

Each Exadata Storage Server is managed and treated individually. The Cell Control Command-Line Interface (CellCLI) utility is used to administrate the local cell, and the Distributed Command-Line Interface (dcli) utility is used to administrate the remote Exadata cell operations. The CellCLI is used to perform most of the administration and management tasks on a local cell server. The dcli utility (noninteractive) has the capability to centralize cell management across all cell servers on an Exadata machine.

Storage Server Architecture

In addition to any traditional storage server components like CPU, memory, network interface controllers (NICs), storage disks, and so on, an Exadata Storage Cell comes preloaded with the OEL operating system and intelligent Exadata Storage Server Software.

Figure 3.2 depicts the typical Quarter Rack Exadata Storage Cell architecture details, two Compute Nodes and three Storage Cells, and how the communication and relation between a Compute Node and Storage Cells are established.

An ASM instance running on the Compute (database) Node communicates with a storage server through an InfiniBand network connection using the special Intelligent Database (iDB) protocol. Additionally, the iDB protocol provides aggregation and failover to the interconnect network bandwidth. An Eighth or Quarter Rack

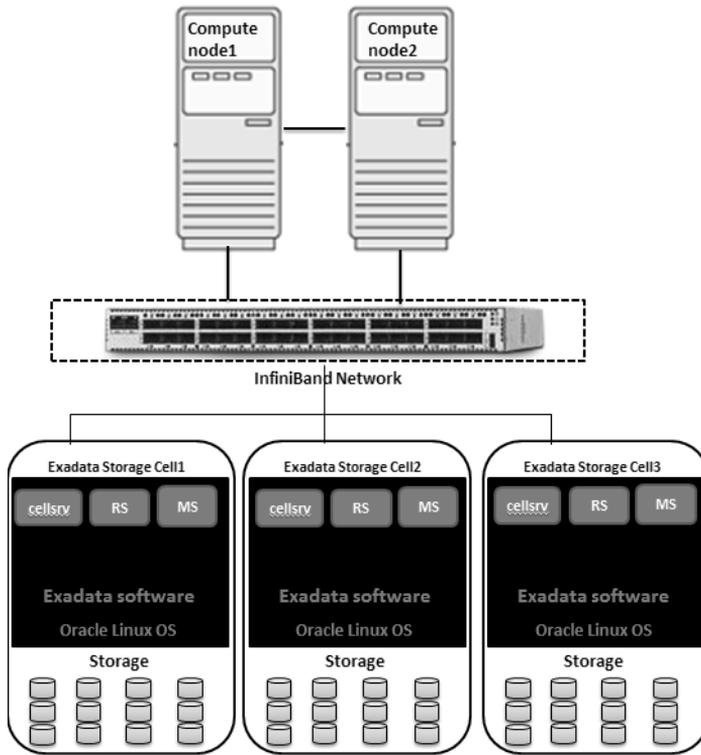


Figure 3.2 Eighth/Quarter Rack Exadata Database Machine compute and cell server architecture

comes with two InfiniBand network switches, known as leaf switches, configured between a cell and Compute Nodes to provide a communication path tolerating any switch failure. A third switch (spine) is provided only in Half and Full Rack capacity.

Each Exadata Storage Server comes with a fixed 12 uniform High Performance (HP) or High Capacity (HC) physical disks, preconfigured OEL operating system, Exadata Storage Software, and three key background services. The storage server can be accessed with three options: local login, secure shell (SSH), and KVM switch.

Cell Software Components and Management

Three key software components that run in the background are responsible for delivering the core functionality of the cell server: Cell Server (CELLSRV), Management Server (MS), and Restart Server (RS).

Cell Server

Cell Server (CELLSRV) is a multithreaded process. Arguably the heaviest among the three processes, it uses the most CPU cycles, and it also uses the special iDB protocol over InfiniBand (Oracle data transfer protocol) for communication between an ASM instance and Storage Cells. It is the primary component running on the cell and is responsible for performing Exadata advanced responsibilities, such as SQL Offloading (Smart Scans), prioritizing and scheduling an I/O on the underlying disks, implementing IORM, and so on.

It is recommended that you set the high limits of soft and hard values for the celladmin user to avoid as few ORA-600 errors as possible. As part of the disk management, when the CELLSRV process discovers that a particular disk is performing poorly, it will notify an ASM instance immediately to take the grid disk offline. Each time a database is started, it gets registered with the cell service on the cell server, and the limit of database connection to each cell service is up to 255.

The following query helps you identify the CELLSRV hang incidents on the cell:

```
# CellCLI> list alerthistory where alertMessage like ".*CELLSRV hang.*" detail
```

To diagnose CELLSRV issues, such as when CELLSRV is hung, consuming a significant amount of CPU and memory, memory leaks, and so on, you can generate a state dump of the CELLSRV process with the following command to troubleshoot the issue:

```
# CellCLI> alter cell events = "immediate cellsrv.cellsrv_statedump(0,0)"
# CellCLI> alter cell events = "immediate cellsrv.cellsrv_statedump(2,0)"
```

The following output is generated upon execution of the command, which can be referred to for further analysis of the current CELLSRV situation:

```
Dump sequence #1 has been written to /opt/oracle/cell111.2.3.3.0_LINUX.X64_131014.1/log/
diag/asm/cell/cell12/trace/svtrc_31243_80.trc
```

```
Cell usdwilo03 successfully altered
Cell cell12 successfully altered
```

Each time a state dump is performed, the sequence count for dump is increased. The trace file name in the preceding example is svtrc_18140_21.trc. The trace file contains detailed information about the cell, that is, cell software version, dump sequence, memory information, cell parameters, statistics, disk owner information, InfiniBand information, and so on. At any point in time, if you want to know the internal working condition of a CELLSRV process, you can generate a state dump to get the complete details.

As mentioned earlier, each cell is managed individually with the CellCLI utility. The CellCLI utility provides a command-line interface to the cell management functions, such as cell initial configuration, cell disk and grid disk creation, and performance monitoring. The CellCLI utility runs on the cell and is accessible from a client computer that has network access to the Storage Cell or is directly connected to the cell. The CellCLI utility communicates with Management Server to administer the Storage Cell.

If you want to manually stop, start, or restart the CELLSRV service on the cell, use the following commands:

```
# CellCLI> alter cell shutdown services cellsrv [FORCE]
```

If you encounter any issue while shutting down the CELLSRV service, use the FORCE option to shut down the service forcefully.

```
# CellCLI> alter cell startup services cellsrv
```

This will start the CELLSRV service on the local cell.

```
# CellCLI> alter cell restart services cellsrv
```

This will stop/start (restart) the CELLSRV service on the local cell.

```
# CellCLI> list cell attributes cellsrvStatus detail
```

This prints the current status of the CELLSRV process on the local cell.

Management Server

Management Server (MS) provides standard cell configuration and management functionality in coordination with CellCLI. It performs the following additional tasks:

- Periodically parses the symbolic links in the /dev/disk/by-path corresponding to the FMOD Flash Disks, to verify their presence and visibility to the underlying OS.
- Tracks down the hardware-level changes on the cell server and notifies the CELLSRV through an ioctl system call.
- Collects, computes, and manages storage server metrics.
- Rebuilds the virtual drives when a disk is replaced.
- Typically, when a disk performs poorly, the associated grid disk and cell disk will be taken offline, and MS service will notify the CELLSRV service.

Apart from these characteristics, MS also triggers the following automated tasks every hour:

- Deletes files older than seven days from the ADR directory, \$LOG_HOME, and all metric history.
- Performs alert log file auto-maintenance whenever the file size reaches 10MB in size and deletes previous copies of the alert log when they become seven days old.
- Notifies when file utilization reaches 80%.

The MS service can start-stop-restart and verify the current status with the following commands:

```
# CellCLI> alter cell shutdown services ms
```

This shuts down the MS service on the local cell.

```
# CellCLI> alter cell startup services ms
```

This starts up the MS service on the local cell.

```
# CellCLI> alter cell restart services ms
```

This stops/starts (restarts) the MS service on the local cell.

```
# CellCLI> list cell attributes msStatus detail
```

This prints the current MS service status.

Restart Server

Restart Server (RS) monitors other services on the cell server and restarts them automatically in case any service needs to be restarted. Also, it handles planned service restarts as part of any software updates. The cellrssrm is the main RS process and spans three child processes: cellrsomt, cellrsbmt, and cellesmmt.

The RS service can start-stop-restart and verify the current status with the following commands:

```
# CellCLI> alter cell shutdown services rs
# CellCLI> alter cell startup services rs
# CellCLI> alter cell restart services rs
# CellCLI> list cell attributes rsStatus detail
```

All three component services are automatically started and stopped whenever the cell server is powered off or on. However, sometimes you might need to stop the service(s) manually; for instance, to enable the write-back Flash Cache feature, you need to stop the cell service.

The `alter cell shutdown services all [FORCE]` command shuts down all services together, and the `alter cell startup services all` command starts up all services together. All grid disks and related ASM disks will become inactive and go offline respectively upon stopping either all services or just the cell server, and the communication between the cell and ASM/RDBMS instances will be disturbed.

The following commands can be used to verify the current status of all three background processes on the cell:

```
# /etc/init.d/celld status
# /etc/init.d/service cell status

rsStatus:          running
msStatus:          running
cellsrvStatus:    running
```

Configuring Mail Server for Alert Notifications

After the Exadata Database Machine initial deployment, configure the SMTP server settings on each cell to receive notification whenever the storage server generates alerts and warnings. The following piece of code shows an example to configure SMTP server settings on the local cell server:

```
# CellCLI > ALTER CELL realmName=ERP_HO,-
  smtpServer= 'your_domain.com',-
  smtpFromAddr='prd.cell01@domain.com', -
  smtpPwd='password123',-
  smtpToAddr='dba_group@domain.com',-
  notificationPolicy='clear, warning, critical',-
  notificationMethod='email,snmp'
```

Once the SMTP settings are configured, use the following command to validate the cell:

```
# CellCLI> ALTER CELL VALIDATE MAIL
```

Displaying Cell Server Details

The following command displays cell server comprehensive details, such as cell services status, cell name, ID, interconnect details, and so on:

```
# CellCLI> list cell detail

name:                cel01
bbuTempThreshold:   60
bbuChargeThreshold: 800
bmcType:            IPMI
cellVersion:        OSS_11.2.3.2.1_LINUX.X64_130109
cpuCount:           24
diagHistoryDays:    7
fanCount:           12/12
fanStatus:          normal
flashCacheMode:     WriteBack
id:                 1210FMM04Y
interconnectCount:  3
interconnect1:      bondib0
iormBoost:          9.2
ipaddress1:         192.168.10.19/22
kernelVersion:      2.6.32-400.11.1.el5uek
locatorLEDStatus:   off
makeModel:          Oracle Corporation SUN FIRE X4270 M2 SERVER SAS
metricHistoryDays:  7
notificationMethod: mail,snmp
notificationPolicy: critical,warning,clear
offloadEfficiency:  53.7
powerCount:         2/2
powerStatus:        normal
releaseVersion:     11.2.3.2.1
upTime:             376 days, 19:02
cellsrvStatus:      running
msStatus:           running
rsStatus:           running
```

Cell Metrics and Alert History

Cell metrics and alert history provide valuable statistics for optimizing the Exadata storage resources and components on the cell. Using the `metricdefinition`, `metriccurrent`, and `metrichistory` commands, you can display the historical and current metrics of any Exadata component, such as cell disk, Flash Cache, grid disks, I/O, host, and so on:

```
CellCLI> list metricdefinition cl_cput detail
name:                CL_CPUT
description:         "Percentage of time over the previous
metricType:         Instantaneous
objectType:         CELL
unit:               %

CellCLI> list metriccurrent where objecttype = 'CELL' detail
name:                CL_BBU_CHARGE
alertState:         normal
collectionTime:     2015-01-14T18:34:40+03:00
metricObjectName:   usdwilo18
metricType:         Instantaneous
metricValue:        0.0 %
objectType:         CELL

name:                CL_BBU_TEMP
alertState:         normal
collectionTime:     2015-01-14T18:34:40+03:00
```

```

metricObjectName:    usdwilo18
metricType:          Instantaneous
metricValue:         0.0 C
objectType:          CELL

name:                CL_CPUPT_CS
alertState:          normal
collectionTime:      2015-01-14T18:34:40+03:00
metricObjectName:    usdwilo18
metricType:          Instantaneous
metricValue:         1.6 %
objectType:          CELL

name:                CL_CPUPT_MS
alertState:          normal
collectionTime:      2015-01-14T18:34:40+03:00
metricObjectName:    usdwilo18
metricType:          Instantaneous
metricValue:         0.0 %
objectType:          CELL

```

```

CellCLI> list metriccurrent cl_cpupt detail
name:                CL_CPUPT
alertState:          normal
collectionTime:      2015-01-14T18:34:40+03:00
metricObjectName:    usdwilo18
metricType:          Instantaneous
metricValue:         2.0 %
objectType:          CELL

```

Querying Cell Alert History

Best practices suggest periodically querying the alert history. The alert history notifications are categorized as Informal, Warning, or Critical. The `activerequest`, `alertdefinition`, and `alerthistory` commands display current and historical alert details. In order to display the alert history that occurred on the cell or a particular component, use one of the following commands:

```

CellCLI> list alerthistory detail
name:                7_1
alertDescription:    "HDD disk controller battery in learn cycle"
alertMessage:        "The HDD disk controller battery is
                    performing a learn cycle. Battery Serial
                    Number : 591 Battery Type           : ibbu08
                    Battery Temperature   : 29 C Full Charge
                    Capacity   : 1405 mAh Relative Charge
                    : 100 % Ambient Temperature   : 24 C"

alertSequenceID:    7
alertShortName:     Hardware
alertType:          Stateful
beginTime:          2014-10-17T13:51:44+03:00
endTime:            2014-10-17T13:51:47+03:00
examinedBy:
metricObjectName:    Disk_Controller_Battery
notificationState:    0
sequenceBeginTime:  2014-10-17T13:51:44+03:00
severity:            info
alertAction:         "All hard disk drives may temporarily
                    enter WriteThrough caching mode as part of

```

the learn cycle. Disk write throughput might be temporarily lower during this time. The flash drives are not affected. The battery learn cycle is a normal maintenance activity that occurs quarterly and runs for approximately 1 to 12 hours. Note that many learn cycles do not require entering WriteThrough caching mode. When the disk controller cache returns to the normal WriteBack caching mode, an additional informational alert will be sent."

```

name: 7_2
alertDescription: "HDD disk controller battery back to normal"
alertMessage: "All disk drives are in WriteBack caching mode. Battery Serial Number : 591 Battery Type : ibbu08 Battery Temperature : 29 C Full Charge Capacity : 1405 mAh Relative Charge : 100 % Ambient Temperature : 24 C"
alertSequenceID: 7
alertShortName: Hardware
alertType: Stateful
beginTime: 2014-10-17T13:51:47+03:00
endTime: 2014-10-17T13:51:47+03:00
examinedBy:
metricObjectName: Disk_Controller_Battery
notificationState: 0
sequenceBeginTime: 2014-10-17T13:51:44+03:00
severity: clear
alertAction: Informational.
    
```

```

# CellCLI> list alerthistory where severity='Critical'
To view alert history of the cell categorized as 'Critical' state
    
```

```

# CellCLI> list alerthistory 4_1 detail
To display more details of the incident mentioned in the above example
    
```

Querying GV\$ Views

The following Exadata-related new V\$ dynamic views provide the cell and its wait events with statistical information that can be used to measure the cell state, IP address used, and so on:

- **V\$CELL**—provides information about cell IP addresses mentioned in the cellip.ora file
- **V\$CELL_STATE**—provides information about all the cells accessible from the database client
- **V\$CELL_THREAD_HISTORY**—contains samples of threads in the cell collected by the cell server
- **V\$CELL_REQUEST_TOTALS**—contains historical samples of requests run by the cell

Storage Architecture and Formulation

So far you have learned the fundamental concepts of cell architecture and cell management. It’s time now to go for the real treat and discuss the core component of Exadata Cell, that is, the storage layer.

Before we jump in and start discussing the Exadata storage architecture and storage preparation, let’s explore the basic differences between non-Exadata and Exadata environments.

A traditional Oracle Database deployment requires three major components: the Oracle Database server, the storage server, and the network layer, as shown in Figure 3.3.

In this particular setup the database is both the “engine” and the “transmission” as it both processes the raw data and delivers information to the user. Here the storage server is merely an I/O facilitator—it simply and blindly serves up requested data blocks to the database. Thus, if the database SQL optimizer decides it must perform a full table scan of a million blocks, both the network and storage server must process or handle one million blocks. Such a request could overwhelm the storage server cache, thus making it less effective for all users. Furthermore, the TCP/IP network protocol packet structure is not well optimized for such simple, massive data transfers—not even with jumbo frames enabled. The general-purpose network packets suffer other limitations, including excessive header overhead waste and processing costs. While this is the most common setup there is, it’s nonetheless quite inefficient.

When it comes to Exadata, an Exadata Oracle Database deployment also contains three key hardware components: the database server, the storage server, and the network between them as shown in Figure 3.4.

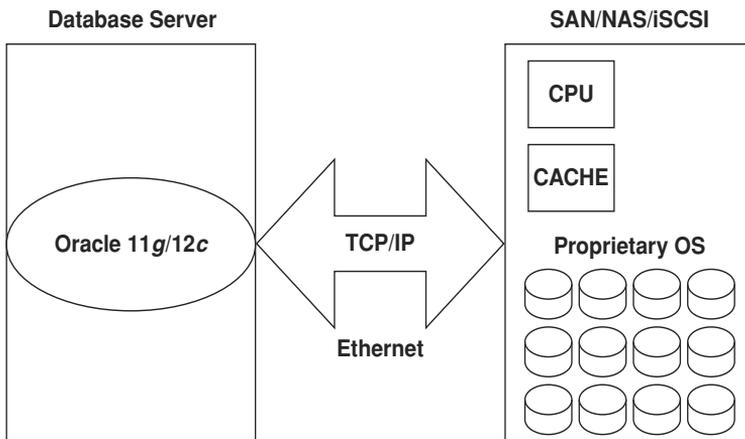


Figure 3.3 Traditional database and storage architecture

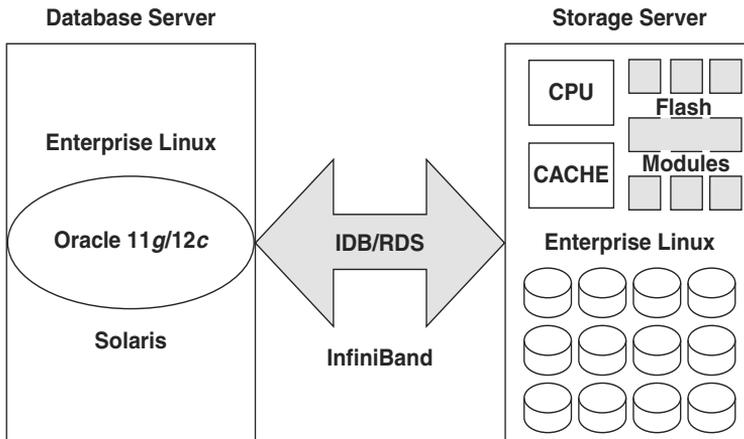


Figure 3.4 Exadata Database Machine architecture

There are four fundamental differences in the Exadata storage hardware architecture in contrast to the non-Exadata architecture, and they make all the difference in the world, especially in relation to storage scalability and performance:

- First and foremost, an Exadata cell contains **Flash Modules** that can be used either as fast disks or as additional cache (more about that later in this chapter).
- Second, the storage server is running Oracle Enterprise Linux as opposed to a proprietary OS—that’s going to enable software architectural options otherwise not possible (again to be covered later in this chapter).
- Third, the high-speed, private network between the database and cell servers is based on InfiniBand rather than Ethernet.
- Fourth and finally, all communication between the database and cell servers uses the iDB protocol transmitted via Reliable Datagram Sockets (RDS).

Let’s examine that last key difference in more detail since it enables or is directly responsible for some of the cell servers’ “special sauce.” RDS is a low-overhead, low-latency, and more CPU-efficient protocol that’s been around for years (predating Exadata). So merely using the RDS protocol between database and cell servers over InfiniBand is superior to the normal deployment scenario, but while better, it’s not what delivers the huge scalability and performance possible via cell servers. It’s the iDB and what software architectures it makes possible that deliver most of the performance gains.

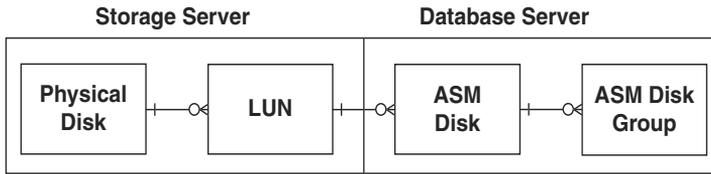


Figure 3.5 Traditional database and storage relationship

Disk Architecture in Non-Exadata

In a traditional Oracle Database deployment using Oracle’s ASM, the storage server disk architecture or layout is generally organized as shown in Figure 3.5.

Typically, the physical disks (or partitions) map to logical unit numbers, or LUNs (or devices); those are then used to create Oracle ASM disks for inclusion in ASM disk groups. While ASM is an option in traditional database deployments, Oracle generally recommends it for most new databases—and especially for RAC setups. There are of course several key benefits of using ASM:

- First and foremost, as a storage mechanism it’s highly integrated into the Oracle technology stack, and hence it works quite effectively and efficiently.
- Second, it eliminates the need for OS file system and logical volume managers (LVMs).
- Third, ASM offers dynamic load balancing and rebalancing of space when new disks are added or removed—something not possible with LVMs.
- Fourth and finally, it was designed from the ground up to work well with the needs and characteristics of Oracle Database I/O.

Disk Architecture in Exadata

Each Exadata Storage Server ships with 12 SAS physical disks of uniform size, either with the High Performance or the High Capacity configuration, and four Flash cards built in.

The initial two disks are mirrored using RAID (mdadm) and are used for the operating system, swap space, Exadata Storage Server software binaries, and various other Exadata configurations. The `df` command on the cell shows the following file system structure; right below the output, there is an explanation of the type of mount points and mapped file systems:

```
$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/md5              10321144    5839912   3956948   60% /
tmpfs                 49378532         0   49378532    0% /dev/shm
/dev/md7              3096272     775708   2163284   27% /opt/oracle
/dev/md4              116576      28583     81974    26% /boot
/dev/md11             5160448    205884   4692428    5% /var/log/oracle
```

- / is the root file system.
- /opt/oracle is where the Exadata software is installed.
- /var/log/oracle is where the cells' OS and crash logs are stored.
- The /dev/md5 and /dev/md6 are the system partitions, active and mirror copy.
- The /dev/md7 and /dev/md8 are the Exadata software installation, active and mirror copy.
- The /dev/md11 is mapped with /var/log/oracle.
- At any given point in time, only four multidevice (MD) mount points can be mounted on the cell.

Approximately 29GB of space per disk is used for this purpose. In order to know whether the LUN is the system partition or not, you can use the following command:

```
CellCLI> list lun 0_0 detail
name:                0_0
cellDisk:            CD_00_usdwilo18
deviceName:          /dev/sda
diskType:            HardDisk
id:                  0_0
isSystemLun:         TRUE
lunAutoCreate:       TRUE
lunSize:              1116.6552734375G
lunUID:              0_0
physicalDrives:      20:0
raidLevel:           0
lunWriteCacheMode:   "WriteBack, ReadAheadNone, Direct, No
                    Write Cache if Bad BBU"
status:              normal
```

There are several significant items to note here:

- First, cell servers have both Flash Cache Modules and traditional physical disks.
- Second, there's a new level of disk abstraction called the **cell disk**, which offers the ability to subdivide a LUN into partitions known as **grid disks**.
- Third, cell disks constructed from Flash Cache Modules can be further divided into Flash Cache or grid disks. Of course, physical-disk-based LUNs can map only to grid disks.
- Finally, only grid disks can be mapped to ASM disks.

At the helm of the storage layer on a cell, a physical disk is the first layer of abstraction, and each physical disk is mapped and appears as a LUN. In contrast to other storage boxes, no manual intervention is required to achieve this task as they are created automatically during Exadata Database Machine initial deployment.

The next setup is to configure the cell disk from the existing LUNs. A cell disk is created based on the existing LUN on the cell server. Once the cell disk is created, the disk can be subdivided into one or more grid disks to make them available for an ASM instance as ASM candidate disks.

As standard practice, when a cell disk is subdivided into multiple grid disks, you can then assign different performance characteristics to each grid disk according to business needs. For instance, you can assign a grid disk from a cell disk at the outermost track of a physical disk to gain the highest level of performance, and another grid disk can be assigned to the inner track of a physical disk to achieve moderate performance. The higher-performance grid disks across all cell servers then can be put together into a single disk group to place any hot data, whereas the lower-performance disks can be assembled into a disk group to store archive logs. For example, the higher-performance grid disks can be used for a data disk group and the lower-performance disks can be used to keep archive logs.

In an Oracle Exadata database deployment the cell servers can only use Oracle's ASM; however, the cell server disk architecture or layout is a little more complex with some rather different and unique options for organization. Figure 3.6 represents the relationship between disk storage and its entities.

The major difference between Flash Cache and Flash-based grid disks is quite simple. Flash Cache is autopilot caching of recently accessed database objects.

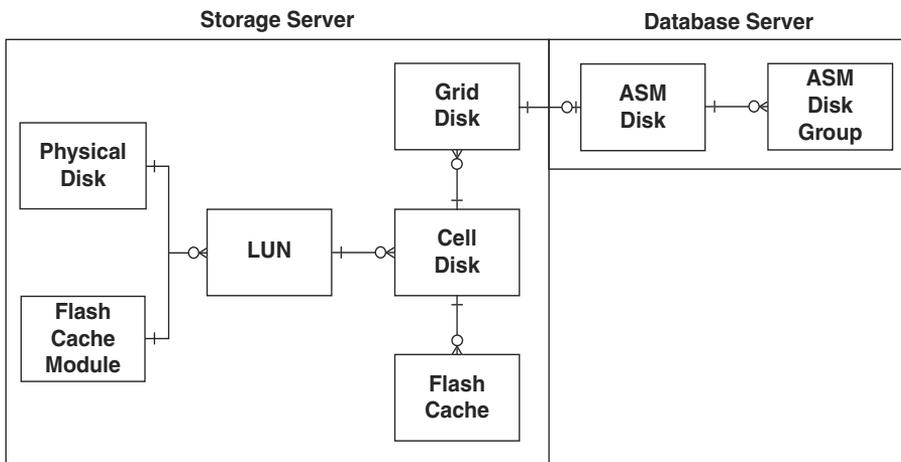


Figure 3.6 Exadata storage (physical, LUN, cell disk, grid disk, and ASM disks) formation flow

Think of it as a supersize System Global Area (SGA) at the storage level. A very similar concept known as **Smart Response Technology (SRT)** exists on some newer Intel CPUs and their chip sets, whereby an SSD can be used as front-end caching of a traditional disk drive. Flash Cache does offer the ability to manually pin database objects into it (much like pinning objects into the SGA). Here's an example of pinning the PARTS table into the Flash Cache:

```
SQL> ALTER TABLE PARTS STORAGE (CELL_FLASH_CACHE KEEP);
```

Flash grid disks, on the other hand, are simply Flash Modules organized into persistent disks for ASM use. In many ways it's like having a fast SSD disk instead of a magnetic disk on your PC. At times there will be database objects that you know will perform better if they are truly Flash based rather than contained on traditional disks (and possibly Flash cached). Hence, there are times when you'll want to create ASM disks and disk groups from Flash Modules to gain the full benefits of that speed. So for those cases you'll want to create cell and grid disks from Flash Cache Modules. The commands for doing so are covered later in this chapter.

System Users for Cell Administration

As mentioned in the beginning of the chapter, each Exadata Storage Server is typically configured with three default users with different roles. Here are the differences between the users and their capabilities:

- **root**—superuser privileges. Used to shut down and start up the storage server.
- **celladmin**—used to perform cell-level administrative tasks such as CREATE, ALTER, MODIFY cell objects, such as cell disks, grid disk, configure notification, and so on, using the CellCLI and dcli utilities
- **cellmonitor**—a monitoring user used to perform cell monitoring tasks. Unlike root and celladmin, it can't be used to CREATE, ALTER, or MODIFY any cell objects.

Following are a few practical examples.

Listing Disk Levels

To list all levels of disks, including physical disks, LUNs, cell disks, and grid disks, use the following commands:

Some CellCLI commands

If you want to list all the commands associated with CellCLI utility, use the following command:

```
CellCLI> help
```

```
HELP [topic]
Available Topics:
ALTER
ALTER ALERTHISTORY
ALTER CELL
ALTER CELLDISK
ALTER FLASHCACHE
ALTER GRIDDISK
ALTER IBPORT
ALTER IORMPLAN
ALTER LUN
ALTER PHYSICALDISK
ALTER QUARANTINE
ALTER THRESHOLD
ASSIGN KEY
CALIBRATE
CREATE
CREATE CELL
```

To list the Flash Cache disks configured on the local cell, run the following command:

```
CellCLI> list lun where disktype = 'flashdisk'
1_0      1_0      normal
1_1      1_1      normal
1_2      1_2      normal
1_3      1_3      normal
2_0      2_0      normal
2_1      2_1      normal
2_2      2_2      normal
```

To list the LUN details, such as to determine if the LUN is a system LUN or not, LUN size, ID, RAID level, device name, and other information on the local node, execute the following command:

```
CellCLI> list lun detail
name:                                0_0
cellDisk:                             CD_00_usdwilo18
deviceName:                            /dev/sda
diskType:                              HardDisk
id:                                     0_0
isSystemLun:                           TRUE
lunAutoCreate:                          TRUE
lunSize:                                1116.6552734375G
lunUID:                                 0_0
physicalDrives:                         20:0
raidLevel:                              0
lunWriteCacheMode:                     "WriteBack, ReadAheadNone, Direct, No
                                         Write Cache if Bad BBU"
status:                                 normal

name:                                0_1
cellDisk:                             CD_01_usdwilo18
deviceName:                            /dev/sdb
```

```

diskType:          HardDisk
id:                0_1
isSystemLun:      TRUE
lunAutoCreate:    TRUE
lunSize:          1116.6552734375G
lunUID:           0_1
physicalDrives:   20:1
raidLevel:        0
lunWriteCacheMode: "WriteBack, ReadAheadNone, Direct, No
                  Write Cache if Bad BBU"
status:           normal

```

To list the physical disk details, such as disk name, status, and so forth, on the local cell, run the following command:

```

CellCLI> list physicaldisk detail
name:                20:0
deviceId:            8
diskType:            HardDisk
enclosureDeviceId:  20
errMediaCount:      0
errOtherCount:      0
luns:                0_0
makeModel:           "HGST      H101212SESUN1.2T"
physicalFirmware:    A690
physicalInsertTime:  2014-05-21T04:24:40+03:00
physicalInterface:   sas
physicalSerial:      DEAT5F
physicalSize:        1117.8140487670898G
slotNumber:          0
status:              normal

name:                20:1
deviceId:            9
diskType:            HardDisk
enclosureDeviceId:  20
errMediaCount:      0
errOtherCount:      0
luns:                0_1
makeModel:           "HGST      H101212SESUN1.2T"
physicalFirmware:    A690
physicalInsertTime:  2014-05-21T04:24:40+03:00
physicalInterface:   sas
physicalSerial:      DE7ZWF
physicalSize:        1117.8140487670898G
slotNumber:          1
status:              normal

```

To list the cell disk details, such as device name, creation time, size, and so on, run the following command:

```

CellCLI> list celldisk detail
name:                CD_00_usdwilo18
comment:
creationTime:        2014-09-24T16:14:52+03:00
deviceName:          /dev/sda
devicePartition:     /dev/sda3
diskType:            HardDisk
errorCount:          0

```

```

freeSpace:          0
id:                 ac757133-886d-465c-b449-8fe35f05519c
interleaving:      none
lun:                0_0
physicalDisk:      DEAT5F
raidLevel:         0
size:               1082.84375G
status:             normal

name:               CD_01_usdwilo18
comment:
creationTime:      2014-09-24T16:14:53+03:00
deviceName:         /dev/sdb
devicePartition:   /dev/sdb3
diskType:           HardDisk
errorCount:         0
freeSpace:          0
id:                 af978555-022a-4440-9c6c-2c05f776b6cc
interleaving:      none
lun:                0_1
physicalDisk:      DE7ZWF
raidLevel:         0
size:               1082.84375G
status:             normal

```

To list the grid disk details, such as cell disks mapped to the physical disk, size, status, and so on, run the following command:

```

CellCLI> list griddisk detail
name:               DG_DBFS_CD_02_usdwilo18
asmDiskgroupName:  DG_DBFS
asmDiskName:        DG_DBFS_CD_02_USDWILO18
asmFailGroupName:  USDWILO18
availableTo:
cachingPolicy:     default
cellDisk:           CD_02_usdwilo18
comment:
creationTime:      2014-09-24T16:19:02+03:00
diskType:           HardDisk
errorCount:         0
id:                 7e2d7848-cf81-4918-bb01-d27ef3da3950
offset:             1082.84375G
size:               33.796875G
status:             active

name:               DG_DBFS_CD_03_usdwilo18
asmDiskgroupName:  DG_DBFS
asmDiskName:        DG_DBFS_CD_03_USDWILO18
asmFailGroupName:  USDWILO18
availableTo:
cachingPolicy:     default
cellDisk:           CD_03_usdwilo18
comment:
creationTime:      2014-09-24T16:19:02+03:00
diskType:           HardDisk
errorCount:         0
id:                 972be19d-5614-4b98-8806-7bdc2faf7630
offset:             1082.84375G
size:               33.796875G
status:             active

```

Configuring Cell Disks

The following command will configure 12 cell disks, one for each LUN, with the default naming convention. This is usually run as part of the initial deployment.

```
# CellCLI> CREATE CELLDISK ALL HARDDISK
```

Alternatively, use the following command to create the cell disks to enable interleaving:

```
# CellCLI> CREATE CELLDISK ALL HARDDISK INTERLEAVING='normal_redundancy'
```

Creating Grid Disks

The following command will create a grid disk at the outermost track layer of a physical disk for high performance:

```
# CellCLI> create griddisk ALL HARDDISK prefix=data, size 500G
```

The next command will create a grid disk at the inner track layer of a physical disk for less I/O-intensive applications:

```
# CellCLI> CREATE GRIDDISK ALL PREFIX=FRA
```

Configuring Flash Grid Disks

The following procedure is used to drop the current Flash Cache and rebuild with the nondefault size:

```
# CellCLI> DROP FLASHCACHE  
# CellCLI> CREATE FLASHCACHE ALL SIZE =200G  
# CellCLI> CREATE GRIDDISK ALL FLASHDISK
```

Once the Exadata storage configuration is done, the next step is to configure the database hosts to access the grid disks. The `cellinit.ora` and `cellip.ora` files must be configured at the Compute Nodes in order to access the grid disk from the cell. The following example shows the contents of each file:

```
#!/etc/oracle/cell/network-config/cellinit.ora  
Ipaddress=192.168.0.13/24
```

The `cellinit.ora` file contains the database server IP address. Each database server will have its own IP address recorded in the `cellinit.ora` file:

```
/etc/oracle/cell/network-config/cellip.ora
cell="192.168.0.11"
cell="192.168.0.12"
cell="192.168.0.13"
```

The `cellip.ora` file contains the IP addresses of all cells, and all Compute Nodes should have the same entries in order to access storage on the cell servers.

Creating an ASM Disk Group

To show how to create an ASM disk group for your database, let's take the grid disks from `cell01` and `cell02` to create a data disk group with high-redundancy capabilities:

```
SQL> CREATE DISKGROUP DG_DATA HIGH REDUNDANCY DISK 'o/*/
DATA_EX01_CD_00_ex01cel01', 'o/*/ DATA_EX01_CD_01_ex01cel01'
, 'o/*/DATA_EX01_CD_02_ex01cel01', 'o/*/ DATA_EX01_CD_00_ex01cel02'
, 'o/*/DATA_EX01_CD_01_ex01cel02', 'o/*/DATA_EX01_CD_02_ex01cel02'
'compatible.asm'='11.2.0.3', 'compatinle.rdbms'='11.2.0.2',
'cell_smart_scan'='TRUE';
```

Managing the Cell Server

Sometimes it becomes necessary, especially before and after patch deployment on the cell as well as on the Compute Nodes, to know the current cell software version and the previous version to which the cell can potentially roll back. In this context, Oracle provides two utilities in `/usr/local/bin`: `imageinfo` and `imagehistory`.

When the `imageinfo` utility is executed as the root user on the cell server as follows, it will help you get the active cell software details, such as cell kernel version, OS version, active cell image details, cell boot partitions, and so on:

```
# imageinfo

Kernel version: 2.6.32-400.11.1.el5uek #1 SMP Thu Nov 22 03:29:09 PST 2012 x86_64
Cell version: OSS_11.2.3.2.1_LINUX.X64_130109
Cell rpm version: cell-11.2.3.2.1_LINUX.X64_130109-1

Active image version: 11.2.3.2.1.130109
Active image activated: 2013-01-30 19:14:40 +0300
Active image status: success
Active system partition on device: /dev/md5
Active software partition on device: /dev/md7

In partition rollback: Impossible
```

```
Cell boot usb partition: /dev/sdm1
Cell boot usb version: 11.2.3.2.1.130109

Inactive image version: 11.2.3.2.0.120713
Inactive image activated: 2012-12-10 11:59:57 +0300
Inactive image status: success
Inactive system partition on device: /dev/md6
Inactive software partition on device: /dev/md8

Boot area has rollback archive for the version: 11.2.3.2.0.120713
Rollback to the inactive partitions: Possible
```

The `imagehistory` utility helps you get all the previous software versions installed on the particular cell:

```
#imagehistory

Version           : 11.2.3.1.1.120607
Image activation date : 2012-07-25 01:25:34 +0300
Imaging mode       : fresh
Imaging status     : success

Version           : 11.2.3.2.0.120713
Image activation date : 2012-12-10 11:59:57 +0300
Imaging mode       : out of partition upgrade
Imaging status     : success

Version           : 11.2.3.2.1.130109
Image activation date : 2013-01-30 19:14:40 +0300
Imaging mode       : out of partition upgrade
Imaging status     : success
```

The `-h` option can be used to list all parameters that are associated with the `imageinfo` and `imagehistory` utilities.

To remove the old `alerthistory` on the cell, you can use the following commands:

```
#CellCLI> drop alerthistory all -- will drop the complete alerthistory info
#CellCLI> drop alerthistory <9_1> -- will drop a particular incident history
```

Troubleshooting the Cell Server

The following sections discuss and demonstrate some of very important tools and utilities provided on Exadata to collect the diagnostic information on a cell. Most of the diagnostic tools and utilities reside under the `/opt/oracle.SupportTool` folder.

SunDiag

The `sundiag.sh` diagnostic collection script exists on each Compute Node and Storage Cell under `/opt/oracle.SupportTools`. The script can also be downloaded

from support.oracle.com. The script helps you gather the required diagnostic information related to problematic disks or any other hardware issues on the cell.

You have to execute the script as root user, as follows:

```
/opt/oracle.SupportTools/sundiag.sh
```

If you would like to gather similar diagnostic information across all cell servers, you will have to execute the script through the dcli utility.

This script generates a timestamped .tar file under /tmp/sundiag_Fileystem which can be uploaded to Oracle Support for analysis.

ExaWatcher

The new ExaWatcher utility located under /opt/oracle.ExaWatcher replaces the traditional OSWatcher utility in Exadata Storage Software 11.2.3.3 and is used for system data collection. The utility is up and running upon system reboot. It collects the statistics for the following components on the cell and keeps the log files under /opt/oracle.ExaWatcher/archive:

- Diskinfo
- IBCardino
- Iostat
- Netstat
- Ps
- Top
- Vmstat

In order to produce or extract the reports from the logs generated by the ExaWatcher utility, you will have to use the GetExaWatcherResults.sh script. You can collect input at various levels:

- FromTime until ToTime extracts range reports.
- ToTime extracts on or before time reports.
- AtTime extracts around the time reports.
- Hours extracts time in range reports.

Following are some examples:

```
# ./GetExaWatcherResults.sh --from <time frame> to <time frame>  
# ./GetExaWatcherResults.sh --at <time frame> --range 2
```

The second example extracts 2 hours starting with the time defined with the `at` parameter.

The `ExaWatcherCleanup` module is used to automatically manage the file system space used by `ExaWatcher`. Based on the limits set for space management, the module is responsible for cleaning up the old log files by removing them.

To get more help on how to use the utility, use the following commands:

```
# ExaWatcher.sh --help
# ExaWatcherResults.sh --help
```

Exachk

`Exachk` is an Oracle Exadata diagnostic tool that comes with different levels of verification and collects hardware, software, firmware, and configuration data on Exadata systems. It is strongly recommended that you include this script as part of your periodic maintenance operation tasks. Also, run the script before any migration, upgrade, or any other major change operations take place.

This script doesn't come with the Exadata machine; you will have to download it (`exachk_225_bundle.zip` file) from Oracle Support Note ID 1070954.1, which requires login credentials. The Note and the `.zip` files contain all the information required to use the tool.

CheckHWnFWProfile

The `CheckHWnFWProfile` utility verifies any hardware and firmware component details and reports if there are any recommended items missing. It is also used to validate the current configuration on the servers. This utility is used without passing any parameters, as shown in the following example:

```
# /opt/oracle.cellos/CheckHWnFWProfile
```

If the current hardware and firmware are to the correct version, it will give the `SUCCESS` output.

To obtain more information on the utilization of this utility use the `-d` option with the command.

Storage Cell Startup and Shutdown

When rebooting or shutting down the Exadata Storage Cell for maintenance or any other valid reason, ensure that you adhere to proper stop/start cell procedure

to guarantee a graceful cell shutdown. This section emphasizes the significance of complying with an appropriate cell stop and start procedure and demonstrates the steps in depth.

One of the key responsibilities of a DMA includes graceful shutdown of the cell, be it for a quick reboot or for maintenance. The shutdown shouldn't impact the underlying ASM instance and the active database that is running. That being said, shutting down the cell and its services without affecting the ASM availability largely depends on the current ASM redundancy level. Under any circumstances, it is a best practice to follow this procedure for the graceful shutdown of a cell:

1. Verify that ASM doesn't have any impact by taking the grid disks offline on the cell. Use the following command to verify the result:

```
# CellCLI> list griddisk attributes name,asmdeactivationoutcome
```

2. If the result of `asmdeactivationoutcome` is `yes` for all the grid disks listed, it is an indication that ASM will not have any impact and it is safe to deactivate all the grid disks, using the next command:

```
# CellCLI> alter griddisk all inactive
```

3. Once you turn off all the grid disks on the cell, run the first command to verify the `asmdeactivationoutcome` output and verify that all the grid disks are now offline, using this command:

```
# CellCLI> list griddisk
```

4. Now it is safe to power off/reboot/shut down the cell. As root, shut down the cell using the following command:

```
$ shutdown -h now -- OS command to shut down the cell server
```

Note

If you intend to take the cell down for a very long period of time, you will have to adjust the ASM disk's `DISK_REPAIR_ATTRIBUTE` default value to prevent ASM from dropping the disks automatically upon taking them offline. The default value is set to 3.6 hours; therefore, if you are taking the cell down for 5 hours, for example, use the following command to set the value through an ASM instance:

```
SQL> ALTER DISKGROUP DG_DATA SET ATTRIBUTE 'DISK_REPAIR_TIME'='5H';
```

You will have to adjust all the required disks on the cell.

Once the cell is rebooted or comes online, follow these instructions to bring the services and grid disks back to action:

1. First step of the procedure:

```
# CellCLI > alter griddisk all active
```

2. Second step of the procedure:

```
# CellCLI> list griddisk attributes name,asmmodestatus
```

3. Last step of the procedure:

```
# CellCLI> list cell detail
```

If you have worked on Oracle Database and cluster technologies previously, you probably know that each of them maintains an alert log file where all important events and sequences are written. Similarly, an alert file is maintain by each cell server to record all the important events of the cell, such as when the services started or stopped, disk warning messages, cell and grid disk creation, and so on. It is highly recommended that you review the logs frequently. You can also refer to the OS file to find out when the cell restarted.

Following are some of the most commonly referenced log/trace files and their locations:

- `/log/diag/asm/cell/{cell name}/trace`
- **MS log**—`/opt/oracle/cell/log/diag/asm/cell/{cell name}/trace/ms-odl.log`
- **OSWatcher logs**—`/opt/oracle.oswatcher/osw/archive`
- **OS messages**—`/var/log/messages`
- **Cell-patching-related logs**—`/var/log/cellos`

Solving Disk Problems

Yet another major responsibility of a DMA includes determining when and how a faulty (dead, predictive failure, poor performance) disk is identified and getting it replaced on an Exadata Storage Server. Although most of the procedure is automated by Oracle, including identification and notification of an underperforming, faulty, or damaged disk, it is equally important for you to understand the factors and procedures involved in troubleshooting and replacing the disk when it becomes necessary to do so.

When a disk confronts performance issues or any sort of hard failure, an alert is generated by the MS background process on the cell server, and it notifies the CELLSRV background service about the alert. At the same time, if OEM is configured, the message is also pushed to Grid Control, through which you can receive an email or SMS message.

Initially, a set of performance tests is carried out by the MS service on the disk on which the performance degradation has been identified to determine whether the behavior is a temporary glitch or a permanent one. If the disk passes the tests successfully, it is brought back to the active configuration; if not, it is marked as performing poorly and an Auto Service Request (ASR), if configured, is opened for disk replacement.

Whenever a disk failure occurs or a disk goes into predictive status, ASM automatically drops the related grid disks of the failed disk either normally or forcefully. After the disk issues are addressed and the disks are ready to go active, ASM automatically brings related grid disks online as part of the Exadata auto disk management, which is controlled by the `_AUTO_MANAGE_EXADATA_DISKS` parameter.

Typically the following actions are performed when disk issues are identified on the cell server:

1. When poor performance is detected, the cell disk and physical disk statuses are changed.
2. All grid disks of the particular cell disk are taken offline.
3. The MS service notifies the CELLSRV service about the findings, and in turn, CELLSRV notifies ASM instances to take the grid disk offline.
4. The MS service on the cell then performs a set of confinement checks to determine if the disk needs to be dropped.
5. If the disk passes the performance tests, the MS service notifies the CELLSRV service to turn all the cell disks and all its grid disks online.
6. If the disk fails the performance tests, the cell disk and physical disk statuses are modified, and the disk is removed from the active configuration.
7. The MS service notifies the CELLSRV service about the disk issues. In turn, the CELLSRV service informs ASM instances to drop all the grid disks of the cell.
8. If ASR is configured, a service request is logged to Oracle Support about disk replacement.
9. You will have to either use the spare disk to replace the faulty disk or request a replacement disk from Oracle.

Disk problems can be categorized in two levels: hard failure and predictive failure. A predictive failure is when a disk is flagged as predictive or in a poor performance state. A hard failure is when a disk goes into critical state.

When you have been notified about the critical failure state of a physical disk, through either email or SMS message, your first priority is to identify the damaged physical disk's exact name, position, location, and slot number through the

cell alert history or by reviewing the cell logs. Also refer to the ASM alert logs to ensure that ASM turned the damaged disk offline (dropped the disk) and ASM rebalancing is completed before replacing the disk.

To view the disk-related alert history, use the following command on the cell:

```
CellCLI> list alerthistory
```

```

7_1      2014-10-17T13:51:44+03:00      info      "The HDD disk controller battery
is performing a learn cycle. Battery Serial Number : 591 Battery Type      : ibbu08
Battery Temperature      : 29 C Full Charge Capacity : 1405 mAh Relative Charge      :
100 % Ambient Temperature : 24 C"
```

```

7_2      2014-10-17T13:51:47+03:00      clear     "All disk drives are in WriteBack
caching mode. Battery Serial Number : 591 Battery Type      : ibbu08 Battery
Temperature      : 29 C Full Charge Capacity : 1405 mAh Relative Charge      : 100 %
Ambient Temperature : 24 C"
```

```

# CellCLI> list physicaldisk WHERE diskType=HardDisk AND status=critical detail
# CellCLI> list physicaldisk WHERE diskType=HardDisk AND status like ".*failure.*" detail
# CellCLI> alter physicaldisk disk_name:disk_id drop for replacement
```

Verify that the grid disks of the cell disk are dropped and the rebalancing operations are completed on the ASM instances:

```
SQL> SELECT name,state from v$asm_diskgroup;
SQL> SELECT * FROM v$asm_operation;
```

Three minutes after replacing the faulty physical disk on the cell, all the grid disks and cell disks are automatically re-created, added subsequently to the respective disk group, and then rebalanced.

Enforcing Cell Security

Exadata offers many layers of security setups to meet your business needs. An Exadata cell server by default comes with open storage security, where there are no restrictions applied on accessing grid disks from ASM or database clients. Apart from open security, Oracle Exadata supports two levels of security: ASM-scoped security and database-scoped security. These control the storage access from ASM cluster or database clients.

Security can control which ASM or database clients can access a specific grid disk or pools of grid disks on the cell. With ASM-scoped security, all database clients of that particular ASM cluster can access the grid disks. You can go further and configure database-scoped security to restrict the storage access at the database level.

Even if you intend to deploy database-scoped storage-level security, you will have to first configure ASM-scoped security. The following sections describe step-by-step procedures for how to enforce ASM-scoped and database-scoped security on the cell.

Configuring ASM-Scoped Security

In order to enforce ASM-scoped security on Exadata, follow this procedure:

1. Shut down ASM and all database instances on the Compute Node.
2. Generate a security key using the `CREATE KEY` command on a cell `CellCLI` prompt which will be used or copied across all cell servers to enforce the security:

```
CellCLI> CREATE KEY
```

3. Create a `cellkey.ora` file for ASM under `/etc/oracle/network-config` on the Compute Node, assign the security key against the ASM instance name, and change the permission and ownership as shown:

```
# cellkey.ora
  key=<key generated in the above command>
  asm=<asm db_unique_name>
  realm=<xyz> -- optional
# chown oracle:dba /etc/oracle/network-config/cellkey.ora
# chmod 640 /etc/oracle/network-config/cellkey.ora
```

4. If you want to change the realm name on the cell, use the following:

```
CellCLI> alter cell realmName=prod_realm
```

5. Assign the security key to the ASM instances across all the cell servers where you want to enforce the security:

```
CellCLI> ASSIGN KEY FOR '+ASM'=<security key>
```

6. Add or modify the grid disk's `availableTo` attribute to add the ASM instance, as follows:

```
CellCLI> list griddisk
```

7. After getting the grid disks' names, change the attribute for each grid disk:

```
CellCLI> alter griddisk <grid disk list> availableTo =' +ASM'
CellCLI> alter griddisk ALL availableTo=' +ASM'
```

8. Restart the ASM and database instances.

This type of security enables grid disk access to all the database clients of that ASM instance.

Configuring Database-Scoped Security

In order to enforce database-scoped security on Exadata, follow the next procedure:

1. Stop ASM and all database instances on the Compute Node.
2. Generate a separate new security key for each database connecting to the ASM instance with the `CREATE KEY` command using the `CellCLI` utility on the cell:

```
CellCLI> CREATE KEY
```

3. Create a `cellkey.ora` key file under `$ORACLE_HOME/admin/<db_name>/pfile` for each database on the Compute Node, assign the key against the database name, and change the permission and ownership as follows:

```
# cellkey.ora
key=<key generated in the above command>
asm=<asm db_unique_name>
realm=<xyz> -- optional
# chmod 640 $ORACLE_HOME/admin/<db_name>/pfile/cellkey.ora
```

4. If you want to change the realm name on the cell, use the following:

```
CellCLI> alter cell realmName=xyz
```

5. Assign the security key to the database on all the cell servers:

```
CellCLI> ASSIGN KEY FOR <DB_NAME1>=<security key1>',
<DB_NAME2>=<security key2>'
```

6. Change the grid disk's `availableTo` attribute, as follows:

```
CellCLI> list griddisk
```

7. After getting the grid disks' names, change the attribute for each grid disk:

```
CellCLI> alter griddisk <griddisk list> availableTo ='+ASM,
<DB_NAME>,<DB_NAME2>'
CellCLI> alter griddisk ALL availableTo='+ASM,<DB_NAME>,<DB_NAME2>'
```

8. Restart the ASM and database instances.

When you want to further restrict individual databases of that ASM instance to access a different pool of grid disks, you will have to enforce database-scoped security as explained previously.

You can list or view the existing security keys using the following command:

```
# CellCLI> list key
```

Exempting Cell Security

At any given time, you can exempt the cell security that has been imposed earlier. You can bring down the database-scoped security to ASM-scoped security and ASM-scoped security to the default open security. Keep in mind that any change and enforcement require ASM and database downtime.

The following procedure can be used to remove the cell security:

1. Stop ASM and databases.
2. Get the list of grid disk attributes assigned.
3. Exempt the databases from the security and remove Access Control List (ACL) setup using the following commands:

```
CellCLI> alter griddisk griddiskName availableTo='+ASM'
CellCLI> assign key for <DB_NAME>=''
```

4. Remove the cellkey.ora file from the respective database Home location.
5. Finally, verify the grid disks to ensure that the databases are exempted from the security list:

```
CellCLI> list griddisk attributes name,availableTo
```

6. The previous steps exempt the databases from database-scoped security. If you want to also remove ASM-scoped security, first use the following commands:

```
CellCLI> alter griddisk all availableTo=''
CellCLI> alter griddisk griddiskName availableTo=''
CellCLI> assign key for +ASM=''
```

7. Finally, remove the cellkey.ora file from /etc/oracle/cell/network-config.

Summary

To manage any environment effectively, it is always important for a DBA/DMA to understand the underlying architecture and the total functionality of the system. This chapter explained all the essentials and the importance of Exadata Storage Cells in detail. You have learned how to gracefully shut down Storage Cell background services, how to configure cell security, and how to create and manage storage on the cell.

This page intentionally left blank



Index

- _ (underscore)
 - hidden underscore parameters, 203
 - underscore parameters, 101–102
- 2u custom network switch space, 7
- 2-way waits, 375
- 3-way waits, 375
- 11.2.0.4 clusterware, upgrading and updating, 321–323
- "11gR2 Clusterware Grid Home," 58
- 12c. *See* OEM 12c; Oracle Database 12c.
- A**
- a `interconnected_quarterback` argument, 473
- AC (Application Continuity), 167–168
- ACFS (ASM Cluster File System)
 - configuring for RMAN backups, 202
 - migrating to Exadata, 275
- ACO (Advanced Compression Option), 164
- Active Session History (ASH), 57
- `activerequest` command, 70–71
- Actuator arms, 413
- Adaptive Plans optimization, 157–158
- Adaptive Query Optimization. *See also* Oracle Database 12c.
 - Adaptive Plans, 157–158
 - Automatic Re-optimization, 159
 - dynamic statistics, 159–164
 - incremental statistics, 160–161
 - join method, 157–158
 - no-workload statistics, 163–164
 - parallel data distribution, 157–158
 - PQ (parallel query) execution, 158
 - workload statistics, 161–163
- ADDM (Automatic Database Diagnostic Monitor), 57
- ADG (Active Data Guard), 45–46, 232
- ADO (Automatic Data Optimization), 164–167, 462–463
- ADR (Automatic Diagnostic Repository), 56
- ADRCI (Automatic Diagnostic Repository Command-line Interpreter), 56
- Ailamaki, Anastassia, 124
- `alertdefinition` command, 70–71
- `alerthistory` command, 70–71
- Alerts
 - history, displaying, 69–70
 - notifications, configuring mail server for, 68
 - troubleshooting RACs, 56
- `_allow_cell_smart_scan_attr` parameter, 101
- `alter cell shutdown services all [FORCE]` command, 68
- `alter cell startup services all` command, 68
- ALTER DATABASE ADD STANDBY LOGFILE command, 225
- `alter diskgroup` command, 319
- ALTER IORMPLAN attribute, 404
- ALTER IORMPLAN command, 385, 436–437
- AMM (Automated Memory Management), 47
- Anti virus software, best practices, 48
- Application Continuity (AC), 167–168

- Archive compression, 126
 - ARCHIVE HIGH compression, sample code, 127
 - ARCHIVE LOW compression, sample code, 127
 - ARCHIVE_LAG_TARGET parameter, 223
 - ARCHIVELOG mode, 46
 - ASH (Active Session History), 57
 - ASM (Automatic Storage Management)
 - benefits of, 74
 - disk groups, creating, 82
 - Flash-based ASM disk group, 448–450
 - overview, 8–9
 - performance tuning, 387–388
 - ASM (Automatic Storage Management), migrating to Exadata
 - choosing redundancy levels, 275
 - rebalance technique, 287
 - ASM Cluster File System (ACFS). *See* ACFS (ASM Cluster File System).
 - ASMM (Automatic Shared Memory Management), 47
 - ASM-scoped cell security, 90
 - Automated Memory Management (AMM), 47
 - Automatic archive switch, 223
 - Automatic Data Optimization (ADO), 164–167, 462–463
 - Automatic Database Diagnostic Monitor (ADDM), 57
 - Automatic Degree of Parallelism, 48
 - Automatic Diagnostic Repository (ADR), 56
 - Automatic Diagnostic Repository Command-line Interpreter (ADRCI), 56
 - Automatic Re-optimization, 159
 - Automating backups, 204–206
 - Avocent MergePoint Unity KVM switches, upgrading, 331
 - Avocent MergePoint Unity Switch plugin, 248
 - AWR (Automatic Workload Repository), 57
- B**
- Backup and restore
 - automating backups, 204–206
 - backup schedules, 213–214
 - backup with RMAN, 210–212
 - Block Change Tracking option, 46
 - database image copy backup, example, 207
 - dedicated 40Gb low-latency InfiniBand connections, 201–202
 - disk-to-disk backups. *See* RMAN (Recovery Manager).
 - examples, 206–209
 - incremental backup, example, 207
 - incremental backups, speeding up, 46
 - log files, 204–205
 - reverting a database to its original state, 243
 - .sql files, 205
 - standby databases. *See* Data Guard.
 - tape media, 202
 - ZFS Storage Appliance, 201–202
 - _backup_disk_bufcnt, 203
 - _backup_disk_bufsz, 203
 - _backup_file_bufcnt, 203
 - _backup_file_bufsz, 203
 - Balancing RAC databases, 383–386
 - BCT (block change tracking), enabling, 222
 - BDP (bandwidth-delay product), calculating, 218–220
 - Best practices
 - anti virus software, 48
 - bonded network interfaces, 46
 - CPU management, 47
 - eliminating SPOFs (single points of failure), 45–46
 - health checks, 46–47
 - logging, 46
 - memory management, 47
 - migrating to Exadata, 290–291
 - parallelization, 48
 - partitioning, 48
 - "RAC and Oracle Clusterware Best Practices Starter Kit," 58
 - for RACs. *See* RACs (Real Application Clusters), best practices.
 - resource management, 47
 - third-party tools and utilities, 48
 - tuning, 48
 - updating Exadata, 46
 - ZFS Storage Appliance, 352–355
 - Block Change Tracking option, 46
 - Block corruption, preventing, 46
 - Bonded network interfaces, best practices, 46
 - Books and publications
 - "11gR2 Clusterware Grid Home," 58
 - "A Case for Fractured Mirrors," 124
 - "A Decomposition Storage Model," 122
 - "Exadata Database Machine and Exadata Storage Server Supported Versions," 59
 - "Oracle Database (RDBMS) . . . Configuration Requirements . . .," 59
 - "RAC and Oracle Clusterware Best Practices Starter Kit," 58
 - "A Relational Model of Data for Large Shared Data Banks," 121
 - "Top 11gR2 Grid Infrastructure Upgrade Issues," 59
 - "Weaving Relations for Cache Performance," 124
 - BPs (bundle patches), applying, 217
 - BUI (Browser User Interface), 342–343
 - BZIP2 compression algorithm, 129
- C**
- c## databases, 215
 - C## databases, 215
 - c option, rman2disk.ksh script, 208
 - CA (channel adapters), 477

- cache-type attribute options, 115–117
- Caching data. *See* ESCF (Exadata Smart Flash Cache).
- "A Case for Fractured Mirrors," 124
- Category resource management, 404
- CDB (container database) backups, 215–216
- cdb\$root (root CDB), 170
- Cell disks
 - configuring, 81
 - creating, 76
 - definition, 75
 - description, 446–448
 - details, listing, 79–80
 - subdividing. *See* Grid disks.
- Cell monitoring, 77
- Cell Nodes
 - overview, 9
 - prolonged shutdown, 86
 - startup/shutdown, 85–87
 - updating Storage software, 299–300
 - upgrading, 312–315
- Cell Offloading
 - checking for, 103–105
 - CPU time statistics, 105–106
 - DB time statistics, 105–106
 - examples, 103–107
 - key statistics, 104–105
 - parameters, 101–102
 - performance tuning, 365–366
 - session wait event statistics, 105–106
 - sort reduction, 106–107
 - wait time statistics, 105–106
- Cell security
 - ASM-scoped, 90
 - database-scoped, 91
 - exempting, 92
 - overview, 89–90
 - SPUs (Security Path Updates), 304
- Cell Server (CELLSRV). *See* CELLSRV (Cell Server).
- Cell servers. *See also* CELLSRV (Cell Server); MS (Management Server); RS (Restart Server).
 - centralizing cell management, 63
 - details, displaying, 68–69. *See also* imagehistory command; imageinfo command.
 - managing, 82–83
 - range, configuring, 62
 - software version, displaying, 82–83
- Cell servers, troubleshooting
 - collecting diagnostics, 83–85
 - disk problems, 87–89
 - Exachk, 85
 - ExaWatcher, 84–85
 - SunDiag, 83–84
- Celladmin users, 77
- cellcli command, 312–313
- CellCLI (Control Command-Line Interface) utility
 - first Flash SSD completed, 436
 - first grid disk log write completed, 436
 - FL_DISK_FIRST metric, 436
 - FL_FLASH_FIRST metric, 436
 - FL_PREVENTED_OUTLIERS metric, 437
 - LIST FLASHLOG command, 436
 - listing Flash Log status, 436
 - optimized redo log writes, counting, 437
 - performance bug, 457
- cellesmmt process, 67–68
- _cell_fast_file_create parameter, 101
- _cell_fast_file_restore parameter, 101
- _cell_file_format_chunk_size parameter, 101
- CELL_FLASH_CACHE storage clause, in Smart Flash Cache
 - architecture, 423–425
 - compression, 426
 - full table scans, 430–431
 - Smart Scans, 429–430
- _cell_index_scan_enabled parameter, 102
- cellinit.ora file, 82
- Cell-level administration, 77
- Cellmonitor users, 77
- _cell_offload_capabilities_ enabled parameter, 102
- cell_offload_decryption parameter, 101
- _cell_offload_hybridcolumnar parameter, 102
- cell_offload_parameters parameter, 101
- _cell_offload_predicate_ reordering_ enabled parameter, 102
- cell_offload_processing parameter, 101
- _cell_offload_timezone parameter, 102
- _cell_offload_virtual_columns parameter, 102
- _cell_range_scan_enabled parameter, 102
- cellrsbmt process, 67–68
- cellrsomt process, 67–68
- cellrstrm process, 67–68
- CELLSRV (Cell Server). *See also* MS (Management Server); RS (Restart Server).
 - command-line interface, 66
 - FORCE option, 66
 - managing Smart Flash Cache, 423–425
 - overview, 65–66
 - stopping/starting/restarting, 66
 - troubleshooting, 65
- _cell_storidx_mode parameter, 102
- Central processing units (CPUs). *See* CPUs (central processing units).
- Certified Platinum Configuration, 294–295
- Change tracking, blocking, 222
- Channel adapter nodes, extracting, 478
- Channel adapters (CA), 477
- Chargeback, database consolidation, 392–393
- CheckHWnFWProfile utility, 85
- CHM (Cluster Health Monitor), 57

- Cisco switches
 - description, 7, 248
 - upgrading, force booting the switch, 331
- cleanup parameter, 299
- Client access networks, 189
- Clock, synchronizing across servers, 221
- Cloning snapshots
 - with Data Guard, 351
 - definition, 351
 - process for, 351
- Cloud Control 12c. *See* OEM 12c.
- Cluster overhead, 376–378
- Clusterware Control (CRSCTL) utility, 50
- Codd, Edgar F., 121
- Columnar storage models
 - data access perspective, 122
 - DML (Data Manipulation Language), 123
 - DSM (Decomposition Storage Model), 122–124. *See also* HCC (Hybrid Columnar Compression).
 - fine-grained hybrids, 124–125
 - Fractured Mirrors, 124
 - index storage perspective, 123
 - NSM (N-ary Storage Model), 122
 - PAX (Partition Attributes Across) model, 124
 - storage space perspective, 122–123
- Commands. *See* Tools and utilities; *specific commands*.
- Common user accounts, 215–216
- COMPRESS FOR ARCHIVE, 126
- COMPRESS FOR QUERY, 126
- Compression. *See also* HCC (Hybrid Columnar Compression).
 - levels, choosing for performance, 372
 - options for ZFS Storage Appliance, 354
 - Smart Flash Cache, 425–426
- Compression Units (CUs), 126, 129–131
- Compute Nodes
 - operating systems supported, 9
 - overview, 5–6
 - in Storage Server, 61
 - updating the, 315–319
- Configuration information
 - InfiniBand, monitoring, 190–194
 - InfiniBand network interface, displaying, 191
 - reviewing Exadata configuration settings, 306–307
 - upgrading a ZFS Storage Appliance, 333
 - validating, 467–469
- Configuration options. *See* Models and configuration options.
- Configuring
 - cell disks, 81
 - cell server range, 62
 - Flash grid disks, 81–82
 - mail server for alert notifications, 68
 - NFS shares, 348–349
 - tftp for Linux, 324–325
 - ZFS Storage Appliance, 333
- Connectivity, verifying, 479, 483
- Consolidated server pools, 391–392
- Consolidating databases. *See* PDBs (Pluggable Databases).
- Container database (CDB) backups, 215–216
- Control Command-Line Interface (CellCLI) utility.
 - See* CellCLI (Control Command-Line Interface) utility.
- CONVERT DATABASE, 285–286
- CONVERT DATAFILE, 285–286
- CONVERT TABLESPACE, 285
- Copeland, George, 122–123
- Copying. *See also* Cloning.
 - database image, backing up, 207
 - file system. *See* Snapshots.
 - LUNs (logical unit numbers). *See* Snapshots.
 - snapshots, 351–352
- Cost factors, migrating to Exadata, 281
- CPU_COUNT parameter, 47, 386, 399–401
- CPUs (central processing units)
 - cpuspeed (CPU speed) statistics, 162–163
 - cpuspeednw (CPU speed) statistics, 164
 - database consolidation settings, 399–405
 - isolation management, database consolidation, 408
 - management best practices, 47
 - prioritizing, 386
 - sizing, migrating to Exadata, 274
 - utilization benchmark, 274
- CPUs (central processing units), time statistics
 - Cell Offloading, 105–106
 - Flash Cache Keep option, 118
 - Smart Scans, 105–106
- cpuspeed (CPU speed) statistics, 162–163
- cpuspeednw (CPU speed) statistics, 164
- Creating
 - backups, 209–213
 - cell disks, 76
 - database tables, 100
 - Flash temporary tablespace, 453–456
 - Flash-based ASM disk groups, 448–450
 - grid disks, 81, 449
 - an ILOM, 249
 - NFS shares, 343–345
 - PDBs (Pluggable Databases), 170–177
 - standby databases, 235–238
- crsctl check command, 51
- crsctl get command, 51
- crsctl query command, 50–51
- crsctl status command, 51
- CRSCTL (Clusterware Control) utility, 50
- Current active image, querying, 308–309
- Current release version, checking, 310–311
- CUs (Compression Units), 126, 129–131
- Custom *vs.* third-party applications, migrating to Exadata, 278

- D**
- d option, `rman2disk.ksh` script, 208
 - Data compression. *See* HCC (Hybrid Columnar Compression).
 - Data corruptions, Far Sync, 234–235
 - Data deduplication, ZFS Storage Appliance, 354
 - Data Guard
 - ADG (Active Data Guard), 232
 - automatic archive switch, 223
 - BCT (block change tracking), enabling, 222
 - BDP (bandwidth-delay product), calculating, 218–220
 - BPs (bundle patches), applying, 217
 - change tracking, blocking, 222
 - clock, synchronizing across servers, 221
 - cloning snapshots, 351
 - FRA (Fast Recovery Area), 222–223
 - network queue size, adjusting, 220–221
 - NTP (Network Time Protocol), enabling, 221
 - patches, applying, 217
 - PEMS (parallel execution message size), 223–224
 - read-only database for reporting purposes, 232
 - redoing transport overhead. *See* Far Sync.
 - reopen option, 228–229
 - RTA (Real-Time Apply), 227–228
 - SDUs (session data units), setting, 217–218
 - standby file management, 231
 - standby-first patching, 231
 - switchover considerations, 242–243
 - TCP Nagle algorithm, enabling/disabling, 221
 - timeout option, 228–229
 - troubleshooting, 242–243
 - wait time for reconnection, setting, 228–229
 - Data Guard, applying changes
 - delaying, 228–229
 - in real time, 227–228
 - Standby-First Patch Apply, 231–232
 - standby-first patching, 231–232
 - Data Guard, Far Sync feature
 - archive logs, retention policy, 233–234
 - Cloud Control 12c, 241–242
 - creating an instance of, 233
 - data corruptions, 234–235
 - DG Broker, configuring, 239–241
 - failover to a standby database, 240–241
 - instantiating Data Guard, 235–238
 - overview, 233
 - standby databases, creating, 235–238
 - Data Guard, logging
 - archive generation rate, 229–230
 - flashback, 227
 - forcing, 226–227
 - log switching, forcing, 223
 - SRLs (standby redo logs), 224–226
 - Data Guard Standby-First certified patches, 302
 - Data Guard-based migration, 283–284
 - Data management. *See* ILM (Information Lifecycle Management).
 - Data Manipulation Language (DML), 123, 140–144
 - Data Pump-based migration, 282
 - Data reorganization and restructuring, migrating to Exadata, 281
 - Data replication tools, migrating to Exadata, 283–284
 - Data warehousing, with HCC, 147–148
 - Database 12c. *See* Oracle Database 12c.
 - Database administrators (DBAs), effects of Exadata on job roles, 4
 - Database block size, changing, 388
 - Database connection string, uploading, 469
 - Database consolidation. *See also* Schema consolidation.
 - instance consolidation, 390
 - models, 389–390
 - PDBs (Pluggable Databases), 169–177, 410–411
 - planning, 390
 - schema consolidation, 390
 - sizing requirements, evaluating, 393–394
 - steering committee, 390
 - ZFS Storage Appliance, 341
 - Database consolidation, grouping applications
 - chargeback, 392–393
 - overview, 391
 - server pools, 391–392
 - Database consolidation, isolation management
 - CPU, 408
 - fault isolation, 406
 - I/O, 408
 - memory, 408
 - operational isolation, 406–407
 - overview, 405
 - patching, 407
 - resource isolation, 408
 - security isolation, 409
 - Database consolidation, setting up
 - balancing latency and throughput, 403–404
 - category resource management, 404
 - CPU settings, 399–405
 - CPU_COUNT parameter, 399–401
 - database resource management, 399–401
 - Flash Log, enabling/disabling, 404
 - Instance Caging, 399–401
 - I/O settings, 394–398
 - IORM (I/O Resource Management), 401–405
 - limiting disk utilization, 404
 - memory settings, 398–399
 - Smart Flash Cache, enabling/disabling, 404
 - storage settings, 394–398
 - Database File System (DBFS). *See* DBFS (Database File System).
 - Database Flash Cache (DBFC) *vs.* Smart Flash Cache, 421–422
 - Database image copy backup, example, 207

- Database machines, discovering. *See* Exadata Database Machine discovery.
- Database Resource Management (DBRM), 47, 385–386
- Database resource management, database consolidation, 399–401
- Database server, 72–74
- Database tables. *See* Tables, database.
- Database-scoped cell security, 91
- DB Compute Nodes. *See* Compute Nodes.
- DB time statistics
 - Cell Offloading, 105–106
 - Flash Cache Keep option, 118
 - Smart Scans, 105–106
- DBAs (Database administrators), effects of
 - Exadata on job roles, 4
- DB_BLOCK_CHECKING parameter, 46, 234–235
- DB_BLOCK_CHECKSUM parameter, 46, 234–235
- DBFC (Database Flash Cache) *vs.* Smart Flash Cache, 421–422
- DB_FLASHBACK_RETENTION_TARGET parameter, 223
- DBFS (Database File System)
 - configuring for RMAN backups, 202
 - eliminating with ZFS Storage Appliance, 341
 - recovering space and resources with ZFS Storage Appliance, 341
- DB_KEEP_CACHE_SIZE parameter, 224
- DB_LOST_WRITE_PROTECT parameter, 46, 234–235
- DBMS_COMPRESSION package, 140–144
- DBMS_STATS.GATHER_SYSTEM_STATS package, 162–163
- DB_RECOVERY_FILE_DEST parameter, 222–223
- DB_RECOVERY_FILE_DEST_SIZE parameter, 222–223
- DB_RECYCLE_CACHE_SIZE parameter, 224
- DBRM (Database Resource Management), 47, 385–386
- dccli command
 - definition, 63
 - listing IORM objective, 405
 - upgrading Cell Nodes, 312–313
- Decomposition Storage Model (DSM), 122–124. *See also* HCC (Hybrid Columnar Compression). "A Decomposition Storage Model," 122
- Default system users, 77
- Deleting files automatically, 67
- Dell software tools, monitoring storage cells, 484–487
- Dell Toad suite, monitoring storage cells, 484–487
- Deploying
 - agents, 246
 - Oracle Management Agent, 250–254
 - plugins, manually, 249
- Description of Exadata. *See* Exadata overview.
- DeWitt, David, 124
- DG Broker
 - configuring, 239–241
 - GRP (guaranteed restore point), 244
 - reverting a database to its original state, 243
 - switchover considerations, 242–243
 - switchover tracing, 243
- DG Broker, configuring, 239–241
- dg_duplicate_database.ksh script, 236–238
- diagcollection.pl script, 57
- Diagnostic data, storing, 56
- Diagnostic information, collecting
 - ADR (Automatic Diagnostic Repository), 56
 - ADRCI (Automatic Diagnostic Repository Command-line Interpreter), 56
 - troubleshooting RACs, 56
- DIAGNOSTIC_DEST init.ora parameter, 56
- Diagnostics, collecting, 83–85
- dir command, 325
- _disable_cell_optimized_backups parameter, 102
- _DISABLE_REBALANCE_COMPACT parameter, 313
- disable_sm command, 470
- Disabling indexes, 370–372
- Discovering
 - database machines. *See* Exadata Database Machine discovery.
 - InfiniBand topology, 477–478
 - node-to-node connectivity, 192
- Disk architecture, Exadata *vs.* non-Exadata, 74–77
- DiskInfo component, collecting statistics about, 84
- DISK_REPAIR_ATTRIBUTE, adjusting, 86
- Disks
 - conventional. *See* Spinning disks.
 - Flash technology. *See* Flash SSD.
 - levels, listing, 77–80
 - troubleshooting, 87–89
 - utilization, limiting for database consolidation, 404
- Disk-to-disk backups. *See* RMAN (Recovery Manager).
- Displaying. *See also* Listing.
 - alert history, 69–70
 - cell server details, 68–69. *See also* imagehistory command; imageinfo command.
 - cell server software version, 82–83
 - driver information, 477, 478–479
 - InfiniBand driver information, 477, 478–479
 - software images, 475–476
 - storage cell metrics, 69–70
 - storage cell server details, 68–69
 - Storage Server alert history, 69–70
- DML (Data Manipulation Language), 123, 140–144
- dNFS, ZFS Storage Appliance, 348–349
- Double mirroring, 9, 45
- Downtime considerations, migrating to Exadata, 280
- Driver information, displaying, 477, 478–479
- Driver information, retrieving, 190
- Dropping PDBs (Pluggable Databases), 173–177
- DSM (Decomposition Storage Model), 122–124. *See also* HCC (Hybrid Columnar Compression).

- Duplicate node and port GUID validation, 193
- Duplicate Target Database For Standby
 - Database command, 235–238
- DUPLICATE TARGET DATABASE FOR STANDBY
 - FROM ACTIVE DATABASE command, 225
- Dynamic statistics, 159–164
- E**
- Economics of Flash SSD, 415–417
- Eighth Rack, 62–63
- 11.2.0.4 clusterware, upgrading and updating, 321–323
- "11gR2 Clusterware Grid Home," 58
- emctl listplugins agent command, 249
- enable command, 325
- enable_sm command, 470
- Endurance of Flash SSD, 418–419
- Engineered database machine, 3
- env_test command, 472
- ESCF (Exadata Smart Flash Cache)
 - architecture, 423–425
 - caching data, 115–119
 - caching data, example script, 116
 - caching eligibility, 423
 - compression, enabling, 425
 - contents, 425
 - vs. Database Flash Cache, 421–422
 - description, 96
 - disks, listing, 78
 - enabling/disabling for database consolidation, 404
 - evolution of, 94–95
 - vs. Flash-based grid disks, 76–77
 - grid disks, configuring, 81–82
 - and indexes, 366
 - KEEP option, effects of, 115–119
 - leveraging for performance, 387
 - monitoring, 427–429
 - new features, 183
 - performance gains, 94–95
 - populating, 97–98
 - prioritizing blocks, 426–427
 - purpose of, 94
 - vs. Storage Server, 94–95
- ESCF (Exadata Smart Flash Cache), database-level features
 - Cell Offloading, 101–107
 - database initialization parameters, 101–102
 - populating data, example, 100
 - Smart Scans, 101–107
 - Storage Indexes, 107–115
 - system statistics, 100–101, 117
 - table creation, example, 100
- ESCF (Exadata Smart Flash Cache), performance
 - CELL_FLASH_CACHE KEEP overhead, 431–432
 - full table scans, 430–431
 - index lookups, 432–433
 - monitoring, 428–429
 - polluting the cache, 430
 - Smart Scans, 428–429
 - write-back cache, 442–443
- ESCF (Exadata Smart Flash Cache), vs. Flash-based tablespace
 - ADO (Automatic Data Optimization), 462–463
 - creating a Flash temporary tablespace, 453–456
 - ILM (Information Lifecycle Management), 462–463
 - index fetch performance, 451–452
 - performance bug, 457
 - redo logs, 456–458
 - scan performance, 452–453
 - storage tiering solutions, 458–462
 - tiering data with partitions, 459–462
- ESCF (Exadata Smart Flash Cache), write-back cache
 - architecture, 441–442
 - enabling/disabling, 442
 - I/O bottlenecks, 440–441
 - overview, 439
 - performance, 442–443
- Ethernet Cisco switches
 - description, 7, 248
 - upgrading, 323–331
 - upgrading, force booting the switch, 331
- Ethernet Cisco switches, upgrading
 - configuring tftp for Linux, 324–325
 - confirming available space, 325–326
 - downloading tftp for Linux, 324–325
 - including boot firmware, 326–327
 - installing tftp for Linux, 324–325
 - user name and password, 329–330
 - verifying user access, 327–331
- Ethernet-channeled network interfaces, 46
- Exachk: Exadata Health Check utility
 - Collection Manager requirements, 469
 - description, 46–47
 - downloading, 46, 85
 - loading results into a repository, 469
 - new features, 468–469
 - overview, 467–468
 - reviewing Exadata configuration settings, 306–307
 - sample code, 468
 - troubleshooting cell servers, 85
- Exadata components. *See specific components.*
- "Exadata Database Machine and Exadata Storage Server Supported Versions," 59
- Exadata Database Machine discovery
 - administering discovered machines, 262–265
 - deploying Oracle Management Agent, 250–254
 - discovering database machines, 250–260
 - exadataDiscoveryPreCheck.pl script, 250
 - monitoring and managing discovered machines, 260–262
 - post-discovery tasks, 260
 - prerequisite checks, 250

- Exadata overview. *See also specific components.*
 - diagnostics. *See* Tools and utilities, Exadata diagnostics; Troubleshooting.
 - effect on DBA job roles, 4
 - engineered database machine, 3
 - OEM 12c, 4–5
 - system description, 2–3
 - Exadata Smart Flash Cache (ESCF). *See* ESCF (Exadata Smart Flash Cache).
 - Exadata X3 In-Memory Database Machine, 14
 - exadataDiscoveryPreCheck.pl script, 250
 - ExaWatcher utility, 84–85. *See also* OSWatcher logs.
 - ExaWatcherCleanup module, 85
 - Exempting cell security, 92
- F**
- Failover
 - FCF (Fast Connection Failover), 167–168
 - setting up, 46
 - to a standby database, 240–241
 - TAF (Transparent Application Failover), 167–168
 - fal.client parameter, 238
 - Far Sync
 - archive logs, retention policy, 233–234
 - Cloud Control 12c, 241–242
 - creating an instance of, 233
 - data corruptions, 234–235
 - DG Broker, configuring, 239–241
 - failover to a standby database, 240–241
 - instantiating Data Guard, 235–238
 - overview, 233
 - standby databases, creating, 235–238
 - Fast Connection Failover (FCF), 167–168
 - Fast Recovery Area (FRA), 222–223
 - fattree argument, 473
 - Fault isolation, 406
 - FCF (Fast Connection Failover), 167–168
 - File system, copying. *See* Snapshots.
 - file systems command, 325
 - File utilization notification, 67
 - Fine-grained hybrids, columnar storage models, 124–125
 - Flash Cache. *See* ESCF (Exadata Smart Flash Cache).
 - Flash Cache Logging, 98
 - Flash Cache Write Back mode, 97–98
 - Flash Cache Write Through mode, 97–98
 - Flash Log, enabling/disabling, 404
 - Flash Modules, 73
 - Flash Recovery Area. *See* FRA (Fast Recovery Area).
 - Flash SSD
 - amount available, 445
 - architecture, 417–420
 - description, 415
 - economics of, 415–417
 - endurance, 418–419
 - in the Exadata architecture, 422
 - free lists, 419–420
 - garbage collection, 419–420
 - latency, 415
 - MLC (multi-level cell) disks, 417–418
 - overprovisioning, 419–420
 - page and block structure, 418–419
 - PCLe SSDs, 420
 - performance, 417–420
 - SATA SSDs, 420
 - SLC (single-level cell) disks, 417–418
 - Smart Flash Cache *vs.* Database Flash Cache, 421–422
 - TLC (triple-level cache) disks, 417–418
 - wear leveling, 419–420
 - write performance, 418–419
 - Flash SSD, as grid disks
 - cell disks, description, 446–448
 - Flash-based ASM disk group, 448–450
 - grid disks, description, 446–448
 - overview, 445–446
 - Flash SSD, *vs.* spinning disks
 - actuator arms, 413
 - disk architecture, 413–414
 - limitations of disks, 413–415
 - Moore's Law, 414
 - platters, 413
 - rotational latency, 414
 - seek time, 414
 - seek times, by drive technology, 416
 - short stroking, 415
 - storage economics, by drive technology, 416
 - striping, 415
 - transfer time, 414
 - Flash technology. *See* Flash SSD.
 - Flashback logging, 227
 - Flashcache parameter, 404
 - Flashlog parameter, 404
 - FL_DISK_FIRST metric, 436
 - FL_FLASH_FIRST metric, 436
 - FL_PREVENTED_OUTLIERS metric, 437
 - FORCE LOGGING mode, 46
 - FORCE option, 66
 - Forcing
 - log switching, 223
 - logging, 226–227
 - FRA (Fast Recovery Area), 222–223
 - Fractured Mirrors, 124
 - Free lists, 419–420
 - Full Rack, 62–63
 - Full table scans, 430–431
- G**
- Garbage collection, 419–420
 - gc cr/current block 2-way wait event, 378
 - gc cr/current block 3-way wait event, 378
 - gc cr/current block busy wait event, 378

- gc cr/current block congested wait event, 378
 - gc cr/current block lost wait event, 378
 - gc cr/current grant 2-way wait event, 378
 - gc cr/current multi block request wait event, 378
 - generateApplySteps command, 300–302
 - GetExaWatcherResults.sh script, 84
 - getmaster command, 470–472
 - Global Cache
 - interconnect latency, 379
 - latency, reducing, 378–380
 - LMS latency, 381–382
 - requests, 374–375
 - wait events, 378
 - Global Cache Fusion, 41, 42
 - Global Cache Service (LMS), 379
 - Grants, 375
 - Grid disks. *See also* Cell disks.
 - assigning performance characteristics, 76
 - creating, 81, 449
 - definition, 75
 - description, 446–448
 - details, listing, 80
 - Flash Cache vs. Flash-based, 76–77
 - Grid Home, updating, 319–323
 - Grid Infrastructure Management Repository, 57
 - GRP (guaranteed restore point), 244
 - GV\$ dynamic views, 71
- H**
- h option, patchmgr tool, 299
 - Half Rack, 62–63
 - Hardware architecture, networking fabric. *See* InfiniBand.
 - Hardware architecture, overview
 - 2u custom network switch space, 7
 - Cisco switch, 7
 - Compute Nodes, 5–6
 - InfiniBand, 6–7
 - naming scheme, 5
 - PDUUs (Power Distribution Units), 7
 - server layer, 5–6
 - shared storage, 6
 - storage cells, 6
 - Hardware component failure sensors, checking manually, 191
 - Harrison, Guy, 491
 - HCC (Hybrid Columnar Compression). *See also* DSM (Decomposition Storage Model).
 - Archive compression, 126
 - ARCHIVE HIGH compression, sample code, 127
 - ARCHIVE LOW compression, sample code, 127
 - COMPRESS FOR ARCHIVE, 126
 - COMPRESS FOR QUERY, 126
 - compressed CU sections, 126
 - compression algorithms, 127–129
 - compression methods, 125
 - compression ratios, 127–129
 - compression types, 129–131
 - CUs (Compression Units), 126, 129–131
 - for data warehousing, 147–148
 - DBMS_COMPRESSION package, 140–144
 - DML (Data Manipulation Language), 140–144
 - for Information Lifecycle Management, 147–148
 - locking, 144–146
 - OLTP compression, sample code, 127
 - within Oracle databases, 125
 - overview, 125–127
 - QUERY HIGH compression, sample code, 127
 - QUERY LOW compression, sample code, 127
 - tokenization, 125
 - uncompressed CU sections, 126
 - uses for, 147–148
 - Warehouse compression, 126
 - HCC (Hybrid Columnar Compression), performance
 - bulk load operations, 132–135
 - bulk read I/O operations, 135–137
 - small I/O operations, 137–139
 - Health checks. *See also* Exachk: Exadata Health Check utility.
 - best practices, 46–47
 - troubleshooting RACs, 55–56
 - Heat Map, 164–167
 - High availability upgrades, 305–306
 - Hill, Mark, 124
 - Home software, patching, 298–299
 - HP hardware, running Exadata, 10
 - Hybrid Columnar Compression (HCC). *See* HCC (Hybrid Columnar Compression).
- I**
- IBCardino component, collecting statistics about, 84
 - ibcheckerrors command, 482–483
 - ibchecknet command, 483
 - ibchecknode command, 483
 - ibcheckport command, 483
 - ibcheckstate command, 192, 481–482
 - ibclearcounters command, 484
 - ibclearerrors command, 484
 - ibdiagnet command, 193
 - ibhosts command, 191, 477
 - iblinkinfo command, 479–481
 - ibnetdiscover command, 192
 - ibping command, 479
 - ibqueryerrors command, 483
 - ibqueryerrors.pl command, 192
 - ibroute command, 471–472
 - ibstat command, 477
 - ibstatus command, 190, 478–479
 - ibswitches command, 192, 478
 - ibswitches parameter, 300
 - ibswitch_precheck parameter, 300
 - ignore_alerts parameter, 300

- ILM (Information Lifecycle Management)
 - ACO (Advanced Compression Option), 164
 - ADO (Automatic Data Optimization), 164–167
 - with HCC, 147–148
 - Heat Map, 164–167
 - overview, 164–167
 - Smart Flash Cache, *vs.* Flash-based tablespace, 462–463
 - ZFS Storage Appliance, 341
- ILOM (Integrated Lights Out Manager). *See also* InfiniBand.
 - creating, 249
 - overview, 195–197
 - plugin, 248
 - targets, 248
- Image 11.2.2.3
 - troubleshooting, 318–319
 - upgrading, 317–318
- Image 11.2.2.4.2, upgrading, 316–317
- imagehistory command
 - description, 475–476
 - managing cell servers, 82–83
- imageinfo command
 - description, 475–476
 - managing cell servers, 82–83
 - sample output, 476
- inactive command, 404
- Incremental backups. *See also* Backup and restore.
 - example, 207
 - with RMAN, 202
 - speeding up, 46
- Incremental statistics, 160–161
- Indexes
 - designing for new applications, 367–368
 - disabling, 370–372
 - identifying redundant, disused, or unnecessary, 369–370
 - and Smart Flash Cache, 366
 - storage, 365
- Indexing strategy, existing applications, 368–372
- Industry use cases, ZFS Storage Appliance, 355
- InfiniBand
 - driver information, displaying, 477, 478–479
 - driver information, retrieving, 190
 - duplicate node and port GUID validation, 193
 - hardware component failure sensors, checking manually, 191
 - log files, 194
 - monitoring settings and configuration, 190–194
 - network interface configuration, displaying, 191
 - network layout, verifying, 191
 - network-related issues, verifying, 194
 - networks, 189
 - node-to-node connectivity, discovering, 192
 - overview, 6–7
 - port health status, querying, 192
 - role of, 186–187
 - routing tables, checking, 471–472
 - storage network, 61
 - subnet manager master information, verifying, 192
 - switch management. *See* ILOM (Integrated Lights Out Manager).
 - switches, updating, 319
 - topology, discovering, 477–478
- InfiniBand, switches
 - ILOM (Integrated Lights Out Manager), 195–197
 - leaf switches, 64
 - monitoring and managing, 195–197
 - spine switches, 64
 - switch software, updating, 299–300
- InfiniBand Network Diagnostics
 - a interconnected_quarterback argument, 473
 - CA (channel adapters), 477
 - channel adapter nodes, extracting, 478
 - checking InfiniBand routing tables, 471–472
 - connectivity, verifying, 479, 483
 - disable_sm command, 470
 - enable_sm command, 470
 - env_test command, 472
 - fattree argument, 473
 - getmaster command, 470–472
 - ibcheckerrors command, 482–483
 - ibchecknet command, 483
 - ibchecknode command, 483
 - ibcheckport command, 483
 - ibcheckstate command, 481–482
 - ibclearcounters command, 484
 - ibclearerrors command, 484
 - ibhosts command, 477
 - iblinkinfo command, 479–481
 - ibping command, 479
 - ibqueryerrors command, 483
 - ibroute command, 471–472
 - ibstat command, 477
 - ibstatus command, 478–479
 - ibswitches command, 478
- InfiniBand driver information, displaying, 477, 478–479
- InfiniBand topology, discovering, 477–478
- infinicheck utility, 473–475
- LID (local identifier), 477
- node connectivity, checking, 483
- OpenSM, 469–470
- overall switch health, checking, 472
- overview, 469–470
- Performance Manager error counters, clearing, 482–483
- ping test, 479
- port connectivity, checking, 483
- port counters, 484
- port link information, reporting, 479–481

- port state, reporting, 481–482
- quarterdeck argument, 473
- router nodes, extracting, 478
- sample code, 470–472
- setsmpriority command, 470–472
- subnet management, 469–470
- subnet priority, setting, 470
- switch nodes, extracting, 478
- torus argument, 473
- verifying InfiniBand topology, 472–475
- verify_topology command, 472–473
- InfiniBand Switches agent, 248
- infinicheck utility, 194, 473–475
- Information Lifecycle Management (ILM). *See* ILM (Information Lifecycle Management).
- init.ora parameters, 48
- Instance Caging, 386, 399–401
- Instance consolidation, 390
- Integrated Lights Out Manager (ILOM). *See* ILOM (Integrated Lights Out Manager).
- I/O
 - isolation management, database consolidation, 408
 - logical, 359
 - physical, 359
 - prioritizing, 385–386
 - seek time (*ioseektim*) statistics, 164
 - setting up database consolidation, 394–398
 - sizing, migrating to Exadata, 272–273
 - transfer speed (*iotfrspeed*) statistics, 164
- IORM (I/O Resource Management). *See also* Resource management.
 - balancing RAC database workloads, 385–386
 - disabling on a per-cell basis, 404
 - setting up database consolidation, 401–405
- ioseektim* (I/O seek time) statistics, 164
- Iostat component, collecting statistics about, 84
- iotfrspeed* (I/O transfer speed) statistics, 164
- ipmitool* command, 248
- Isolation management. *See* Database consolidation, isolation management; Schema consolidation, isolation management.
- IT structure and strategy, migrating to Exadata, 270
- J**
- join method, 157–158
- K**
- _kcfis_cell_passthru_enabled* parameter, 102
- _kcfis_cell_passthru_fromcpu_enabled* parameter, 102
- _kcfis_disable_platform_decryption* parameter
 - Cell Offloading, 102
 - Smart Scan, 102
 - Storage Indexes, 109
 - _kcfis_io_prefetch_size* parameter, 102
 - _kcfis_kept_in_cellfc_enabled* parameter, 102
 - _kcfis_large_payload_enabled* parameter, 102
 - _kcfis_nonkept_in_cellfc_enabled* parameter, 102
 - _kcfis_storageidx_diag_mode* parameter, 109
 - _kcfis_storageidx_disabled* parameter, 109
- KEEP clause, 426–427
- KEEP option, effects of, 115–119
- Khoshafian, Setrag, 122–123
- KVM agent, 249
- KVM switches, upgrading, 331
- L**
- l option, *rman2disk.ksh* script, 208
- Latency
 - balancing with throughput, 403–404
 - Flash SSD, 415
 - ZFS Storage Appliance, 353
- Leaf switches, 64
- LID (local identifier), 477
- Limit parameter, 404
- Linux, 9
- LIST FLASHLOG command, 436
- Listing. *See also* Displaying.
 - cell disk details, 79–80
 - disk levels, 77–80
 - Flash Cache disks, 78
 - Flash Log status, 436
 - grid disk details, 80
 - LUN details, 78–79
 - physical disk details, 79
- LMS (Global Cache Service), 379
- LMS (Lock Management Server), 41
- Load balancing
 - Private Cluster Interconnect, 42
 - RACs, 41
- Lock Management Service. *See* LMS (Global Cache Service).
- Locking HCC, 144–146
- Log files
 - checking, 56
 - commonly used, 87
 - from ExaWatcher, 84
 - InfiniBand, 194
 - MS log, 87
 - OS messages, 87
 - OSWatcher, 87
 - related to cell patching, 87
- LOG_ARCHIVE_TRACE parameter, 243
- Logging. *See also* Smart Flash Logging.
 - best practices, 46
 - Far Sync archive logs, retention policy, 233–234
 - log file naming conventions, 204–205
 - redo logs, Smart Flash Cache *vs.* Flash-based tablespace, 456–458
 - SRLs (standby redo logs), 224–226

- Logging, with Data Guard
 - archive generation rate, 229–230
 - flashback, 227
 - forcing, 226–227
 - log switching, forcing, 223
 - SRLs (standby redo logs), 224–226
- Logical corruptions, preventing, 46
- Logical I/O, 359
- Logical migration *vs.* physical, 281, 284
- LUNs (logical unit numbers)
 - copying. *See* Snapshots.
 - description, 74–77
 - details, listing, 78–79
- LZJB (lzjb) compression, 354
- LZO compression algorithm, 128–129
- M**
- m option, `rman2disk.ksh` script, 208
- MAA (Maximum Availability Architecture), 45–46
- Management networks, 189
- Management Server (MS). *See* MS (Management Server).
- MAX_CONNECTIONS parameter, 228
- maxthr (maximum system throughput) statistics, 162–163
- mbrc (multiblock count) statistics, 162–163
- Memory
 - isolation management, database consolidation, 408
 - settings for database consolidation, 398–399
 - sizing, migrating to Exadata, 274
- Memory management
 - AMM (Automated Memory Management), 47
 - ASMM (Automatic Shared Memory Management), 47
 - best practices, 47
- metriccurrent command, 69–70
- metricdefinition command, 69–70
- metrichistory command, 69–70
- Migrating
 - databases across servers, 284–286
 - partitions, 155–157
 - tablespace data files, 155–157
- Migrating to Exadata
 - best practices, 290–291
 - character set changes, 290
 - compression type, choosing, 291
 - dropping indexes, 290
 - phases of, 268. *See also specific phases.*
 - RAT (Real Application Testing), 291
- Migrating to Exadata, architectural strategy
 - ACFS (ASM Cluster File System) for database storage, 275
 - ASM redundancy levels, choosing, 275
 - CPU sizing, 274
 - CPU utilization benchmark, 274
 - default features, 271
 - I/O sizing, 272–273
 - IT structure and strategy, 270
 - meeting SLAs, 269–270
 - memory sizing, 274
 - Oracle Exadata Simulation, 276–277, 291
 - overview, 268
 - performance simulation, 275–277, 291
 - POC (proof-of-concept) testing, 271
 - POV (proof-of-value) testing, 271
 - primary milestones and tasks, 269
 - specifically enabled features, 271
 - spindle types, choosing, 275
 - storage volume sizing, 275–277
- Migrating to Exadata, migration testing
 - backup and restore strategy, 288–289
 - documentation, 290
 - Exachk utility, 289
 - monitoring and alerting, 289
 - overview, 287–288
 - patch deployment documentation, 290
 - patch promotion lifecycle, 289
 - patch testing and validation, 290
 - patching, 289–290
 - POC (proof-of-concept) testing, 271
 - post-go-live, 289–290
 - POV (proof-of-value) testing, 271
 - RAT (Real Application Testing), 291
 - TFA (Trace File Analyzer), 289
- Migrating to Exadata, planning and design
 - accounting for paradigm change, 279–280
 - architectural characteristics, 280
 - ASM rebalance technique, 287
 - complexity issues, 281
 - CONVERT DATABASE, 285–286
 - CONVERT DATAFILE, 285–286
 - CONVERT TABLESPACE, 285
 - cost factors, 281
 - custom *vs.* third-party applications, 278
 - Data Guard-based migration, 283–284
 - Data Pump-based migration, 282
 - data reorganization and restructuring, 281
 - data replication tools, 283–284
 - determining migration strategies, 280–287
 - downtime considerations, 280
 - Exadata features, choosing, 279
 - migrating databases across servers, 284–286
 - migration with CTAS/IIS, 282–283
 - overview, 277
 - physical migration *vs.* logical, 281, 284
 - physical standby databases, 284
 - process description, 286
 - transportable tablespaces, 284–286
 - tuning third-party applications, 278
 - vendor certification, 278
- Migration with CTAS/IIS, 282–283
- Mirroring, 9, 45
- MLC (multi-level cell) disks, 417–418

- Models and configuration options, overview
 - Exadata on HP hardware, 10
 - hardware progression, 33–35
 - historical synopsis, 10
 - storage cells, 33
 - Storage Expansion Racks, 31–33
 - SuperCluster M6-32, 29–31
 - SuperCluster T4-4, 25–27
 - SuperCluster T5-8, 27–29
- Monitoring
 - architecture, 246–247
 - discovered machines, 260–262
 - InfiniBand settings and configuration, 190–194
 - network interfaces. *See* OEM 12c.
 - physical links, 190–194
 - Smart Flash Cache, 427–429
 - switches, InfiniBand. *See* OEM 12c.
- Monitoring storage cells, tools and utilities
 - Dell software tools, 484–487
 - Dell Toad suite, 484–487
 - OEM (Oracle Enterprise Manager), 487–491
 - SQL Optimizer, 486–487
- Moore, Gordon, 414
- Moore's Law, 414
- Mounting NFS shares, 348–349
- Moving. *See* Migrating.
- `mreadtim` (multiblock) read times, 162–163
- MS (Management Server). *See also* CELLSRV (Cell Server); RS (Restart Server).
 - deleting alert files automatically, 67
 - deleting files automatically, 67
 - file utilization notification, 67
 - overview, 66–67
 - service status, printing, 67
 - stopping/starting/restarting, 67
- MS log files, 87
- Multiblock count (`mbrcc`) statistics, 162–163
- Multi-level cell (MLC) disks, 417–418
- Multitenant architecture. *See* PDBs (Pluggable Databases).
- N**
 - `-n` option, `rman2disk.ksh` script, 208
 - Naming conventions, Oracle patches, 304
 - Naming scheme for Exadata hardware, 5
 - N-ary Storage Model (NSM), 122
 - `Netstat` component, collecting statistics about, 84
 - `net_timeout` option, 228–229
 - Network architecture, 187–188
 - Network interfaces, monitoring and managing. *See* InfiniBand; OEM 12c.
 - Network Time Protocol (NTP), enabling, 221
 - Networking
 - client access networks, 189
 - components, 185–186
 - InfiniBand networks, 189
 - management networks, 189
 - physical link monitoring, 190–194
 - resource management, new features, 183
 - setup requirements, 188–189
 - troubleshooting tools and utilities, 190–194
 - Networking fabric. *See* InfiniBand.
 - Network-related issues, verifying, 194
 - Networks
 - interface configuration, displaying, 191
 - layout, verifying, 191
 - queue size, adjusting, 220–221
 - NFS services, enabling in ZFS Storage Appliance, 345–348
 - NFS shares
 - creating, 343–345
 - mounting, 348–349
 - Node connectivity, checking, 483
 - Node-to-node connectivity, discovering, 192
 - No-workload statistics, 163–164
 - NSM (N-ary Storage Model), 122
 - NTP (Network Time Protocol), enabling, 221
- O**
 - OEM 12c
 - Avocent MergePoint Unity Switch plugin, 248
 - Cisco switch plugin, 248
 - creating an ILOM, 249
 - deploying agents, 246
 - `emctl listplugins agent` command, 249
 - with Far Sync, 241–242
 - ILOM targets, 248
 - InfiniBand Switches agent, 248
 - KVM agent, 249
 - manual plugin deployment, 249
 - monitoring architecture, 246–247
 - monitoring storage cells, 487–491
 - network interfaces, monitoring and managing, 197–199
 - Oracle ILOM plugin, 248
 - overview, 4–5
 - PDU agent, 249
 - plugins, 248–249
 - prerequisite checks, 249
 - switches, monitoring and managing, 197–199
 - targets discovery, 246–247
 - Offloading. *See* Cell Offloading.
 - OLTP compression, sample code, 127
 - One-off patches, 304
 - OPatch utility, 298–299
 - OpenSM, 469–470
 - Operating systems supported, 9
 - Operational isolation
 - database consolidation, 406–407
 - schema consolidation, 407–408
 - OPlan tool, 300–302
 - OPS (Oracle Parallel Server), 41

- ORAchk tool, troubleshooting RACs, 55–56
 - Oracle Database 12c
 - AC (Application Continuity), 167–168
 - consolidating databases. *See* PDBs (Pluggable Databases).
 - FCF (Fast Connection Failover), 167–168
 - masking outages from end users, 167–168
 - Multitenant architecture. *See* PDBs (Pluggable Databases).
 - optimization. *See* Adaptive Query Optimization.
 - TAF (Transparent Application Failover), 167–168
 - Transaction Guard, 168
 - Oracle Database 12c, partitioning
 - migrating partitions, 155–157
 - partial indexes, 149–153
 - partition index maintenance, 153–155
 - Oracle Enterprise Manager (OEM). *See* OEM 12c.
 - Oracle Enterprise Manager Cloud Control 12c. *See* OEM 12c.
 - Oracle Exadata Simulation, migrating to Exadata, 276–277, 291
 - Oracle Home, upload path, 469
 - Oracle Multitenant architecture. *See* PDBs (Pluggable Databases).
 - Oracle Parallel Server (OPS), 41
 - Oracle products. *See specific products.*
 - Oracle Real Application Clusters, 41. *See also* RACs (Real Application Clusters).
 - ORATOP utility, 58
 - OS messages log files, 87
 - OS Watcher Black Box (OSWBB), 58
 - OSWatcher logs, 87. *See also* ExaWatcher utility.
 - Outages
 - AC (Application Continuity), 167–168
 - FCF (Fast Connection Failover), 167–168
 - masking from end users, 167–168
 - TAF (Transparent Application Failover), 167–168
 - Transaction Guard, 168
 - Overprovisioning, 419–420
 - Overview of Exadata. *See* Exadata overview.
- P**
- Page and block structure of Flash SSD, 418–419
 - Paradigm change, migrating to Exadata, 279–280
 - Parallel data distribution, 157–158
 - Parallel DML. *See* PEMS (parallel execution message size).
 - Parallel execution message size (PEMS), 223–224
 - PARALLEL_ parameters, 48
 - Parallel query. *See* PEMS (parallel execution message size).
 - Parallel query (PQ) execution, 158
 - Parallel recovery. *See* PEMS (parallel execution message size).
 - PARALLEL_DEGREE_POLICY parameter, 48
 - Parallelization, best practices, 48
 - Parameters. *See specific parameters.*
 - Partition Attributes Across (PAX) model, 124
 - Partitioning, best practices, 48
 - Partitions
 - migrating, 155–157
 - partial indexes, 149–153
 - partition index maintenance, 153–155
 - tiering data, 459–462
 - Password-less connections, Compute and Storage nodes, 487–491
 - Passwords, Ethernet Cisco switches, 329–330
 - patch option, patchmgr tool, 299
 - Patch Set Updates (PSUs), 303
 - patch_check_prereq parameter, 300
 - Patches
 - Data Guard Standby-First certified, 302
 - list of supported, 293
 - Patches, for Oracle
 - naming conventions, 304
 - one-off patches, 304
 - overview, 302–303
 - patching standard, 304
 - PSUs (Patch Set Updates), 303
 - SPUs (Security Path Updates), 304
 - Patching
 - Cell Nodes, 313–315
 - in a consolidated environment, 407
 - tools and utilities for. *See* Tools and utilities, for patching.
 - ZFS Storage Appliance, 311–312
 - Patching Exadata. *See also* Upgrading Exadata.
 - applying patches, 217
 - custom changes by users, 3
 - downloading patches, 311–312
 - list of supported software and patches, 293
 - QFSDP (Quarterly Full Stack Download), 297
 - standby database first, 231–232
 - Standby-First Patch Apply, 231–232
 - patchmgr tool, 299–300
 - PAX (Partition Attributes Across) model, 124
 - PCLe SSDs, 420
 - PDBs (Pluggable Databases)
 - cloning, 173–177
 - consolidation model, 169–177
 - creating, 170–177
 - database consolidation, 410–411
 - description, 170
 - dropping, 173–177
 - overview, 169
 - Pluggable Database, 170
 - RAC services, 178–183
 - root CDB (cdb\$root), 170
 - Root Container Database, 170
 - unplugging/plugging, 177–178
 - PDU agent, 249
 - PDUs (Power Distribution Units), 7, 331

- PEMS (parallel execution message size), 223–224
- Performance, Smart Flash Cache
 - vs.* Flash-based tablespace, scans, 452–453
 - index fetch, 451–452
- Performance, Smart Scans, 429–430
- Performance characteristics, assigning to grid disks, 76
- Performance gains, Smart Flash Cache, 94–95
- Performance information, validating, 467–469
- Performance Manager error counters, clearing, 482–483
- Performance simulation, migrating to Exadata, 275–277, 291
- Performance tuning. *See also* ESCF (Exadata Smart Flash Cache), performance.
 - best practices, 48
 - designing applications for, 362–364
 - Flash SSD, 417–420
 - overview, 357–358
 - SQL tuning, 372–374
 - systematic tuning, 358–359
 - third-party applications, 278
 - troubleshooting, 359–362
- Performance tuning, designing databases for
 - choosing compression levels, 372
 - disabling indexes, 370–372
 - identifying redundant, disused, or unnecessary indexes, 369–370
 - index design, new applications, 367–368
 - indexing strategy, existing applications, 368–372
 - offloading, 365–366
 - overview, 364–365
 - Smart Flash Cache and indexes, 366
 - storage indexes, 365
- Performance tuning, I/O
 - ASM, configuring, 387–388
 - database block size, changing, 388
 - overview, 386
 - Smart Flash Cache, leveraging, 387
 - write-back facility, configuring, 387
- Performance tuning RACs
 - 2-way waits, 375
 - 3-way waits, 375
 - balancing RAC database workloads, 385–386
 - balancing RAC databases, 383–386
 - cluster overhead, 376–378
 - DBRM (Database Resource Management), 385–386
 - Global Cache Service (LMS), 379
 - grants, 375
 - Instance Caging, 386
 - LMS (Global Cache Service), 379
 - prioritizing CPUs (central processing units), 386
 - prioritizing I/O, 385–386
 - resource management, 385–386
- Performance tuning RACs, Global Cache interconnect latency, 379
- latency, reducing, 378–380
- LMS latency, 381–382
- requests, 374–375
- wait events, 378
- Physical disks, listing details, 79
- Physical I/O, 359
- Physical link monitoring, 190–194
- Physical migration *vs.* logical, 281, 284
- Ping test, 479
- Planning and design
 - database consolidation, 390
 - migrating to Exadata. *See* Migrating to Exadata, planning and design.
- Planning and design, upgrading a ZFS Storage Appliance
 - Certified Platinum Configuration, 294–295
 - downloading patches, 311–312
 - overview, 294–296
 - patch release cycle, 296–297
 - time requirements, 294
- Planning and design, upgrading Exadata
 - Certified Platinum Configuration, 294–295
 - overview, 294–296
 - patch release cycle, 296–297
 - time requirements, 294
- Pluggable Databases (PDBs). *See* PDBs (Pluggable Databases).
- Plugging/unplugging PDBs, 177–178
- Plugins, 248–249
- POC (proof-of-concept) testing, 271
- Populating Smart Flash Cache data, 97–98
- Port connectivity, checking, 483
- Port counters
 - clearing, 484
 - clearing error counters, 484
 - querying and reporting nonzero ports, 484
 - validating, 484
- Port health status, querying, 192
- Port link information, reporting, 479–481
- Port state, reporting, 481–482
- POV (proof-of-value) testing, 271
- Power Distribution Units (PDUs), 7, 331
- PQ (parallel query) execution, 158
- Prioritizing
 - blocks in Smart Flash Cache, 426–427
 - CPUs (central processing units), 386
 - I/O, 385–386
- Private Cluster Interconnect
 - load balancing, 42
 - troubleshooting and tuning tool, 57
- Private Cluster Interconnect, checking, 57
- ProcWatcher script, 58
- Proof-of-concept (POC) testing, 271
- Ps component, collecting statistics about, 84

PSU patch, applying to Database Home, 323
 PSUs (Patch Set Updates), 303

Q

QFSDP (Quarterly Full Stack Download), 297
 Quarter Rack, 62–63
 quarterdeck argument, 473
 Quarterly Full Stack Download (QFSDP), 297
 QUERY HIGH compression, sample code, 127
 QUERY LOW compression, sample code, 127

R

-r option, rman2disk.ksh script, 208–209
 "RAC and Oracle Clusterware Best Practices Starter Kit," 58
 RAC Configuration Audit Tool (RACcheck), 58
 RACcheck (RAC Configuration Audit Tool), 58
 RACs (Real Application Clusters)
 effects on DBAs, 42–43
 Exadata *vs.* non-Exadata machines, 40, 42
 LMS (Lock Management Server), 41
 load balancing, 41
 managing with OEM 12c, 49
 overview, 7–8, 41–43
 with PDBs (Pluggable Databases), 178–183
 performance, 40
 performance tuning. *See* Performance tuning
 RACs.
 setting up, 43–45
 significance of, 40
 utilities and commands, 50–55
 RACs (Real Application Clusters), best practices
 anti virus software, 48
 bonded network interfaces, 46
 CPU management, 47
 effective resource management, 47
 enabling logging, 46
 Ethernet-channeled network interfaces, 46
 MAA (Maximum Availability Architecture), 45–46
 maintain current versions, 46
 memory management, 47
 optimal tuning, 48
 parallelization, 48
 partitioning, 48
 periodic health checks, 46–47
 prevent logical corruptions, 46
 preventing data block corruption, 46
 "RAC and Oracle Clusterware Best Practices Starter Kit," 58
 set up fail over ability, 46
 third-party tools and utilities, 48
 undo retention, specifying, 46
 RACs (Real Application Clusters), troubleshooting
 ADDM (Automatic Database Diagnostic Monitor), 57
 ADR (Automatic Diagnostic Repository), 56

ADRCI (Automatic Diagnostic Repository Command-line Interpreter), 56
 alerts, checking, 56
 ASH (Active Session History), 57
 AWR (Automatic Workload Repository), 57
 CHM (Cluster Health Monitor), 57
 collecting diagnostic information, 56
 health check, 55–56
 log files, checking, 56
 with OEM 12c framework, 57–58
 ORAchk tool, 55–56
 Private Cluster Interconnect, checking, 57
 storing diagnostic data, 56
 TFA (Trace File Analyzer) utility, 56
 Three As, 56–57
 tools and utilities for, 57–58
 trace logs, inspecting, 57
 tracing, enabling, 57
 Ramamurthy, Ravi, 124
 RAT_UPLOAD_CONNECT_STRING environmental variable, 469
 RAT_UPLOAD_ORACLE_HOME environmental variable, 469
 RAT_UPLOAD_PASSWORD environmental variable, 469
 RAT_UPLOAD_TABLE environmental variable, 469
 RAT_UPLOAD_USER environmental variable, 469
 RDBMS (relational database management system)
 definition, 121
 DSM (Decomposition Storage Model), 122–124
 history of, 121
 NSM (N-ary Storage Model), 122
 RDS (Reliable Datagram Sockets), 73–74
 Read-only database for reporting purposes, 232
 Real-Time Apply (RTA), 227–228
 Rebalancing data blocks after disk replacement, 313
 Recovery. *See* Backup and restore.
 Recovery Manager (RMAN). *See* RMAN (Recovery Manager).
 RECV_BUF_SIZE, setting, 220
 Relational database management system (RDBMS). *See* RDBMS (relational database management system).
 "A Relational Model of Data for Large Shared Data Banks," 121
 Reliable Datagram Sockets (RDS), 73–74
 reopen option, 228–229
 Replication. *See* PEMS (parallel execution message size).
 RESMGR: CPU quantum wait events, 47
 Resource isolation
 database consolidation, 408
 schema consolidation, 408–409
 Resource management. *See also* IORM (I/O Resource Management).
 best practices, 47
 category resource management, 404

- database consolidation, 399–401, 404
 - database resource management, 399–401
 - DBRM (Database Resource Management), 47, 385–386
 - IORM (I/O Resource Management), 401–405
 - networking, new features, 183
 - RACs, best practices, 47
 - Restart Server (RS), 67–68
 - Restarting. *See* Stopping/starting/restarting.
 - RMAN (Recovery Manager)
 - _ (underscore), hidden underscore parameters, 203
 - backup, example, 210–212
 - backup schedules, 213–214
 - CDB (container database) backups, 215–216
 - common user accounts, 215–216
 - CONVERT DATABASE, 285–286
 - CONVERT DATAFILE, 285–286
 - CONVERT TABLESPACE, 285
 - creating backups, 209–213
 - incremental backups, 202
 - overview, 202
 - PDBs (Pluggable Databases), backing up, 215–216
 - recovery, example, 210–212
 - settings, 203
 - RMAN (Recovery Manager), `rman2disk.ksh` script
 - automating backups, 204–206
 - database image copy backup, example, 207
 - incremental backup, example, 207
 - log files, 204–205
 - parameters, 208
 - .sql files, 205
 - template files, 206
 - usage examples, 206–209
 - `rman2disk.ksh` script
 - automating backups, 204–206
 - database image copy backup, example, 207
 - incremental backup, example, 207
 - log files, 204–205
 - parameters, 208
 - .sql files, 205
 - template files, 206
 - usage examples, 206–209
 - `-rollback_check_prereq` parameter, 300
 - `-rolling` option, `patchmgr` tool, 300
 - Rolling upgrades, 305–306
 - Root CDB (`cdb$root`), 170
 - Root Container Database, 170
 - Root users, 77
 - Rotational latency, 414
 - Router nodes, extracting, 478
 - Routing tables, checking, 471–472
 - RS (Restart Server), 67–68
 - RTA (Real-Time Apply), 227–228
- S**
- SATA SSDs, 420
 - Schema consolidation, definition, 390. *See also* Database consolidation.
 - Schema consolidation, isolation management
 - fault isolation, 406
 - operational isolation, 407–408
 - resource isolation, 408–409
 - security isolation, 409–410
 - Schema owner, uploading, 469
 - Schema password, uploading, 469
 - SDUs (session data units), setting, 217–218
 - Security. *See* Cell security.
 - Security isolation
 - database consolidation, 409
 - schema consolidation, 409–410
 - Security Path Updates (SPUs), 304
 - Seek time, 414, 416
 - `SEND_BUF_SIZE`, setting, 220
 - `_serial_direct_read` parameter, 102
 - Server architecture, Storage Server, 63–64
 - Server Control (`SRVCTL`) utility, 50
 - Server layer, 5–6
 - Server pools, 391–392
 - Session data units (SDUs), setting, 217–218
 - Session wait event statistics, Cell Offloading, 105–106
 - `setmpriority` command, 470–472
 - Shared storage, 6
 - `SHARED_POOL_SIZE` parameter, 224
 - Short stroking disks, 415
 - `show ip ssh verify` command, 330
 - `shownhealthy` command, 191
 - Single points of failure (SPOFs), best practices, 45–46
 - Single-block (`sreadtim`) read times, 162–163
 - Single-level cell (SLC) disks, 417–418
 - Sizing
 - CPUs, 274
 - database consolidation requirements, 393–394
 - I/O, 272–273
 - memory, 274
 - storage volumes, 275–277
 - Skounakis, Marios, 124
 - SLC (single-level cell) disks, 417–418
 - SM (subnet manager), verifying master information, 192
 - Smart Flash Cache. *See* ESCF (Exadata Smart Flash Cache).
 - Smart Flash Logging
 - controlling, 436–437
 - distribution of log file sync waits, 438–439
 - enabling, 436
 - Flash Log status, listing, 436
 - monitoring, 436–437
 - overview, 98–99, 433–435
 - testing, 437–439

- Smart Response Technology (SRT), 77
- Smart Scans
 - checking for, 103–105
 - CPU time statistics, 105–106
 - DB time statistics, 105–106
 - examples, 103–107, 109–115
 - key statistics, 104–105
 - parameters, 101–102
 - performance, 429–430
 - session wait event statistics, 105–106
 - sort reduction, 106–107
 - wait time statistics, 105–106
- `sminfo` command, 192
- Snapshots
 - architecture, 350
 - automating, 350
 - cloning, 351–352
 - copying, 351–352
 - definition, 349
 - deployment strategy, 350
 - naming conventions, 349
- Software architecture, overview
 - ASM (Automatic Storage Management), 8–9
 - Compute Nodes, 9
 - RACs (Real Application Clusters), 7–8
 - Storage Cell software, 9
- Software image versions, querying, 308–309
- Software images, displaying, 475–476
- Solaris, 9
- Solid-state disk technology. *See* Flash SSD.
- Sort reduction, Cell Offloading, 106–107
- Spindle types, migrating to Exadata, 275
- Spine switches, 64
- Spinning disks, *vs.* Flash SSD
 - actuator arms, 413
 - disk architecture, 413–414
 - limitations of disks, 413–415
 - Moore's Law, 414
 - platters, 413
 - rotational latency, 414
 - seek time, 414
 - seek times, by drive technology, 416
 - short stroking, 415
 - storage economics, by drive technology, 416
 - striping, 415
 - transfer time, 414
- SPOFs (single points of failure), best practices, 45–46
- SPUs (Security Path Updates), 304
- SQL, performance tuning, 372–374
 - `.sql` files, 205
- SQL Optimizer, monitoring storage cells, 486–487
- `sreadtim` (single-block) read times, 162–163
- SRT (Smart Response Technology), 77
- `srvctl config` command, 52–53
- `srvctl status` command, 53–54
- SRVCTL (Server Control) utility, 50
- Standby databases. *See also* Data Guard.
 - creating, 235–238
 - migrating to Exadata, 284
- Standby file management, 231
- `STANDBY_FILE_MANAGEMENT` parameter, 231
- Standby-first patching, 231
- Stopping/starting/restarting
 - CELLSRV, 66
 - MS (Management Server), 67
 - RS (Restart Server), 67–68
 - Storage Server, 67–68
- Storage architecture, Storage Server, 72–74
- Storage Cell Nodes. *See* Cell Nodes.
- Storage cells
 - alert history, querying, 70–71
 - cell storage, 61
 - command-line interface, 66
 - configuration, 66–67
 - management, 65–67
 - metrics, displaying, 69–70
 - monitoring. *See* Monitoring storage cells.
 - overview, 6, 33
 - server details, displaying, 68–69
 - statistics, querying, 71
- Storage economics, by drive technology, 416
- Storage Expansion Racks, 31–33
- Storage Indexes
 - database design, 365
 - overview, 107–109
- Storage Indexes, effects on
 - CPU time, 111–113
 - data localization, 114–115
 - DB time, 111–113
 - eliminating sorts, 110–111
 - key system statistics, 111–113
 - query execution time, 110
 - reducing recursive calls, 110–111
 - wait time, 112–114
- Storage management. *See* ASM (Automatic Storage Management).
- Storage Server. *See also* ESCF (Exadata Smart Flash Cache).
 - administration and management utility, 63
 - alert history, displaying, 69–70
 - alert notifications, configuring mail server for, 68
 - CellCLI (Control Command-Line Interface)
 - utility, 63. *See also* MS (Management Server).
 - CELLSRV (Cell Server), 65–66
 - centralizing cell management, 63
 - Compute Nodes, 61
 - configurations, 62
 - database server, 72–74
 - dcli (Distributed Command-Line Interface)
 - utility, 63
 - default system users, 77. *See also* System administration.

- deleting alert files automatically, 67
 - deleting files automatically, 67
 - disk architecture, Exadata *vs.* non-Exadata, 74–77. *See also* Cell disks; Grid disks.
 - "Exadata Database Machine and Exadata Storage Server Supported Versions," 59
 - file utilization notification, 67
 - Flash Modules, 73
 - GV\$ dynamic views, 71
 - InfiniBand storage network, 61
 - LUNs (logical unit numbers), 74–77
 - MS (Management Server), 66–67. *See also* CellCLI (Control Command-Line Interface) utility.
 - new features, 183
 - vs.* non-Exadata architecture, 73–74
 - overview, 61–63
 - preconfigured components, 61
 - RDS (Reliable Datagram Sockets), 73–74
 - RS (Restart Server), 67–68
 - server architecture, 63–64
 - service status, verifying, 67–68
 - vs.* Smart Flash Cache, 94–95
 - stopping/starting/restarting, 67–68
 - storage architecture, 72–74
 - storage server, 72–74
 - V\$ dynamic views, 71
 - X4 software and hardware capacity, 63
 - Storage Server administration and management, 63
 - Storage settings, database consolidation, 394–398
 - Storage tiering solutions, 458–462
 - Storage volume sizing, migrating to Exadata, 275–277
 - Striping disks, 415
 - Su, Qi, 124
 - Subdividing cell disks. *See* Grid disks.
 - Subnet management, 469–470
 - Subnet manager (SM), verifying master information, 192
 - Subnet priority, setting, 470
 - SunDiag utility, 83–84, 466–467
 - sundiag.sh, 83–84
 - SuperCluster M6-32, 29–31
 - SuperCluster T4-4, 25–27
 - SuperCluster T5-8, 27–29
 - Superuser privileges, 77
 - Switch health, checking, 472
 - Switch management. *See* ILOM (Integrated Lights Out Manager).
 - Switch nodes, extracting, 478
 - Switches
 - automatic archive switch, 223
 - monitoring and managing with OEM 12c, 197–199
 - Switches, Cisco
 - description, 7, 248
 - upgrading, force booting the switch, 331
 - Switches, InfiniBand
 - HTTP/HTTPS, disabling/enabling, 197
 - leaf, 64
 - monitoring and managing. *See* OEM 12c.
 - spine, 64
 - Switchover considerations, 242–243
 - Switchover tracing, 243
 - System administration
 - cell monitoring, 77
 - celladmin users, 77
 - cell-level administration, 77
 - cellmonitor users, 77
 - default system users, 77
 - disk levels, listing, 77–80
 - root users, 77
 - superuser privileges, 77
 - Systematic performance tuning, 358–359
- T**
- Table name, uploading, 469
 - Tables, database
 - checking InfiniBand routing tables, 471–472
 - creating, 100
 - table name, uploading, 469
 - TAF (Transparent Application Failover), 167–168
 - Tape media, backup and restore, 202
 - Targets discovery, 246–247
 - tcellsim.sql script, 276–277
 - TCP Nagle algorithm, enabling/disabling, 221
 - TCP.NODELAY parameter, 221
 - Testing. *See* Migrating to Exadata, migration testing.
 - textqueuelen, adjusting, 220–221
 - TFA (Trace File Analyzer) utility
 - vs.* diagcollection.pl script, 57
 - troubleshooting RACs, 56
 - tftp for Linux
 - configuring, 324–325
 - confirming available space, 325–326
 - downloading, 324–325
 - installing, 324–325
 - user name and password, 329–330
 - verifying user access, 327–331
 - Third-party applications, tuning, 278
 - Third-party tools and utilities, best practices, 48
 - Third-party *vs.* custom applications, migrating to Exadata, 278
 - Three As of troubleshooting, 56–57
 - 3-way waits, 375
 - Timeout option, 228–229
 - TLC (triple-level cache) disks, 417–418
 - Tokenization, 125
 - Tools and utilities. *See also specific commands.*
 - CellCLI (Control Command-Line Interface) utility, 63
 - Clusterware Control (CRSCTL) utility, 50

Tools and utilities (*continued*)

- dcli (Distributed Command-Line Interface) utility, 63
 - InfiniBand log file collection, 194
 - infinicheck utility, 194
 - ORATOP utility, 58
 - OSWatcher logs, 87. *See also* ExaWatcher utility.
 - OSWBB (OS Watcher Black Box), 58
 - ProcWatcher script, 58
 - RAC Configuration Audit Tool (RACcheck), 58
 - for RACs, 50–55, 58–59
 - Server Control (SRVCTL) utility, 50
 - Storage Server administration and management, 63
 - third-party, best practices, 48
 - troubleshooting networks, 190–194
 - /usr/local/bin/env_test, 194
 - /usr/local/bin/listlinkup, 194
 - /usr/local/bin/nm2version, 194
- Tools and utilities, Exadata diagnostics
- Exachk: Exadata Health Check utility, 46–47, 467–469
 - imagehistory command, 475–476
 - imageinfo command, 475–476
 - overview, 465–466
 - software images, displaying, 475–476
 - SunDiag utility, 466–467
 - troubleshooting hardware issues, 466–467
 - validating configuration and performance information, 467–469
- Tools and utilities, for patching
- available products and patches, 301
 - developing patching instructions, 300–302
 - Home software, 298–299
 - InfiniBand switch software, 299–300
 - OPatch utility, 298–299
 - OPlan tool, 300–302
 - patchmgr tool, 299–300
 - Storage software on the Cell Nodes, 299–300
- Tools and utilities, InfiniBand Network
- Diagnostics
 - a interconnected_quarterback argument, 473
 - CA (channel adapters), 477
 - channel adapter nodes, extracting, 478
 - checking InfiniBand routing tables, 471–472
 - connectivity, verifying, 479, 483
 - disable_sm command, 470
 - enable_sm command, 470
 - env_test command, 472
 - fattree argument, 473
 - getmaster command, 470–472
 - ibcheckerrors command, 482–483
 - ibchecknet command, 483
 - ibchecknode command, 483
 - ibcheckport command, 483
 - ibcheckstate command, 481–482
 - ibclearcounters command, 484
 - ibclearerrors command, 484
 - ibhosts command, 477
 - ibblinkinfo command, 479–481
 - ibping command, 479
 - ibqueryerrors command, 483
 - ibroute command, 471–472
 - ibstat command, 477
 - ibstatus command, 478–479
 - ibswitches command, 478
 - InfiniBand driver information, displaying, 477, 478–479
 - InfiniBand topology, discovering, 477–478
 - infinicheck utility, 473–475
 - LID (local identifier), 477
 - node connectivity, checking, 483
 - OpenSM, 469–470
 - overall switch health, checking, 472
 - overview, 469–470
 - Performance Manager error counters, clearing, 482–483
 - ping test, 479
 - port connectivity, checking, 483
 - port counters, clearing, 484
 - port counters, clearing error counters, 484
 - port counters, querying and reporting nonzero ports, 484
 - port counters, validating, 484
 - port link information, reporting, 479–481
 - port state, reporting, 481–482
 - quarterdeck argument, 473
 - router nodes, extracting, 478
 - sample code, 470–472
 - setsmpriority command, 470–472
 - subnet management, 469–470
 - subnet priority, setting, 470
 - switch nodes, extracting, 478
 - torus argument, 473
 - verifying InfiniBand topology, 472–475
 - verify_topology command, 472–473
 - Tools and utilities, monitoring storage cells
 - Dell software tools, 484–487
 - Dell Toad suite, 484–487
 - OEM (Oracle Enterprise Manager), 487–491
 - SQL Optimizer, 486–487
 - "Top 11gR2 Grid Infrastructure Upgrade Issues," 59
 - Top component, collecting statistics about, 84
 - Topology, InfiniBand
 - discovering, 477–478
 - verifying, 472–475
 - torus argument, 473
 - Trace File Analyzer (TFA) utility. *See* TFA (Trace File Analyzer) utility.
 - Trace logs, inspecting, 57
 - Tracing, enabling, 57
 - Transaction Guard, 168

- Transfer time, disks, 414
 - Transparent Application Failover (TAF), 167–168
 - Transport overhead, redoing. *See* Far Sync.
 - Transportable tablespaces, migrating to Exadata, 284–286
 - TRIM command, 419
 - Triple mirroring, 9, 45
 - Triple-level cache (TLC) disks, 417–418
 - Troubleshooting
 - CellCLI performance bug, 457
 - CELLSRV, 65
 - Data Guard problems, 242–243
 - hardware issues, 466–467
 - image 11.2.2.3, 318–319
 - networks, 190–194
 - Oracle Support resources, 58–59
 - performance tuning, 359–362
 - RACs (Real Application Clusters). *See* RACs (Real Application Clusters), troubleshooting.
 - Tuning. *See* Performance tuning.
 - 12c. *See* OEM 12c; Oracle Database 12c.
 - 2u custom network switch space, 7
 - 2-way waits, 375
- U**
- Underscore (_)
 - hidden underscore parameters, 203
 - underscore parameters, 101–102
 - Undo retention, specifying, 46
 - UNDO_RETENTION mode, 46
 - Unplugging/plugging PDBs, 177–178
 - upgrade option, patchmgr tool, 300
 - Upgrading a ZFS Storage Appliance
 - configuration, 333
 - overview, 333
 - recommended browser, 333
 - updating the ZFS BIOS, 335–336
 - upgrade stages, 334
 - Upgrading a ZFS Storage Appliance, planning for Certified Platinum Configuration, 294–295
 - downloading patches, 311–312
 - overview, 294–296
 - patch release cycle, 296–297
 - time requirements, 294
 - Upgrading Cisco switches, 323–331
 - Upgrading Exadata. *See also* Patching Exadata.
 - best practices, 46
 - high availability upgrades, 305–306
 - rolling upgrades, 305–306
 - Upgrading Exadata, full stack upgrades
 - 11.2.0.4 clusterware, upgrading and updating, 321–323
 - current active image, querying, 308–309
 - current release version, checking, 310–311
 - downloading patches, 311–312
 - Ethernet Cisco switches, 323–331
 - Grid Home, updating, 319–323
 - image 11.2.2.3, troubleshooting, 318–319
 - image 11.2.2.3, upgrading, 317–318
 - image 11.2.2.4.2, upgrading, 316–317
 - InfiniBand switches, updating, 319
 - KVM switches, 331
 - patching Cell Nodes, 313–315
 - PDU (Power Distribution Units), 331
 - PSU patch, applying to Database Home, 323
 - rebalancing data blocks, 313
 - recommended core requirements, 307–308
 - software image versions, querying, 308–309
 - time requirements, 307
 - updating the Compute Nodes, 315–319
 - upgrade path, 307–311
 - upgrading Cell Nodes, 312–315
 - Upgrading Exadata, planning for Certified Platinum Configuration, 294–295
 - overview, 294–296
 - patch release cycle, 296–297
 - time requirements, 294
 - Use cases, ZFS Storage Appliance, 355
 - User name, Ethernet Cisco switches, 329–330
 - /usr/local/bin/env_test utility, 194
 - /usr/local/bin/listlinkup utility, 194
 - /usr/local/bin/nm2version utility, 194
 - Utilities. *See* Tools and utilities.
- V**
- VALIDATE DATABASE command, 239–241
 - Validating
 - configuration and performance information, 467–469
 - configuration information, 467–469
 - duplicate node and port GUID, 193
 - patch testing, 290
 - performance information, 467–469
 - port counters, 484
 - V\$CELL command, 71
 - V\$CELL_REQUEST_TOTALS command, 71
 - V\$CELL_STATE command, 71
 - V\$CELL_THREAD_HISTORY command, 71
 - Vended applications. *See* Third-party applications.
 - Vendor certification, 278
 - verify command, 328
 - Verifying InfiniBand topology, 472–475
 - verify-topology command, 191
 - verify_topology command, 472–473
 - Versions, displaying for cell servers, 82–83
 - Versions of Exadata
 - "Exadata Database Machine and Exadata Storage Server Supported Versions," 59
 - overview, 10–11
 - X2-2, 11–12
 - X2-8, 13–14
 - X3-2, 14–16
 - X3-8, 16–18
 - X4-2, 18–21

Versions of Exadata (*continued*)

X4-8, 21–23

X5-2, 23–25

Vmstat component, collecting statistics about, 84

W

Wait time for reconnection, setting, 228–229

Wait time statistics

Cell Offloading, 105–106

Flash Cache Keep option, 112–114

Smart Scans, 105–106

Warehouse compression, 126

Wear leveling, 419–420

"Weaving Relations for Cache Performance," 124

Workload statistics, 161–163

Write-back cache, Smart Flash Cache

architecture, 441–442

enabling/disabling, 442

I/O bottlenecks, 440–441

overview, 439

performance, 442–443

Write-back facility, configuring, 387

X

X2-2 version, 11–12

X2-8 version, 13–14

X3-2 version, 14–16

X3-8 version, 16–18

X4-2 version, 18–21

X4-8 version, 21–23

X5-2 version, 23–25

Z

-z option, `rman2disk.ksh` script, 209

ZFS Storage Appliance

backup and restore, 201–202

best-practice settings, 352–355

BUI (Browser User Interface), 342–343

clones, 351–352

configuring NFS shares, 348–349

data compression options, 354

data deduplication, 354

database consolidation, 341

dNFS, 348–349

downloading patches for, 311–312

eliminating the DBFS, 341

enabling NFS services, 345–348

ILM (Information Lifecycle Management), 341

industry use cases, 355

latency, 353

line of products, 338–340

mounting NFS shares, 348–349

NFS shares, creating, 343–345

patching, 311–312

recovering space and resources from the DBFS,
341

snapshots, 349–352

storage capacity, 340–342

storage profiles, choosing, 340

ZFS Storage Appliance, upgrading

Certified Platinum Configuration, 294–295

configuration, 333

downloading patches, 311–312

overview, 294–296, 333

patch release cycle, 296–297

planning for, 294–297

recommended browser, 333

time requirements, 294

updating the ZFS BIOS, 335–336

upgrade stages, 334

ZFS Storage Appliance simulator, 355–356

ZFSSA (ZFS Storage Appliance). *See* ZFS Storage
Appliance.

ZLIB compression algorithm, 128–129