



Integrated Cloud Applications & Platform Services

Exadata Database Machine: 12c Administration Workshop

Student Guide - Volume I

D92887GC20

Edition 2.0 | November 2016 | D97405

Learn more from Oracle University at education.oracle.com



Oracle University and Error : You are not a valid Partner use only

ORACLE®

Author

Peter Fusek

Technical Contributors and Reviewers

Akshay Shah

Alex Tsukerman

Amit Ganesh

Andrew Babb

Aslam Edah-Tally

Barb Lundhild

Bharat Baddepudi

Bill Hodak

Boris Erlikhman

Branislav Valny

Bruce Kyro

Bruno Dagostini

Bundit Jianpinitnan

Caroline Johnston

Christian Craft

Dan Norris

Darryl Presley

Dave Winter

David Hernandez Mendoza

David Hitchcock

Deba Chatterjee

Douglas Utzig

Ed Gilowski

Eric Siglin

Francois Marcoux

Georg Schmidt

Gerlinde Frenzen

Harald van Breederode

Ira Singer

James He

James Womack

Jean-Francois Verrier

Jerry Lee

Jia Shi

Jignesh Patel

Jim Hall

Jim Spiller

Jim Viscusi

Joe Fu

Joel Goodman

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Editors

Arijit Ghosh

Aishwarya Menon

Graphic Designer

Seema Bopaiah

Publishers

Sujatha Nagendra

Sumesh Koshy

Asief Baig

Juan Loaiza	Richard Exley	Editors
Kam Shergill	Robert Carlin	Arijit Ghosh
Kevin Jernigan	Robert Pastijn	Aishwarya Menon
Kodi Umamageswaran	Roger Hansen	
Krishnanjani Chitta	Sabyasachi Banerjee	Graphic Designer
Lachlan Williams	Sean Kim	Seema Bopaiah
Larry Justice	Selcuk Aya	
Lawrence To	Scott Gossett	Publishers
Louis Nagode	Sriram Palapudi	Sujatha Nagendra
Mahesh Subramaniam	Steven Lemme	Sumesh Koshy
Maria Billings	Sue Lee	Asief Baig
Maria Colgan	Sugam Pandey	
Mark Fuller	Sumeet Lahorani	
Mark Scardina	Sundararaman Sridharan	
Mark Van de Wiel	Suresh Gogune	
Marshall Presser	Tim Shetler	
Martin Jensen	Trevor Bowen	
Michael Cebulla	Umesh Panchaksharaiah	
Michael Nowak	Uwe Hesse	
Naoki Kato	Varun Malhotra	
Nikolaus Dilger	Vern Wagman	
Nilesh Choudhury	Vijay Sridharan	
Ravindra Dani	Vikram Kapoor	
Raymond Dutcher	Vimala Jacob	

Contents

1 Introduction

- Course Objectives 1-2
- Audience and Prerequisites 1-3
- Course Contents 1-4
- Additional Resources 1-5
- Practice 1 Overview: Introducing the Laboratory Environment 1-6

2 Exadata Database Machine: Overview

- Objectives 2-2
- Introducing Exadata Database Machine 2-3
- Why Exadata Database Machine? 2-4
- Introducing Exadata Storage Server 2-6
- Exadata Storage Server Architecture: Overview 2-7
- Exadata Storage Server Features: Overview 2-8
- Exadata X6-2 High Capacity Storage Server Hardware: Overview 2-10
- Exadata X6-2 Extreme Flash Storage Server Hardware: Overview 2-11
- Exadata Storage Server X6-2 Configuration Options 2-12
- Exadata Database Machine X6-2 Database Server Hardware: Overview 2-13
- Exadata Database Machine X6-2 Full Rack 2-14
- Elastic Configuration 2-15
- Elastic Configuration Examples 2-16
- Start Small and Grow 2-17
- Exadata Database Machine X6-8 2-18
- Exadata Database Machine X6-8 Database Server Hardware: Overview 2-19
- Exadata Database Machine X6-8 Use Cases 2-20
- Oracle Exadata Virtual Machines 2-21
- InfiniBand Network: Overview 2-23
- Exadata Storage Expansion Racks 2-24
- Exadata Database Machine Support: Overview 2-25
- Oracle Platinum Services: Enhanced Support at No Additional Cost 2-26
- Quiz 2-27
- Summary 2-29

3 Exadata Database Machine Architecture
Objectives 3-2
Exadata Architecture: Overview 3-3
Exadata Network Architecture 3-5
InfiniBand Network Architecture 3-7
Active Bonding InfiniBand Connectivity 3-8
Leaf Switch Topology 3-9
Spine and Leaf Topology 3-10
Scale Performance and Capacity Beyond a Single Rack 3-11
Typical Scaling Scenarios 3-12
Scaling Out to Eight Racks 3-14
Scaling Out Between 9 and 18 Racks 3-15
Scaling Out Between 19 and 36 Racks 3-16
Interconnecting Quarter Racks and Eighth Racks 3-17
InfiniBand Network External Connectivity 3-19
Exadata Software Architecture: Overview 3-20
Support for Mixed Database Versions 3-23
High Capacity Storage Server: Disk Storage Entities and Relationships 3-24
High Capacity Storage Server: Flash Storage Entities and Relationships 3-26
Extreme Flash Storage Server: Flash Storage Entities and Relationships 3-27
Disk Group Configuration 3-28
Quiz 3-29
Summary 3-33
Practice 3 Overview: Introducing Exadata Cell Architecture 3-34
4 Key Capabilities of Exadata Database Machine
Objectives 4-2
Classic Database I/O and SQL Processing Model 4-3
Exadata Smart Scan Model 4-4
Exadata Smart Storage Capabilities 4-6
Exadata Smart Scan Scale-Out: Example 4-9
Hybrid Columnar Compression: Overview 4-12
Hybrid Columnar Compression: Data Organization 4-13
Exadata Smart Flash Cache: Overview 4-14
Exadata Smart Flash Cache Intelligent Caching: Overview 4-15
Exadata Smart Flash Cache Intelligent Caching Details 4-16
Using Exadata Smart Flash Cache: Write-Through Cache 4-18
Using Exadata Smart Flash Cache: Write-Back Cache 4-20
Columnar Flash Caching 4-22
Exadata Smart Flash Log: Overview 4-23
Exadata Storage Index: Overview 4-24

Storage Index with Partitions: Example 4-26
Performance Optimization for SQL Queries with Minimum or Maximum Functions 4-27
Exafusion Direct-to-Wire Protocol 4-28
Exadata Network Resource Management 4-29
Snapshot Databases for Test and Development 4-30
I/O Resource Management: Overview 4-31
Cell-to-Cell Data Transfer 4-32
Multiplied Benefits 4-33
Exadata Benefits for Data Warehousing and Analytics 4-34
Exadata Benefits for OLTP 4-36
Quiz 4-37
Summary 4-38
Practice 4 Overview: Introducing Exadata Features 4-39

5 Exadata Database Machine Initial Configuration

Objectives 5-2
Exadata Implementation: Overview 5-3
Key Documentation 5-5
Exadata Site Preparation 5-6
Exadata Configuration Tool: Overview 5-7
Exadata Configuration Tool: Customer Details 5-8
Exadata Configuration Tool: Hardware Selection 5-10
Exadata Configuration Tool: Rack Details 5-11
Exadata Configuration Tool: Define Customer Networks 5-12
VLAN Support 5-13
Exadata Configuration Tool: Administration Network 5-14
Administration Network IP Address Allocation: Example 5-15
Exadata Configuration Tool: Client Ethernet Network 5-16
Client Ethernet Network IP Address Allocation: Example 5-17
Exadata Configuration Tool: InfiniBand Network 5-18
InfiniBand Network IP Address Allocation: Example 5-19
Exadata Configuration Tool: Backup Network 5-20
Exadata Configuration Tool: Identify Compute Node OS 5-21
Configuring Virtualized Compute Nodes 5-22
Exadata Configuration Tool: Review and Edit 5-24
Exadata Configuration Tool: Define Clusters 5-25
Multiple Cluster Configuration Options 5-26
Exadata Configuration Tool: Cluster Configuration - Part 1 5-27
Exadata Configuration Tool: Cluster Configuration - Part 2 5-28
Exadata Storage Configuration Example 5-30

Choosing the Right Disk Group Redundancy Setting	5-32
Quorum Disks on Database Servers	5-34
Exadata Configuration Tool: Review and Edit	5-36
Exadata Configuration Tool: Alerting	5-37
Exadata Configuration Tool: Platinum Configuration	5-38
Exadata Configuration Tool: Oracle Configuration Manager	5-39
Exadata Configuration Tool: Auto Service Request	5-40
Exadata Configuration Tool: Grid Control Agent	5-41
Exadata Configuration Tool: Comments	5-42
Exadata Configuration Tool: Generate	5-43
Exadata Configuration Tool: Finish	5-44
Exadata Hardware Installation: Overview	5-45
Configuring Exadata: Overview	5-46
Loading the Configuration Information and Installing the Software	5-47
Running the Exadata Deployment Tool	5-48
Result After Installation and Configuration	5-49
Supported Additional Configuration Activities	5-50
Hardware Re-Racking	5-51
Unsupported Configuration Activities	5-53
Quiz	5-54
Summary	5-56
Practice 5 Overview: Using the Exadata Configuration Tool	5-57

6 Exadata Storage Server Configuration

Objectives	6-2
Exadata Storage Server Administration: Overview	6-3
Exadata Storage Server Administrative User Accounts	6-4
Exadata Storage Server Users, Roles, and Privileges	6-5
Exadata Storage Server Users, Roles, and Privileges: Examples	6-6
Running CellCLI Commands from Database Servers	6-7
ExaCLI: Examples	6-9
Executing Commands Across Multiple Servers Using dcli	6-10
dcli: Examples	6-11
Executing Commands Across Multiple Servers Using exadcli	6-12
exadcli: Examples	6-13
Testing Storage Server Performance by Using CALIBRATE	6-14
CALIBRATE: Example	6-15
Configuring the Exadata Cell Server Software	6-16
Starting and Stopping Exadata Cell Server Software	6-17
Configuring Cell Disks	6-18
Configuring Grid Disks	6-19

Sparse Grid Disks	6-20
Configuring Hosts to Access Exadata Cells	6-21
Configuring ASM and Database Instances to Access Exadata Cells	6-22
Configuring ASM Disk Groups by Using Exadata Storage	6-23
Specifying Content Type for a Disk Group	6-24
Reconfiguring Exadata Storage	6-26
Optional Configuration Tasks	6-28
Exadata Storage Security: Overview	6-29
Exadata Storage Security Implementation	6-30
Quiz	6-32
Summary	6-35
Practice 6 Overview: Configuring Exadata	6-36

7 I/O Resource Management

Objectives	7-2
I/O Resource Management: Overview	7-3
I/O Resource Management Concepts	7-5
I/O Resource Management Plans	7-6
I/O Resource Management Plans: Example	7-7
IORM Architecture	7-10
Getting Started with IORM	7-11
Setting the IORM Objective	7-12
Enabling Intradatabase Resource Management	7-13
Intradatabase Plan: Example	7-14
Enabling IORM for Multiple Databases	7-15
Interdatabase Plan: Example	7-16
Category Plan: Example	7-17
Complete Example	7-18
Using Share-Based Allocation in the Interdatabase Plan	7-21
Setting Database I/O Utilization Limits	7-22
I/O Resource Management Profiles	7-23
Interdatabase Plans and Database Roles	7-24
Using Database I/O Metrics	7-25
I/O Resource Management for Flash	7-27
Flash Cache and Flash Log Resource Control	7-28
Flash Cache Space Resource Management	7-29
Using Exadata I/O Resource Management with Oracle Database 12c	7-30
Quiz	7-31
Summary	7-35
Additional Resources	7-36

8 Recommendations for Optimizing Database Performance
Objectives 8-2
Optimizing Performance 8-3
Flash Memory Usage 8-4
Write Back Flash Cache on Extreme Flash Cells 8-5
Influencing Caching Priorities 8-6
Choosing the Flash Cache Mode for Non-Extreme Flash Cells 8-7
Setting the Flash Cache Mode 8-8
Table Compression Usage 8-9
Index Usage 8-11
ASM Allocation Unit Size 8-12
Extent Size 8-13
Exadata Specific System Statistics 8-14
Exadata I/O Latency Capping 8-15
Setting the Exadata Cell I/O Timeout Threshold 8-16
Quiz 8-17
Summary 8-19
Practice 8: Overview Optimizing Database Performance with Exadata 8-20
9 Using Smart Scan
Objectives 9-2
Exadata Smart Scan: Overview 9-3
Smart Scan Requirements 9-4
Situations Preventing Smart Scan 9-6
Monitoring Smart Scan in SQL Execution Plans 9-7
Smart Scan Execution Plan: Example 9-8
Example of a Situation Preventing Smart Scan 9-10
Smart Scan Join Processing with Bloom Filters 9-11
Smart Scan Join Filtering: Example 9-12
Other Situations Affecting Smart Scan 9-13
Exadata Storage Server Statistics: Overview 9-15
Exadata Storage Server Wait Events: Overview 9-16
Smart Scan Statistics: Example 9-17
Smart Scan Wait Events: Example 9-18
Concurrent Transaction: Example 9-19
Extreme Concurrent Transaction: Example 9-20
Migrated Rows: Example 9-21
I/O Sent Directly to Database Server to Balance CPU Usage: Example 9-22
Column Filtering: Example 9-23
Summary 9-24

Quiz 9-25
Practice 9 Overview: Using Smart Scan 9-27

10 Consolidation Options and Recommendations

Objectives 10-2
Consolidation: Overview 10-3
Different Consolidation Types 10-4
Core Principles for Database Consolidation 10-6
Recommended Consolidation Approach 10-7
Recommended Storage Configuration for Consolidation 10-8
Alternative Storage Configurations 10-10
Benefits and Limitations of Partitioned Storage Configurations 10-12
Cluster Configuration Options 10-13
Operating System Parameter Recommendations 10-14
Database Memory Recommendations 10-16
CPU Management Recommendations 10-17
Process Management Recommendations 10-19
Other Recommendations 10-21
Isolating Management Roles 10-22
Schema Consolidation Recommendations 10-24
Consolidation Using Virtual Machines 10-25
Consolidation Using Oracle Multitenant Architecture 10-26
General Maintenance Considerations 10-28
Quiz 10-29
Summary 10-31
Additional Resources 10-32

11 Migrating Databases to Exadata Database Machine

Objectives 11-2
Migration Best Practices: Overview 11-3
Performing Capacity Planning 11-4
Exadata Migration Considerations 11-5
Choosing the Right Migration Path 11-6
Logical Migration Approaches 11-7
Physical Migration Approaches 11-9
Reducing Down Time for Migration by Using Transportable Tablespaces 11-12
Other Approaches 11-13
Post-Migration: Best Practices 11-14
Quiz 11-15
Summary 11-17

Additional Resources 11-18
Practice 11 Overview: Migrating to Exadata Using Transportable Tablespaces 11-19

12 Bulk Data Loading

Objectives 12-2
Bulk Data Loading Architectures for Exadata 12-3
Staging Data Files Using DBFS 12-4
Staging Data Files Using ACFS 12-5
Staging Data Files Using External File Systems 12-6
Comparison of Staging Area Options 12-7
Bulk Data Loading Using Oracle DBFS: Overview 12-9
Preparing the Data Files 12-10
Configuring a DBFS Staging Area 12-11
Configuring the Target Database 12-16
Loading the Target Database 12-17
Quiz 12-19
Summary 12-21
Additional Resources 12-22
Practice 12 Overview: Bulk Data Loading Using Oracle DBFS 12-23

13 Exadata Database Machine Platform Monitoring: Introduction

Objectives 13-2
Monitoring Technologies and Standards 13-3
Simple Network Management Protocol (SNMP) 13-4
Intelligent Platform Management Interface (IPMI) 13-5
Integrated Lights Out Manager (ILOM) 13-6
Exadata Storage Server: Metrics, Thresholds, and Alerts 13-7
Automatic Diagnostic Repository (ADR) 13-8
Enterprise Manager Cloud Control 13-9
Enterprise Manager Database Control and Enterprise Manager Database Express 13-10
Quiz 13-11
Summary 13-12

14 Configuring Enterprise Manager Cloud Control to Monitor Exadata Database Machine

Objectives 14-2
Enterprise Manager Cloud Control Architecture: Overview 14-3
Enterprise Manager Cloud Control: Supported Exadata Configurations 14-4
Cloud Control Monitoring Architecture for Exadata 14-5

Configuring Cloud Control to Monitor Exadata	14-6
Pre-Discovery Configuration and Verification	14-7
Deploying the Oracle Management Agent	14-9
Discovering Exadata	14-10
Discovering Additional Targets	14-11
Post-Discovery Configuration and Verification	14-12
Configuring an Exadata Dashboard	14-13
Quiz	14-14
Summary	14-17
Additional Resources	14-18
Practice 14 Overview: Exadata Monitoring Configuration	14-19

15 Monitoring Exadata Storage Servers

Objectives	15-2
Lesson Overview	15-3
Exadata Storage Server Metrics and Alerts Architecture	15-4
Monitoring Exadata Storage Server with Metrics	15-6
Monitoring Exadata Cell Metrics: Examples	15-8
Monitoring Exadata Storage Server with Alerts	15-9
Monitoring Cell Alerts and Creating Thresholds: Examples	15-11
Isolating Faults with Exadata Storage Server Quarantine	15-13
Monitoring Exadata Storage Server with Active Requests	15-15
Automatic Hard Disk Scrubbing and Repair	15-16
Adaptive Hard Disk Scrubbing	15-17
Cell Alert Summary	15-18
Cell Diagnostic Packages	15-19
Monitoring Exadata Storage Server with Enterprise Manager: Overview	15-20
Monitoring Hardware Failure and Sensor State	15-22
Monitoring Exadata Storage Server Availability	15-23
Checking for Undelivered Alerts	15-24
Checking for Disk I/O Errors	15-25
Checking for Network Errors	15-26
Monitoring File System Free Space	15-27
Comparing Metrics Across Multiple Storage Servers	15-28
Monitoring Metrics in a Storage Server	15-29
Third-Party Monitoring Tools	15-30
Quiz	15-31
Summary	15-33
Practice 15 Overview: Monitoring Exadata Storage Server	15-34

16 Monitoring Exadata Database Machine Database Servers

- Objectives 16-2
- Monitoring Database Servers: Overview 16-3
- Monitoring Hardware 16-4
- Monitoring the Operating System 16-5
- Monitoring Oracle Grid Infrastructure and Database 16-6
- Monitoring Oracle Management Agent 16-7
- Database Monitoring with Enterprise Manager Cloud Control 16-8
- Monitoring Database Servers with MS and DBMCLI: Overview 16-9
- Running DBMCLI 16-10
- Starting and Stopping Management Services on Exadata Database Servers 16-11
- Configuring Management Services on Exadata Database Servers 16-12
- Monitoring Database Server Metrics: Examples 16-13
- Quiz 16-14
- Summary 16-15
- Practice 16 Overview: Oracle Database Monitoring 16-16

17 Monitoring the InfiniBand Network

- Objectives 17-2
- InfiniBand Network Monitoring: Overview 17-3
- InfiniBand Network Monitoring with Enterprise Manager Cloud Control 17-4
- Monitoring the InfiniBand Switches 17-5
- Monitoring the InfiniBand Switch Ports 17-6
- Monitoring the InfiniBand Ports on Exadata Servers 17-7
- Monitoring the InfiniBand Fabric: Subnet Manager Master Location 17-8
- Monitoring the InfiniBand Fabric: Network Topology and Link Status 17-9
- Quiz 17-10
- Summary 17-11
- Practice 17 Overview: InfiniBand Monitoring 17-12

18 Monitoring Other Exadata Database Machine Components

- Objectives 18-2
- Monitoring the Cisco Ethernet Switch 18-3
- Monitoring the Power Distribution Units 18-4
- Monitoring the KVM Switch 18-5
- Quiz 18-6
- Summary 18-7

19 Other Useful Exadata Monitoring Tools

- Objectives 19-2
- Exachk: Overview 19-3

Running Exachk	19-4
Exachk Output	19-6
Exachk Daemon	19-7
ExaWatcher: Overview	19-8
TFA Collector: Overview	19-9
Running TFA Collector on Exadata	19-10
Using ADRCI on Exadata Storage Servers	19-11
imageinfo: Overview	19-12
imagehistory: Overview	19-13
Integrated Lights Out Manager (ILOM): Overview	19-14
Using ILOM	19-15
Accessing the Browser Interface	19-16
Powering the Device On or Off	19-17
Locating the Device	19-18
Viewing Hardware Status and Specifications	19-19
Monitoring Power Consumption	19-20
Accessing the Command-Line Interface (CLI)	19-21
CLI: Examples	19-22
Quiz	19-23
Summary	19-26
Additional Resources	19-27

20 Backup and Recovery

Objectives	20-2
Backup and Recovery: Overview	20-3
Using RMAN with Exadata	20-4
General Recommendations for RMAN	20-5
Exadata Disk-Based Backup Strategy	20-7
Disk-Based Backup Recommendations	20-8
Disk-Based Backup on Non-Exadata Storage	20-10
Tape-Based Backup Strategy	20-12
Tape-Based Backup Architecture	20-13
Tape-Based Backup Recommendations	20-14
Connecting the Media Server by Using Ethernet	20-16
Tape-Based Backup Recommendations	20-17
Hybrid Backup Strategy	20-18
Restore and Recovery Recommendations	20-19
Backup and Recovery of Database Machine Software	20-20
Quiz	20-21
Summary	20-22

Additional Resources	20-23
Practice 20 Overview: Using RMAN Optimizations for Exadata	20-24

21 Exadata Database Machine Maintenance Tasks

Objectives	21-2
Exadata Maintenance: Overview	21-3
Powering Exadata Off and On	21-4
Safely Shutting Down a Single Exadata Storage Server	21-5
Replacing a Damaged Physical Disk	21-6
Safe Disk Removal	21-8
Replacing a Damaged Flash Card	21-9
Moving All Disks from One Cell to Another	21-10
Using the Exadata Cell Software Rescue Procedure	21-12
Quiz	21-14
Summary	21-17

22 Patching Exadata Database Machine

Objectives	22-2
Patching and Updating: Overview	22-3
Patching and Updating: Key Information Sources	22-4
Maintaining Exadata Storage Server Software	22-5
Using patchmgr to Orchestrate Storage Server Patching	22-6
Maintaining Database Server Software	22-7
Assisted Patching Using OPlan	22-8
Assisted Patching Using the DB Node Update Utility	22-9
Using patchmgr to Orchestrate Database Server Patching	22-10
Maintaining Other Software	22-11
Recommended Patching Process	22-12
Test System Recommendations	22-14
Quiz	22-15
Summary	22-16
Additional Resources	22-17

23 Exadata Database Machine Automated Support Ecosystem

Objectives	23-2
Auto Service Request: Overview	23-3
ASR Process	23-4
ASR Requirements	23-5
Configuring the ASR Manager	23-6
Configuring Exadata for ASR	23-7
Activating ASR Assets	23-8

Verifying the ASR Configuration	23-9
Oracle Configuration Manager: Overview	23-10
Configuring Oracle Configuration Manager	23-11
Quiz	23-12
Summary	23-15
Additional Resources	23-16

24 Oracle Database Exadata Cloud Service Overview

Objectives	24-2
Introducing Exadata Cloud Service	24-3
Service Configuration Options	24-5
Service Connection Options	24-7
Service Architecture	24-9
Service Availability	24-10
Service Scalability	24-11
Service Access and Security	24-12
Data Security	24-14
Management Responsibilities	24-15
Storage Configuration	24-17
Storage Management Details	24-19
Simple Web-Based Provisioning	24-20
Simple Web-Based Management	24-21
REST APIs	24-22
Backup and Recovery	24-23
Patching Exadata Cloud Service	24-25
Migrating to Exadata Cloud Service	24-26
Summary	24-27

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

1

Introduction

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Course Objectives

After completing this course, you should be able to:

- Describe the key capabilities of Exadata Database Machine
- Identify the benefits of using Exadata Database Machine for different application classes
- Describe the architecture of Exadata Database Machine and its integration with Oracle Database, Clusterware, and ASM
- Complete the initial configuration of Exadata Database Machine
- Describe various recommended approaches for migrating to Exadata Database Machine
- Configure I/O Resource Management
- Monitor Exadata Database Machine health and optimize performance



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Audience and Prerequisites

- This course is primarily designed for administrators who will configure and administer Oracle Exadata Database Machine.
- Prior knowledge and understanding of the following is assumed:
 - Oracle Database including RAC and ASM
 - General operating system, network, storage, and system administration concepts
- Recommended prior training:
 - Oracle Database Administration
 - Oracle RAC, Clusterware, and ASM Administration
 - Linux System Administration



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This course is primarily designed for administrators who will configure and administer Oracle Exadata Database Machine.

Please be mindful of the prerequisites because this course does not teach all aspects of the technologies used inside Exadata Database Machine. Rather it focuses on topics that are specific to Exadata Database Machine.

Prior knowledge and understanding of Oracle Database 11g Release 2 or later, including Automatic Storage Management (ASM) and Real Application Clusters (RAC), is assumed. In addition, a working knowledge of Unix/Linux is assumed along with an understand of general networking, storage, and system administration concepts.

For students who do not meet these prerequisites, prior training in Oracle Database administration, along with administration of Oracle RAC, clusterware and ASM, is recommended. For Oracle Database 12c, the following courses are recommended:

- *Oracle Database 12c: Administration Workshop*
- *Oracle Database 12c: Clusterware Administration*
- *Oracle Database 12c: ASM Administration*
- *Oracle Database 12c: RAC Administration*

In addition, prior training in Linux system administration fundamentals is recommended.

Course Contents

1. Introduction (this lesson)
2. Exadata Database Machine: Overview
3. Exadata Database Machine Architecture
4. Key Capabilities of Exadata Database Machine
5. Exadata Database Machine Initial Configuration
6. Exadata Storage Server Configuration
7. I/O Resource Management
8. Recommendations for Optimizing Database Performance
9. Using Smart Scan
10. Consolidation Options and Recommendations
11. Migrating Databases to Exadata Database Machine
12. Bulk Data Loading
13. Exadata Database Machine Platform Monitoring: Introduction
14. Configuring Enterprise Manager Cloud Control to Monitor Exadata Database Machine
15. Monitoring Exadata Storage Servers
16. Monitoring Exadata Database Machine Database Servers
17. Monitoring the InfiniBand Network
18. Monitoring Other Exadata Database Machine Components
19. Other Useful Monitoring Tools
20. Backup and Recovery
21. Exadata Database Machine Maintenance Tasks
22. Patching Exadata Database Machine
23. Exadata Database Machine Automated Support Ecosystem



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows the order of lessons in this course.

Additional Resources

- Oracle.com Exadata home page
 - <http://www.oracle.com/exadata>
- Oracle Technology Network (OTN) Exadata home page
 - <http://otn.oracle.com/server-storage/engineered-systems/exadata/index.html>
- Exadata documentation
 - <http://docs.oracle.com/en/engineered-systems/>
- OTN Exadata discussion forum
 - <http://forums.oracle.com/forums/forum.jspa?forumID=829>
- Oracle Learning Library
 - <http://www.oracle.com/oll>
 - Search for demonstrations with *Exadata* or *Database Machine* in the title.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 1 Overview: Introducing the Laboratory Environment

In this practice, you will be introduced to the laboratory environment used to support all the practices during this course.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

2

Exadata Database Machine: Overview

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Provide an overview-level description of Exadata Storage Server and the different Exadata Database Machine configurations
- Outline the key capacity and performance specifications for Exadata Database Machine



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Introducing Exadata Database Machine

- Fully integrated platform for Oracle Database
- Based on Exadata Storage Server storage technology
- High-performance and high-availability for all Oracle Database workloads
- Balanced hardware configurations
- Scale-out architecture
- Well suited for cloud and database consolidation platform
- Simple and fast to implement



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Database Machine is a fully integrated Oracle Database platform based on Exadata Storage Server storage technology. Exadata provides a high-performance, highly available and highly scalable solution for all database workloads, ranging from scan-intensive analytics to highly concurrent OLTP applications.

Special attention has been paid to ensure that Exadata is a well-balanced platform. Throughout the hardware architecture of Exadata, components and technologies have been specially matched to eliminate bottlenecks while maintaining good hardware utilization.

Using the unique clustering and workload management capabilities of Oracle Database, along with complimentary Exadata-specific features, Exadata is well-suited for consolidating multiple databases onto a single Exadata system and underpinning private and public cloud implementations. Delivered as a complete package of software, servers, and storage, Exadata is simple and fast to implement.

Note: Although Exadata is a fully integrated platform solution that comprises specific hardware and software components, Oracle offers Exadata hardware and software components as a series of separately purchasable items. Customers can choose from the different hardware configurations that are available. Appropriate licensing of Oracle Database and Exadata cell software is also required. In addition, Exadata is highly complementary with clustering and parallel operations, so Oracle Real Application Clusters and Oracle Partitioning are highly recommended software options for Exadata.

Why Exadata Database Machine?

Exadata is designed to address common issues:

- Issues for Data Warehousing and Analytics:
 - Supporting large, complex queries
 - Managing massive databases
- OLTP issues:
 - Supporting large user populations and transaction volumes
 - Delivering quick and consistent response times
- Consolidation issues:
 - Efficiently supporting mixed workloads
 - Prioritizing workloads
- Configuration Issues:
 - Creating a balanced configuration without bottlenecks
 - Building and maintaining a robust system that works



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Database Machine is an engineered system that is designed to address common issues faced by most database users, especially as databases increase in size and complexity. During this course you will learn about Exadata technologies and practices that address the following issues and requirements:

Issues for Data Warehousing and Analytics:

- Supporting large, complex queries:
 - Getting enough I/O throughput to support massive scans
 - Driving the I/O throughput over the storage network
 - Avoiding unproductive I/O
 - Parallel processing
- Managing massive databases:
 - Easily and effectively managing storage space
 - Utilizing resources effectively while controlling runaway queries
 - Efficiently compressing data

OLTP issues:

- Supporting large user populations and transaction volumes:
 - Getting enough I/Os per second
 - Caching frequently accessed data
- Delivering quick and consistent response times:
 - Minimizing I/O latency
 - Efficient commit processing

Consolidation issues:

- Efficiently supporting mixed workloads:
 - Supporting different workloads on the same system
 - Isolating workloads to avoid conflict
- Prioritizing workloads:
 - Managing resources based on established priorities
 - Dynamically adjusting resource allocations based on current system configuration and workload observations

Configuration Issues:

- Creating a balanced configuration without bottlenecks:
 - Hardware components matched with each other across the system
 - Hardware specifications matched to software capabilities
- Building and maintaining a robust system that works:
 - Hardware, firmware, and software compatibility
 - Configuration best practices aiding consistency and supportability
 - Intelligent monitoring tools

Introducing Exadata Storage Server

Exadata Storage Server



- High-performance storage for Oracle Database
- 64 bit Intel-based server
- Preinstalled software:
 - Exadata Storage Server Software
 - Oracle Linux x86_64
 - Drivers and Utilities
- Only available in conjunction with Oracle Engineered Systems

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

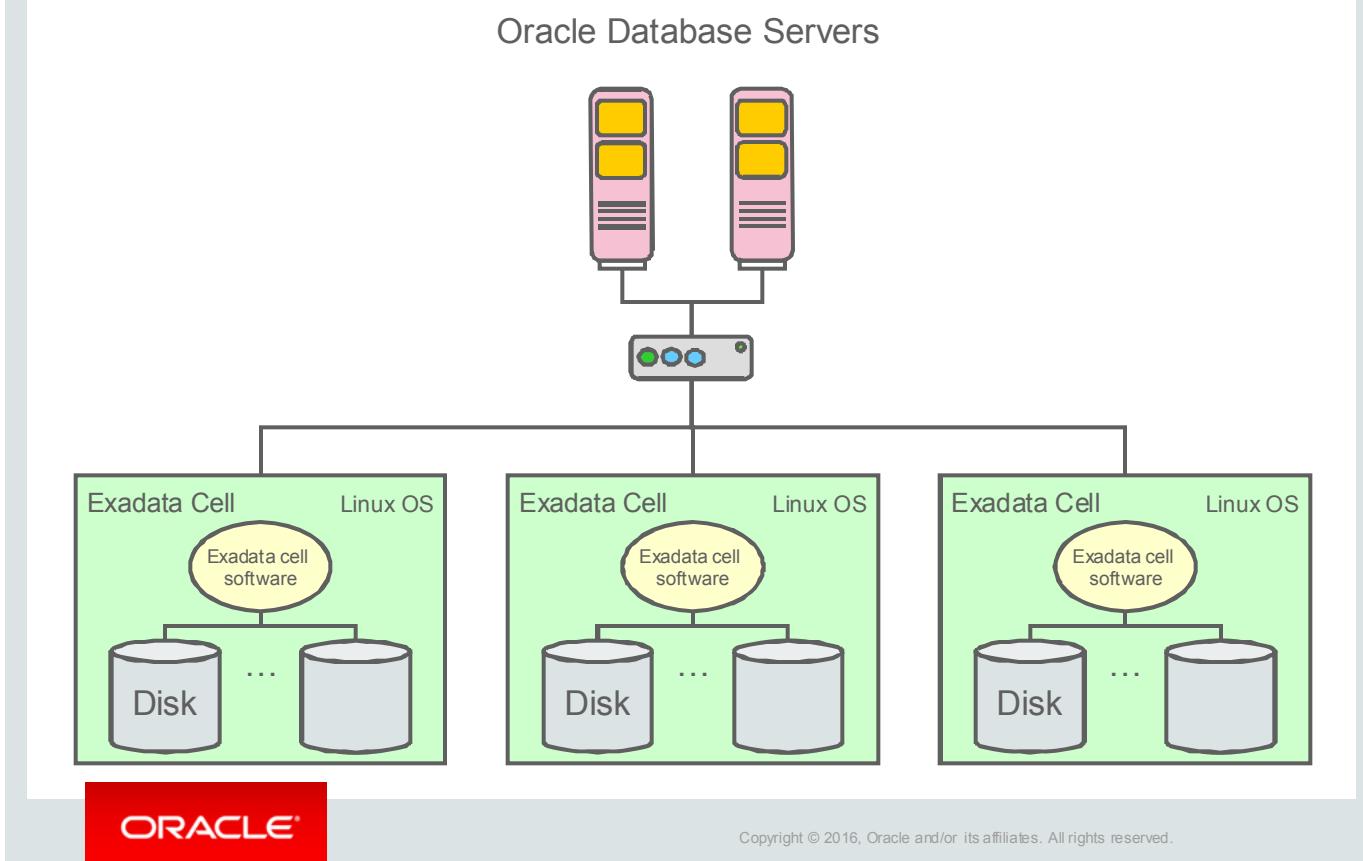
Exadata Storage Server is highly optimized storage for use with Oracle Database. It delivers outstanding I/O and SQL processing performance for all Oracle Database applications.

Each Exadata Storage Server is based on a 64 bit Intel-based server. Oracle also provides the storage server software to impart database intelligence to the storage, and tight integration with Oracle Database and its features. Each Exadata Storage Server is shipped with all the hardware and software components preinstalled including the Exadata Storage Server software, Oracle Linux x86_64 operating system and InfiniBand protocol drivers.

Exadata Storage Server is only available for use in conjunction with specific Oracle Engineered Systems such as Exadata Database Machine or Oracle Super Cluster. Individual Exadata Storage Servers can be purchased; however, they must be connected to an Engineered System that is designed to use Exadata Storage Servers. Custom configurations using Exadata Storage Servers are not supported.

Oracle Zero Data Loss Recovery Appliance also uses a storage server that is based on Exadata Storage Server technology.

Exadata Storage Server Architecture: Overview



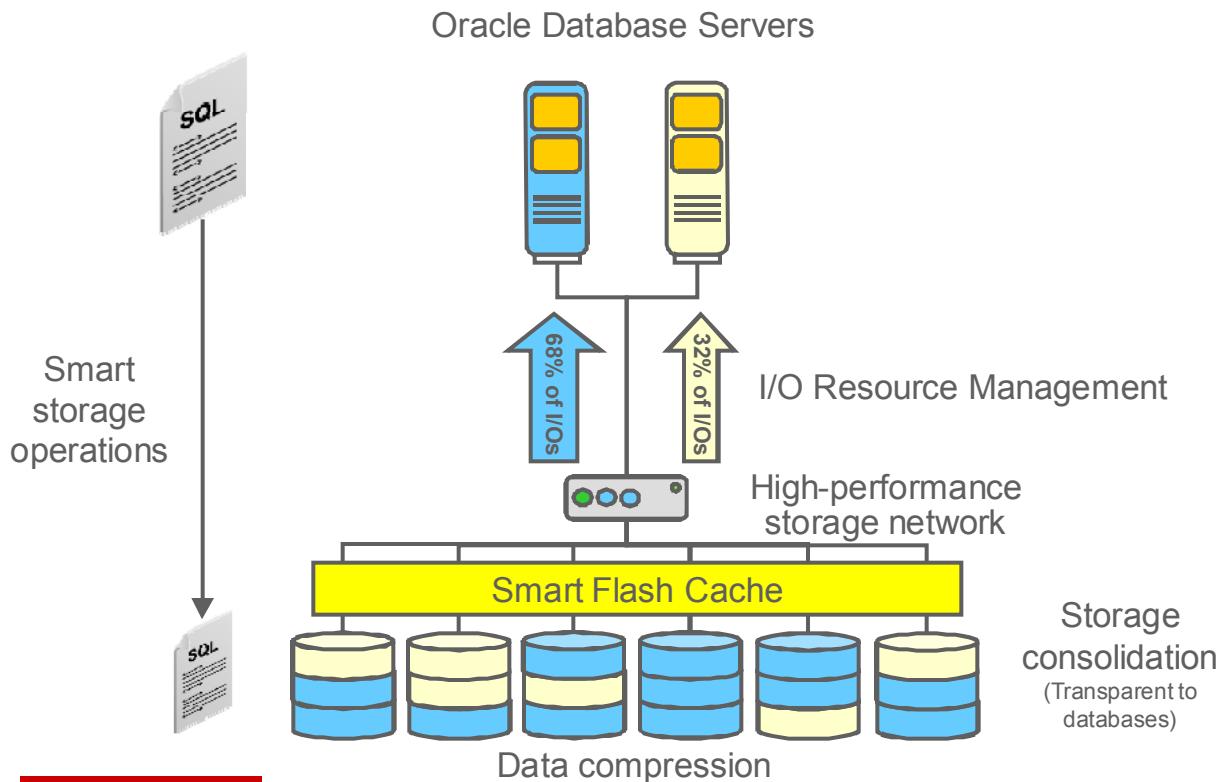
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Storage Server is a self-contained storage platform that houses disk or PCI connected flash storage and runs the Exadata Storage Server software provided by Oracle. A single Exadata Storage Server is also called a cell. A cell is the building block for a storage grid. More cells provide greater capacity and I/O bandwidth. Databases are typically deployed across multiple cells, and multiple databases can share the storage provided by a single cell. The databases and cells communicate with each other via a high-performance InfiniBand network.

Each cell is a highly optimized storage platform for Oracle Database files although you can use Database File System (DBFS), a feature of Oracle Database, or Oracle ASM Cluster Filesystem (ACFS), a feature of Oracle Grid Infrastructure, to store your business files on Exadata Storage Server.

Like other storage arrays, each cell is a computer with CPUs, memory, a bus, disks, network adapters, and the other components normally found in a server. In the case of Exadata Storage Server, first-class Intel CPUs are used to power Exadata's smart storage capabilities. It also runs an operating system (OS), which in the case of Exadata Storage Server is Linux x86_64. The Oracle-provided software resident in the Exadata cell runs under this operating system. The OS is accessible in a restricted mode to administer and manage Exadata Storage Server.

Exadata Storage Server Features: Overview



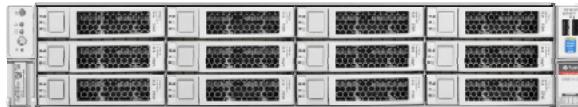
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide illustrates the main features associated with Exadata Storage Server:

- A key advantage of Exadata Storage Server is the ability to offload some database processing to the storage servers. With Exadata Storage Server, the database can offload single table scan predicate filters and projections, join processing based on bloom filters, along with CPU-intensive decompression and decryption operations. This ability is known as Smart Scan. In addition to Smart Scan, Exadata Storage Server has other smart-storage capabilities including the ability to offload incremental backup optimizations, file creation operations, and more. This approach yields substantial CPU, memory, and I/O bandwidth savings in the database server, which can result in massive performance improvements compared with conventional storage.
- Even for queries that do not use Smart Scan, Exadata Storage Server has many advantages over conventional storage. Exadata Storage Server is highly optimized for fast processing of large queries. It has been carefully architected to ensure no bottlenecks in the controller or in other components inside the storage server. It makes intelligent use of high-performance flash memory to boost performance and also uses a state-of-the-art InfiniBand network that has much higher throughput than conventional storage networks.

- Oracle Exadata Smart Flash Cache holds frequently accessed data in very fast flash storage. This happens automatically without the user having to take any action. Oracle Exadata Smart Flash Cache is smart because it knows when to avoid trying to cache data that will not be reused or will not fit in the cache.
- Exadata Storage Server supports Hybrid Columnar Compression. This feature provides very high levels of data compression, which enables the database to reduce the number of I/Os required to scan a table. For example, Hybrid Columnar Compression often delivers compression ratios of 10 to 1. In such cases, only 1 GB of I/O is required to scan a 10 GB data set.
- Exadata Storage Server ensures that I/O resources are made available whenever, and to whichever database needs them based on priorities and policies that you can define. The Database Resource Manager (DBRM) and Exadata Storage Server I/O Resource Management (IORM) work together to manage intradatabase and interdatabase I/O resource usage to ensure that your defined service-level agreements (SLAs) are met when multiple applications and databases share Exadata storage.
- Oracle Automatic Storage Management (ASM) is used to evenly distribute the storage load for every database across the storage pool provided by the Exadata Storage Servers. Every database can use all the available disks to maximize performance, or separate disk groups may be used to isolate the storage for different databases. Exadata Storage Server works equally well with single-instance or Oracle Real Application Clusters (RAC) databases. Users and database administrators use the same tools and knowledge they are already familiar with.

Exadata X6-2 High Capacity Storage Server Hardware: Overview



ORACLE
EXADATA

Processors	20 Intel CPU Cores 2 x Ten-Core Intel Xeon E5-2630 v4 (2.2 GHz)
System Memory	128 GB DDR4 Memory (8 x 16 GB)
Disk Drives	96 TB 12 x 8 TB 7,200 RPM High Capacity SAS Disk Drives
Flash	12.8 TB 4 x 3.2 TB Sun Flash Accelerator F320 NVMe PCIe Cards
Disk Controller	Disk Controller Host Bus Adapter with 1GB Write Cache
InfiniBand Network	Dual-Port QDR (40Gb/s) InfiniBand Host Channel Adapter
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	2 x Redundant Hot-Swappable Power Supplies

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows an overview of the Exadata X6-2 High Capacity Storage Server hardware.

Note: Throughout this course, all hardware specifications, performance metrics, and capacity metrics are current as of April 2016, and are subject to change.

Data sheets containing detailed specifications and metrics can be found on the Oracle.com Exadata home page at <http://www.oracle.com/exadata>.

Exadata X6-2 Extreme Flash Storage Server Hardware: Overview



ORACLE®
EXADATA

Processors	20 Intel CPU Cores 2 x Ten-Core Intel Xeon E5-2630 v4 (2.2 GHz)
System Memory	128 GB DDR4 Memory (8 x 16 GB)
Flash Drives	25.6 TB 8 x 3.2 TB Sun Flash Accelerator F320 NVMe PCIe Drives
InfiniBand Network	Dual-Port QDR (40Gb/s) InfiniBand Host Channel Adapter
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	2 x Redundant Hot-Swappable Power Supplies

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata X5-2 introduced Extreme Flash Storage Servers. The Extreme Flash Storage Server replaced the previously available Exadata Storage Server with High Performance (HP) disks.

PCI flash delivers ultra-high performance by placing flash memory directly on the high speed PCI bus, rather than behind a slower disk controller, and Extreme Flash drives use the latest NVMe (Non-Volatile Memory Express) flash protocol to achieve extremely low I/O overhead.

With Exadata X6-2, each Extreme Flash Storage Server contains eight 3.2 TB PCI Flash drives. Therefore, each Extreme Flash Storage Server X6-2 has a raw capacity of 25.6 TB, which is double the capacity of Exadata X5-2, and much larger than the 14.4 TB capacity of the previously available HP disk configuration.

Exadata Storage Server X6-2 Configuration Options

	Extreme Flash	High Capacity Disks
Raw Disk Capacity ¹	25.6 TB	96 TB
Uncompressed Data Capacity ²	9.3 TB	36.3 TB
SQL Disk Bandwidth ³	N/A	1.8 GB/sec
SQL Disk IOPS ⁴	N/A	2600
SQL Flash Bandwidth ³	25 GB/sec	21.5 GB/sec
SQL Flash Read IOPS ⁴	320000	320000
SQL Flash Write IOPS ⁵	295000	295000



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Storage Server X6-2 is available in two configurations: Extreme Flash (EF) or with high capacity (HC) disks. The table in the slide lists the key capacity and throughput specifications for both configuration options.

Note 1: Raw capacity is calculated by using $1 \text{ TB} = 1000 \times 1000 \times 1000 \times 1000 \text{ bytes}$.

Note 2: Actual space for uncompressed end-user data, computed after single mirroring (ASM normal redundancy) while also providing adequate space to re-establish the mirroring protection after a disk failure. Calculated by using $1 \text{ TB} = 1024 \times 1024 \times 1024 \times 1024 \text{ bytes}$. The actual user data capacity varies by application.

Note 3: Bandwidth is the peak physical scan bandwidth that is achieved by running SQL, and assuming that there is no database compression. Effective user data bandwidth is higher when database compression is used. In all cases, actual performance varies by application.

Note 4: Based on 8K Oracle Database I/O requests. Note that the I/O size greatly affects Flash IOPS, so IOPS based on smaller I/Os is not relevant for databases. The value is approximate because the throughput varies on different Exadata Database Machine configurations.

Note 5: Based on 8K Oracle Database I/O requests. Flash write I/Os are measured at the storage servers after ASM mirroring, which usually issues multiple storage I/Os to maintain redundancy. The value is approximate because the throughput varies on different Exadata Database Machine configurations.

Exadata Database Machine X6-2

Database Server Hardware: Overview



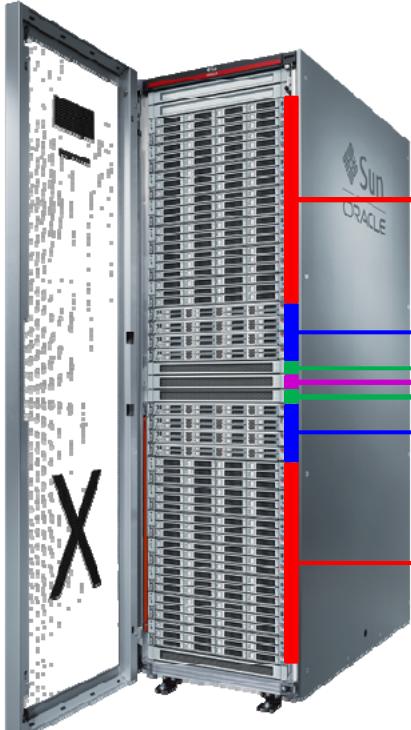
Processors	44 Intel CPU Cores 2 x 22-Core Intel Xeon E5-2699 v4 Processors (2.2GHz)
System Memory	256 GB (Expandable to 768 GB)
Disk Drives	4 x 600 GB 10K RPM SAS Disk Drives (Expandable to 8 Disks)
Disk Controller	Disk Controller Host Bus Adapter with 1GB Write Cache
Network Interfaces	<ul style="list-style-type: none">• Dual-Port QDR (40Gb/s) InfiniBand Host Channel Adapter• Four 1/10 Gb Ethernet Ports (copper)• Two 10Gb Ethernet Ports (optical)
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	2 x Redundant Hot-Swappable Power Supplies



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Database Machine X6-2 uses standardized database server hardware based on 64-bit Intel-based servers. The slide shows an overview of the Exadata Database Machine X6-2 database server hardware.

Exadata Database Machine X6-2 Full Rack



14 Exadata Storage Servers
(all High Capacity or all Extreme Flash)

8 Database Servers

2 QDR (40Gb/s) InfiniBand Switches (36 port)

1 Cisco Ethernet Switch (48 port)

2 Power Distribution Units (PDUs) at rear

Spare Parts Kit:

- High Capacity: 1 disk drive and 1 flash card
- Extreme Flash: 1 flash drive

ORACLE®

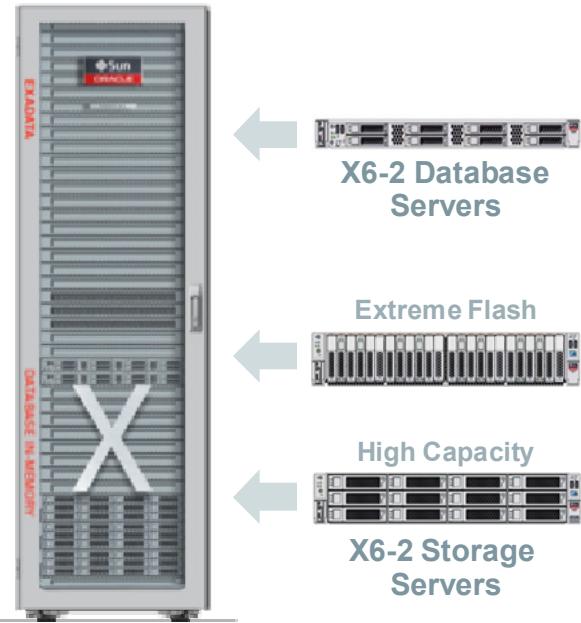
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Database Machine is offered in a variety of standard configurations. The contents of an Exadata Database Machine X6-2 Full Rack configuration are illustrated in the slide.

Elastic Configuration

- Start with a Quarter Rack:
 - 2 database servers
 - 3 storage servers
- Add database or storage servers:
 - Maximum of 22 servers or full rack
 - A mixture of High Capacity and Extreme Flash storage servers allowed in the same rack
- Assembly options are as follows:
 - The entire rack can be factory assembled.
 - Servers can also be incrementally added at the customer site.
 - X6-2 servers can be added to V2, X2, X3, X4, X5, and X6 Exadata systems.

**X6-2 Quarter Rack
(EF or HC)**



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata X5 introduced the concept of elastic configuration, which continues with Exadata X6. Elastic configuration provides an extremely flexible and efficient mechanism to customize the computing power or storage capacity of Exadata.

The starting point for elastic configuration is an Exadata Quarter Rack, which contains two database servers and three storage servers. From this starting point, any number of database servers or storage servers can be added, up to a maximum of 22 servers or until the rack is physically full.

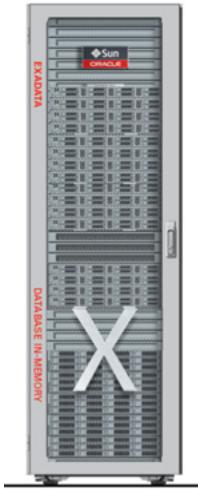
For example, an elastic configuration that maximizes storage would contain two database servers and 18 storage servers, at which point the rack would be physically full. However, an elastic configuration that maximizes database servers would contain three storage servers and 19 database servers, at which point the 22 server maximum is reached.

Elastic configuration allows a mixture of Extreme Flash (EF) and High Capacity (HC) storage servers within the same rack; however, different storage types cannot be used in a single disk group. Also, there must be a minimum of at least two Exadata Storage Servers present for each different storage server type.

Elastic configurations can be factory-assembled by Oracle, or customers can incrementally add servers to an existing Exadata Database Machine, including V2, X2, X3, X4, X5, and X6 machines.

Elastic Configuration Examples

Database In-Memory



16 Database Servers
and
5 High Capacity
Storage Servers

Extreme Flash OLTP



8 Database Servers
and
8 Extreme Flash
Storage Servers

Warehousing and Analytics



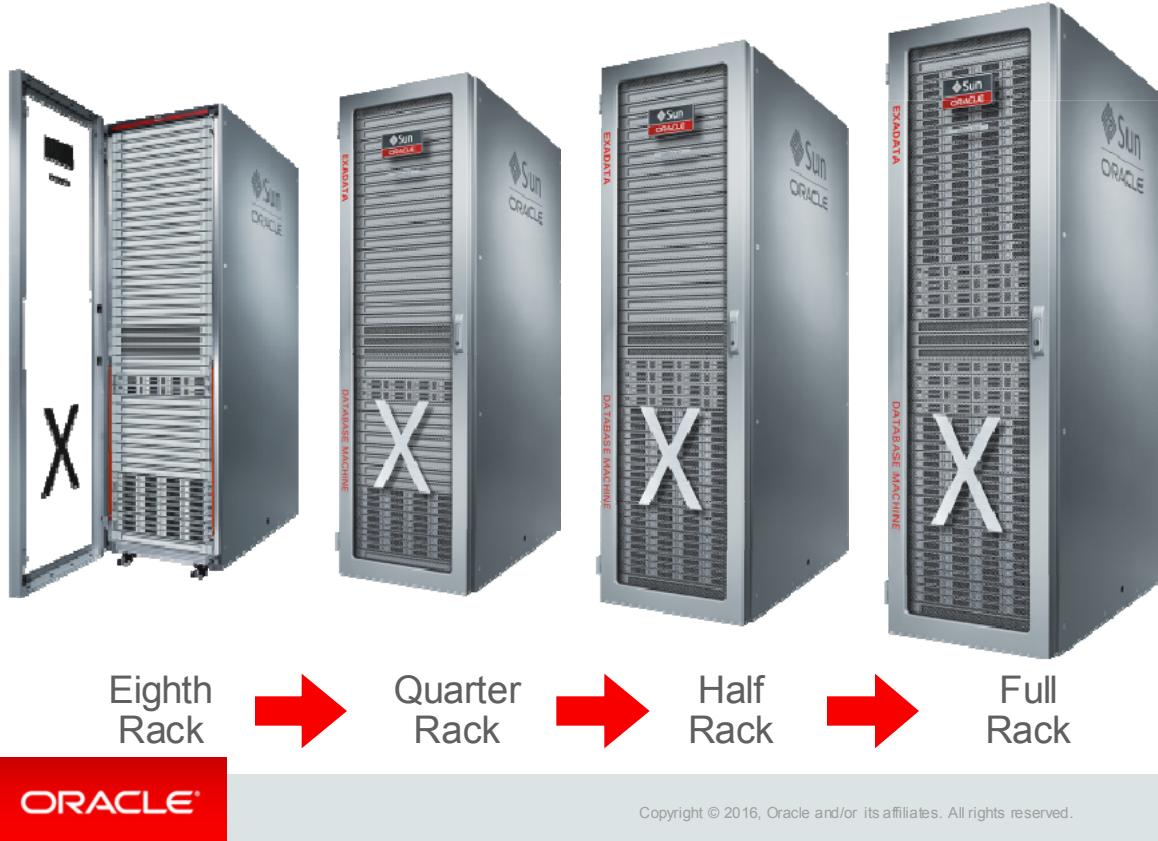
8 Database Servers
and
14 High Capacity
Storage Servers

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows three elastic configuration examples, which are suited to different workloads, and which illustrate the flexibility of elastic configurations. Note that these are general examples only and are not specific recommendations for the workload types listed in the slide.

Start Small and Grow



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

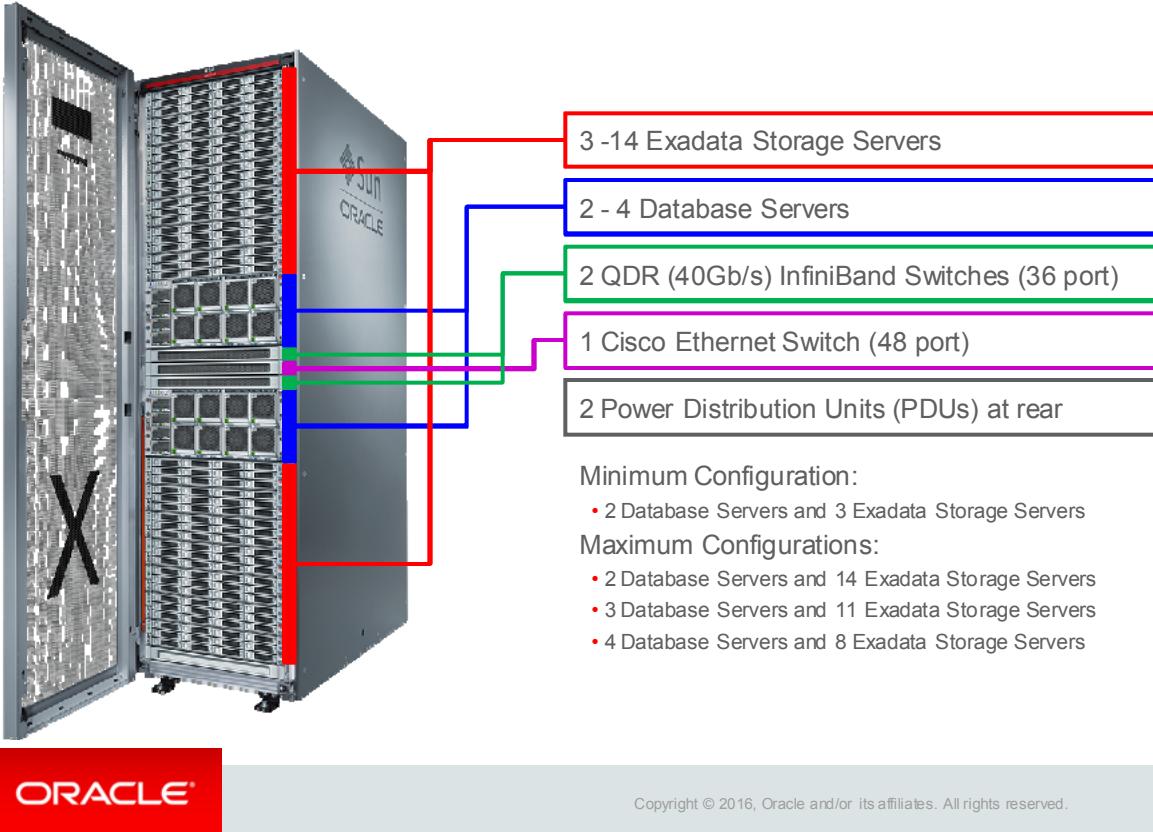
The scale-out architecture of Exadata Database Machine, coupled with elastic configuration, enables you to start with an Eighth Rack and grow from there.

Just like a Quarter Rack, an Eighth Rack contains two database servers, three Exadata cells, and two InfiniBand switches. However, in an Eighth Rack, half of the database server CPU cores are disabled. The Eighth Rack Extreme Flash storage servers have half the of the CPU cores and half of the flash drives enabled, whereas the Eighth Rack High Capacity storage servers have half of the CPU cores enabled and half of the disks and flash cards removed.

A Half Rack is a standard configuration containing four database servers and seven Exadata Storage Servers.

Note that an Exadata Database Machine environment can grow beyond the capacity of a single rack. Multiple rack configurations are discussed later in the lesson titled “Exadata Database Machine Architecture.”

Exadata Database Machine X6-8



In addition to Exadata Database Machine X6-2, Oracle offers Exadata Database Machine X6-8. The major difference is that the 2-socket database servers in Exadata Database Machine X6-2 are replaced by larger 8-socket database servers. Each of the larger database servers contains 144 CPU cores, at least two TB RAM, and more network interfaces.

Exadata Database Machine X6-8 can be configured with Exadata X6-2 Storage Servers, including the Extreme Flash (EF) storage servers.

Exadata Database Machine X6-8 also supports elastic configuration. In this arrangement, Exadata Database Machine X6-8 can be configured with a minimum of two database servers and three storage servers. From there, up to two more 8-socket database servers can be added, along with additional storage servers. The slide lists the maximum configurations for an X6-8 rack based on the number of database servers in the rack.

Exadata Database Machine X6-8

Database Server Hardware: Overview



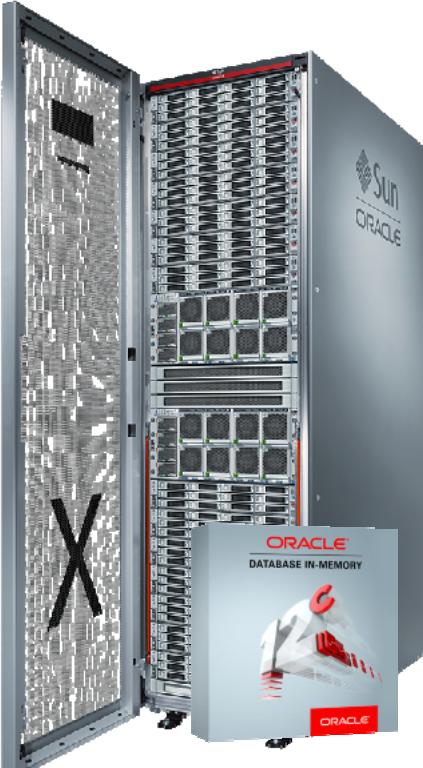
Processors	144 Intel CPU Cores 8 x 18-Core Intel Xeon E7-8895 v3 Processors (2.6 GHz)
System Memory	2 TB (Expandable to 6 TB)
Disk Drives	8 x 600 GB 10K RPM Disk Drives
Disk Controller	Disk Controller Host Bus Adapter with 1GB Write Cache
Network Interfaces	<ul style="list-style-type: none">• 8 x QDR (40Gb/s) InfiniBand Ports• 10 x 1 Gb Ethernet Ports• 8 x 10 Gb Ethernet Ports
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	4 x Redundant Hot-Swappable Power Supplies



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Database Machine X6-8 uses powerful 8-socket database server hardware based on 64-bit Intel-based servers. The slide shows the overview of each Exadata Database Machine X6-8 database server.

Exadata Database Machine X6-8 Use Cases



- Ideal for situations that require very large memory capacity
- Established use cases:
 - Large scale OLTP
 - Large scale analytics
- Newer use cases:
 - Database consolidation
 - Database as a Service (DBaaS)
 - Public or private cloud
 - Database In-Memory
 - With fault-tolerance

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

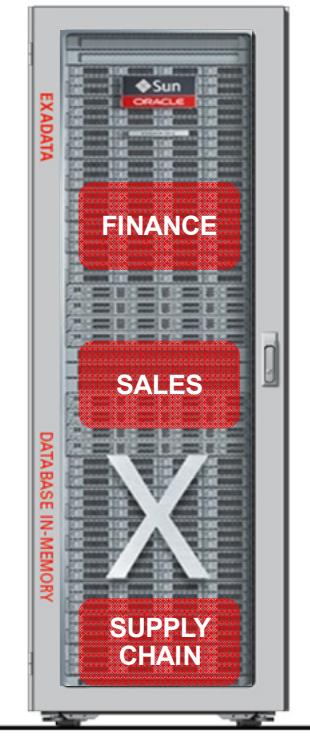
Exadata Database Machine X6-8 is the ideal choice for situations that require very large memory capacity, such as large scale OLTP and analytics.

With the introduction of newer software capabilities, Exadata Database Machine X6-8 is well suited as a platform for database consolidation and for delivering database as a service (DBaaS) in a public or private cloud implementation.

The large memory capacity of the X6-8 database servers is also ideally suited for use in conjunction with the Oracle Database In-Memory option. On Exadata platforms, Oracle Database In-Memory supports fault-tolerant configuration, enabling data to be simultaneously populated in the in-memory data store on multiple database servers. Under this arrangement, if a database instance fails, the surviving in-memory stores are automatically and transparently used to satisfy user queries.

Oracle Exadata Virtual Machines

- The database servers can run multiple VMs.
- The VMs provide isolation for consolidated environments, enabling:
 - Hard limits on CPU and memory
 - Independent OS and administration
 - Multiple clusters within a set of servers
 - Agility and flexibility to address changing business needs
- VMs use High-speed InfiniBand with SR-IOV.
 - Similar speed to InfiniBand on native hardware
 - Full support of Exadata features
- VMs are considered a trusted partition.
 - Database options need to be licensed only for the VMs that use them.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

With Exadata software release 12.1.2.1.0 and later, Oracle Virtual Machine (OVM) can be used on the X6-2, X5-2, X4-2, X3-2, and X2-2 database servers to deliver higher levels of isolation between workloads. Virtual machine (VM) isolation is desirable for workloads that cannot be trusted to restrict their security, CPU, or memory usage in a shared environment. Examples include hosted or cloud environments, cross-department consolidation, test and development environments, and non-database or third party applications running on an Exadata Database Machine. OVM can also be used to consolidate workloads that require different versions of clusterware, for example, SAP applications that require specific clusterware patches and versions. Exadata VMs can dynamically expand or shrink CPUs and memory based on the workload requirement of the applications running in the VM.

The isolation provided by VMs comes at the cost of increased resource usage, management, and patching because a separate OS, clusterware, and database installation is needed for each VM. Therefore, it is generally not advisable to place each database in a separate VM. Rather, a blended approach is recommended that leverages the Oracle Database resource management and multitenant technologies to consolidate within a database where this is feasible.

Exadata VMs use high speed InfiniBand networking with Single Root IO Virtualization (SR-IOV) to ensure that performance within a VM is similar to the raw hardware performance. Exadata VMs support the full range of Exadata features, such as Smart Scan.

The VMs on Exadata are considered trusted partitions, and therefore, software can be licensed at the VM level instead of the physical processor level.

With Exadata software release 12.1.2.1.0 and later, Oracle Linux is the only supported OS for use inside an Exadata VM.

InfiniBand Network: Overview

InfiniBand:

- Is the Exadata Database Machine interconnect fabric:
 - Provides highest performance available – 40 Gb/sec each direction
 - Is widely used in high-performance computing since 2002
- Is used for storage networking, RAC interconnect and high-performance external connectivity:
 - Less configuration, lower cost, higher performance
- Looks like normal Ethernet to host software:
 - All IP-based tools work transparently – TCP/IP, UDP, SSH, and so on
- Has the efficiency of a SAN:
 - Zero copy and buffer reservation capabilities
- Uses Zero-loss Zero-copy Datagram Protocol (ZDP)
 - High performance, zero-copy implementation of RDSc3
 - Open source software developed by Oracle
 - Very low CPU overhead



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

InfiniBand is the only storage network supported inside Exadata Database Machine because of its performance and proven track record in high-performance computing. It has the efficiency of a SAN, using zero copy and buffer reservation. Zero copy means that data is transmitted without intermediate buffer copies in the various network layers. Buffer reservation is used so that the hardware knows exactly where to place buffers ahead of time. These are two important characteristics that distinguish InfiniBand from Ethernet.

The Zero-loss Zero-copy Datagram Protocol (ZDP) is used in conjunction with InfiniBand. It is a zero-copy implementation of the industry standard Reliable Datagram Sockets (RDSc3) protocol. ZDP is open source software that is developed by Oracle. It is like UDP but more reliable. ZDP has a very low CPU overhead, with tests showing only a 2 percent CPU utilization while transferring 1 GB/sec of data.

InfiniBand provides a unified network fabric for Exadata storage and the Oracle RAC interconnect. This facilitates easier configuration and fewer cables and switches. The Exadata InfiniBand network can also be used for high-performance external connectivity, such as to connect backup servers, ETL servers, or tape libraries. InfiniBand is also used to connect other Oracle Engineered Systems, such as Oracle Exalogic Elastic Cloud or Oracle Exalytics In-Memory Machine.

The InfiniBand implementation in Exadata Database Machine uses the open source RDSc/Open Fabrics Enterprise Distribution (OFED). The OFED packages are included in the Exadata software stack.

Exadata Storage Expansion Racks



ORACLE®

- Designed to quickly and easily add substantial storage capacity to Exadata Database Machines
- Configuration Options:
 - Start with a Quarter Rack:
 - Four storage servers
 - Three InfiniBand switches
 - Add Exadata Storage Servers:
 - Any combination of High Capacity or Extreme Flash
 - Up to a total of 19 Exadata Storage Servers in a rack

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Oracle Exadata Storage Expansion Rack is engineered to be the fastest and simplest way to add substantial storage capacity to an Exadata Database Machine. A natural extension of the Exadata Database Machine, the Exadata Storage Expansion Rack is offered with a minimum configuration, known as a Quarter Rack, that contains four Exadata Storage Servers and three InfiniBand switches.

After the initial quarter rack, additional High Capacity (HC), Extreme Flash (EF) or combination of HC and EF storage servers can be added as needed up to a total maximum of 19 storage servers in each storage expansion rack.

Note that Exadata Storage Expansion Racks come equipped with three 36-port QDR (40 Gb/sec) InfiniBand Switches, which enables them to be interconnected with any combination of up to 18 Exadata Database Machine racks or Exadata Storage Expansion Racks via the InfiniBand fabric. Details on interconnecting multiple racks are provided later in the course.

Exadata Database Machine Support: Overview

Support Offering	Key Features
Oracle Hardware Warranty	<ul style="list-style-type: none">Included with Exadata, 1 year termAnytime web access, Local business hours phone access4 hour response during normal business hoursOn-site response and parts exchange within 2 business days subject to availability and location
Oracle Premier Support	<ul style="list-style-type: none">Support for Oracle Database and Exadata cell softwareAnytime web or phone accessSoftware enhancements, fixes and upgradesProactive tools, including alerts and configuration guidance
Oracle Premier Support for Systems	<ul style="list-style-type: none">Support for server and storage hardware and firmware; includes Oracle LinuxAnytime web or phone accessOn-site hardware response within 2 hours for Severity1 issues if the Exadata system is within a 2 hour service coverage area
Oracle Customer Data and Device Retention	<ul style="list-style-type: none">Provides replacements for failed disk drivesCustomer retains the failed disk drivesProvides additional security for sensitive data

See also <http://www.oracle.com/support/policies.html>



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle provides a complete and integrated set of support offerings for Exadata. Customers can use a single contact point to access all of the support services outlined on the slide. There is also a single point of accountability, meaning that issues will never remain unresolved while separate support organizations wait for each other to progress.

The support services outlined in the slide are modular so that customers can choose the right level of support for their needs. Oracle Hardware Warranty is included with every Exadata Database Machine and is the minimum level of support. Typically, Exadata is deployed for enterprise-scale business-critical applications, and in these cases Oracle recommends that customers purchase Oracle Premier Support and Oracle Premier Support for Systems. Oracle Customer Data and Device Retention is recommended where security or privacy requirements compel customers to ensure that sensitive data never leaves their enterprise.

In addition to the support offerings outlined here, Oracle provides installation and configuration services for Exadata. These services are highly recommended to ensure an efficient and trouble-free start with Exadata. Additional services exist to help customers with upgrades. A specialized service is also available for customers who want to interconnect multiple Exadata racks.

Oracle Platinum Services: Enhanced Support at No Additional Cost

Platinum Service	Features	Benefits
Remote Fault Monitoring	24/7 Fault monitoring Event filtering and qualification Reporting on event management A single global knowledge base, tool set and client portal	Fastest identification, notification, and restoration of issues Focus on critical events Full visibility of faults detected by Oracle Leverage Oracle's collective knowledge
Accelerated Response	24/7 Response Times: <ul style="list-style-type: none"> • 5-min fault notification • 15-min restoration or escalation to development • 30-min joint debugging Escalation process and hotline with dedicated escalation managers	Highest level of response with the fastest path to issue resolution
Patch Deployment	Assess and Analyze: Produce a quarterly patch plan Plan and Deploy: Proactively plan and deploy patches four times per year	Expert support staff available 24x7 Proactive identification of best practice configuration for optimal performance Minimize business disruption and ensure systems performance



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle Platinum Services is a special entitlement under Oracle Premier Support that provides customers with additional services at no extra cost. In addition to receiving the complete support essentials provided with Oracle Premier Support, qualifying Oracle Platinum Services customers also receive Oracle remote fault monitoring, accelerated response times and patch deployment services. The table in the slide outlines these additional features and associated benefits.

In order to qualify for Oracle Platinum Services, customers must:

- Be running a Platinum Certified Configuration on one of the following Oracle systems; Exadata Database Machine, Exalogic Elastic Cloud, SuperCluster, Zero Data Loss Recovery Appliance or ZFS Storage Appliance (Racked System). A Platinum Certified Configuration is a defined combination of certified components that have been tested and certified by Oracle. The current matrix of certified configurations is available at <http://www.oracle.com/us/support/library/certified-platinum-configs-1652888.pdf>
- Install the Oracle Advanced Support Gateway software to enable remote monitoring, restoration and patching services
- Agree to let Oracle deploy patches to covered systems on their behalf four times per year

For more details about Oracle Platinum Services, visit

<http://www.oracle.com/us/support/premier/engineered-systems-solutions/platinum-services/overview/index.html>

Quiz



What is the maximum number of Exadata Storage Servers supported in a single-rack Exadata Database Machine?

- a. 14
- b. 18
- c. 19

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b

A full rack Exadata Database Machine contains 14 storage servers. However, with elastic configuration you could have a single-rack Exadata Database Machine with two database servers and 18 storage servers. A storage expansion rack may contain up to 19 storage servers, but it is not an Exadata Database Machine.

Quiz



What are three unique benefits of Exadata Storage Server compared to traditional storage servers?

- a.** Large disk sizes
- b.** Smart storage capabilities
- c.** Higher storage network bandwidth
- d.** High RAM capacity
- e.** Integrated database I/O resource management

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b, c, e

Summary

In this lesson, you should have learned to:

- Provide an overview-level description of Exadata Storage Server and the different Exadata Database Machine configurations
- Outline the key capacity and performance specifications for Exadata Database Machine



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

3

Exadata Database Machine Architecture

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the Exadata network architecture
- Describe the Exadata software architecture
- Describe the Exadata Storage Server storage entities and their relationships
- Describe how multiple Exadata racks can be interconnected



ORACLE®

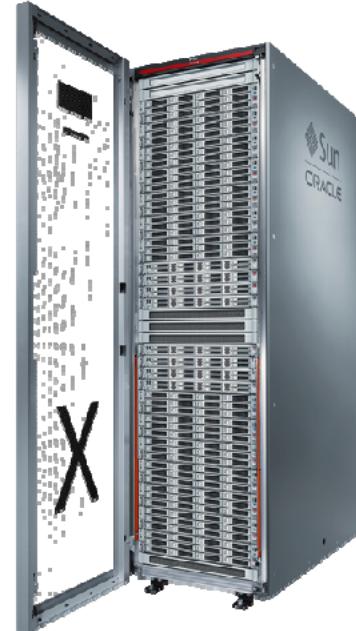
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Architecture: Overview

Exadata Database Machine provides a highly available, high-performance platform for Oracle Database based on the clustered architecture supported by Oracle RAC.

Key components include:

- Powerful and intelligent shared storage provided by Exadata Storage Server
- Storage mirroring provided by ASM
- High bandwidth and low latency cluster interconnect and storage networking provided using InfiniBand technology
- Powerful and well-balanced database servers joined together in a cluster



ORACLE®

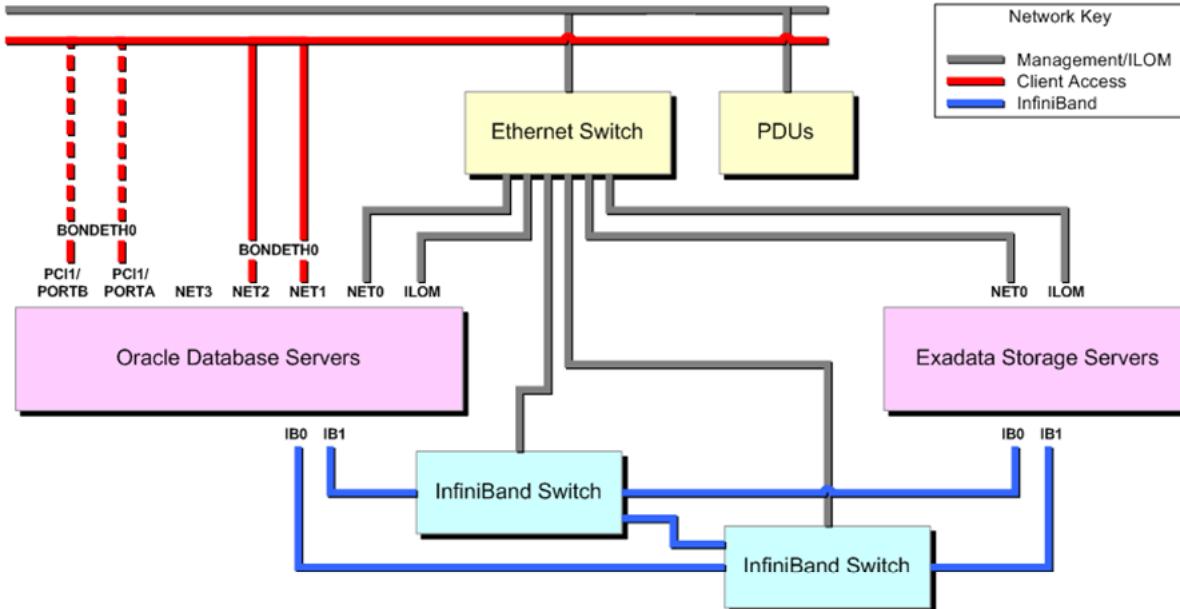
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Database Machine provides a resilient, high-performance platform for clustered and non-clustered implementations of Oracle Database. The fundamental architecture underpinning Exadata is the same core architecture that underpins Oracle Real Application Clusters (RAC) software. Key elements of the Exadata architecture are introduced below and further details follow in the remainder of this lesson:

- **Shared storage:** Exadata provides intelligent, high-performance shared storage to both single-instance and RAC implementations of Oracle Database using Exadata Storage Server technology. Storage supplied by Exadata Storage Servers is made available to Oracle databases using the Automatic Storage Management (ASM) feature of Oracle Database. ASM adds resilience to Exadata storage by providing a mirroring scheme that can be used to maintain redundant copies of data on separate Exadata Storage Servers. This protects against data loss if a storage server is lost. Normal protection ensures that two copies of data are maintained on separate Exadata Storage Servers while high level protection maintains three copies of data on three separate Exadata Storage Servers.
- **Storage network:** Exadata contains a storage network based on InfiniBand technology. This provides high bandwidth and low latency access to the Exadata Storage Servers. Fault tolerance is built into the network architecture through the use of multiple redundant network switches and network interface bonding.

- **Server cluster:** The database servers in Exadata are designed to be powerful and well balanced so that there are no bottlenecks within the server architecture. They are equipped with all of the components required for Oracle RAC, enabling customers to easily deploy Oracle RAC across a single Exadata Database Machine. Where processing requirements exceed the capacity of a single Database Machine, customers can join multiple Database Machines together to create a single unified server cluster.
- **Cluster interconnect:** The high bandwidth and low latency characteristics of InfiniBand are ideally suited to the requirements of the cluster interconnect. Because of this, Exadata is configured by default to also use the InfiniBand fabric as both the storage network and the cluster interconnect.
- **Shared cache:** In a RAC environment, the instance buffer caches are shared. If one instance has an item of data in its cache that is required by another instance, that data will be shipped to the required node using the cluster interconnect. This key attribute of the RAC architecture significantly aids performance because the memory-to-memory transfer of information via the interconnect is significantly quicker than writing and reading the information using disk. With Exadata, the shared cache facility uses the InfiniBand-based high-performance cluster interconnect.

Exadata Network Architecture



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide shows how the major components of Exadata Database Machine X6-2 are connected to each other. The diagram shows the Ethernet switch, Power Distribution Units (PDUs), and InfiniBand switches. For the sake of clarity, only one database server and one Exadata server are shown. Exadata contains three network types:

- **Management/ILOM:** This Ethernet/IP network is used to manage Exadata. Using this network, administrators can access the database servers and the Exadata Storage Servers by using remote login facilities such as Secure Shell (SSH). The database servers and Exadata Storage Servers also provide an Ethernet interface for Integrated Lights-Out Management (ILOM). ILOM provides a powerful set of remote administration facilities. Using ILOM, administrators can remotely monitor and control the state of the server hardware. The InfiniBand switches and PDUs also provide Ethernet ports for remote monitoring and management purposes.
- **Client Access:** This Ethernet network is primarily used to provide database connectivity via the Oracle Net software. When Exadata is initially configured, a bonded network interface (BONDETH0) is configured by default (using NET1 and NET2). Alternatively, customers can choose to configure a single client network interface (typically using NET1).

A bonded client access network interface can provide protection if a network interface fails. However, using bonded interfaces may require additional configuration in the customer's network. Each X6-2 database server also contains a spare Ethernet port (NET3) that can be used to configure an additional client access network.

The standard Ethernet ports (NET0 to NET3) support 1 gigabit Ethernet (1 GbE), and 10 gigabit Ethernet (10 GbE) using the 10GBASE-T standard.

Each X6-2 database server is also equipped with two 10 GbE optical interfaces (PCI1/PORTA and PCI1/PORBT), which can be used for client connectivity. These interfaces can be bonded together (as shown in the diagram) or connected to separate networks. These ports support SFP+ form factor short range or long range transceivers, or copper TwinAx. Customers must have the required network infrastructure to make use of these interfaces.

- **InfiniBand:** The InfiniBand network provides a reliable high-speed storage network and cluster interconnect for Exadata. It can also be used to provide high-performance external connectivity to backup servers, ETL servers, and middleware servers such as Oracle Exalogic Elastic Cloud. Each database server and Exadata Storage Server is connected to two InfiniBand switches. The InfiniBand network is described in greater detail later in this lesson.

The network architecture for Exadata Database Machine X6-8 is essentially the same as X6-2. The following differences exist:

- Each database server has a total of eight 10 GbE network ports and eight 1 Gb Ethernet ports, which may be used for client access.
- Each database server is configured with four dual-port InfiniBand network interfaces (8 ports), which are connected to the InfiniBand network.

InfiniBand Network Architecture

- 36-port managed QDR (40 Gb/s) InfiniBand switches
 - 2 leaf switches used to connect server InfiniBand ports
- X6-2 Exadata Storage Servers and X6-2 Database Servers
 - Each server has one dual-port QDR (40 Gb/s) InfiniBand HCA.
 - Each HCA port is connected to a different switch for high availability.
- X6-8 Database Servers
 - Each server has four dual-port QDR (40 Gb/s) InfiniBand HCAs.
 - Each pair of HCA ports are connected to different switches for high availability.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Within Exadata, the primary role of the InfiniBand network is to provide a reliable, high-speed interconnect to support database-server-to-storage-server communications. It is also used to facilitate communication and coordination between database servers by using Oracle Clusterware and the Oracle Real Application Clusters (RAC) software. It is important to note that the InfiniBand network is internal to Exadata and there is no requirement for additional InfiniBand infrastructure within the customer's network.

Each Exadata X6-2 or X6-8 configuration contains two 36-port managed QDR (40 Gb/s) InfiniBand switches, known as leaf switches. Each database server and each Exadata cell has at least two InfiniBand network ports, and each port is connected to a different leaf switch.

This architecture facilitates high availability because there are multiple paths in the InfiniBand fabric. If an InfiniBand server port fails, the server can still function admirably by using the remaining port. If a switch fails, the InfiniBand network can be maintained by the remaining switch with excellent performance.

Active Bonding InfiniBand Connectivity

Each Exadata server has at least one dual-port QDR (40 Gb/s) InfiniBand HCA:

- Originally, active-passive bonding was used to deliver high availability.
 - Bandwidth was limited by the underlying PCIe architecture.
- Starting with X4 models, both HCA ports can be simultaneously active:
 - This effectively doubles the InfiniBand network bandwidth.
 - The latest InfiniBand HCAs can exploit the full bandwidth.
 - Each InfiniBand port is configured with a separate IP address.
 - No bonded network interface is configured.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

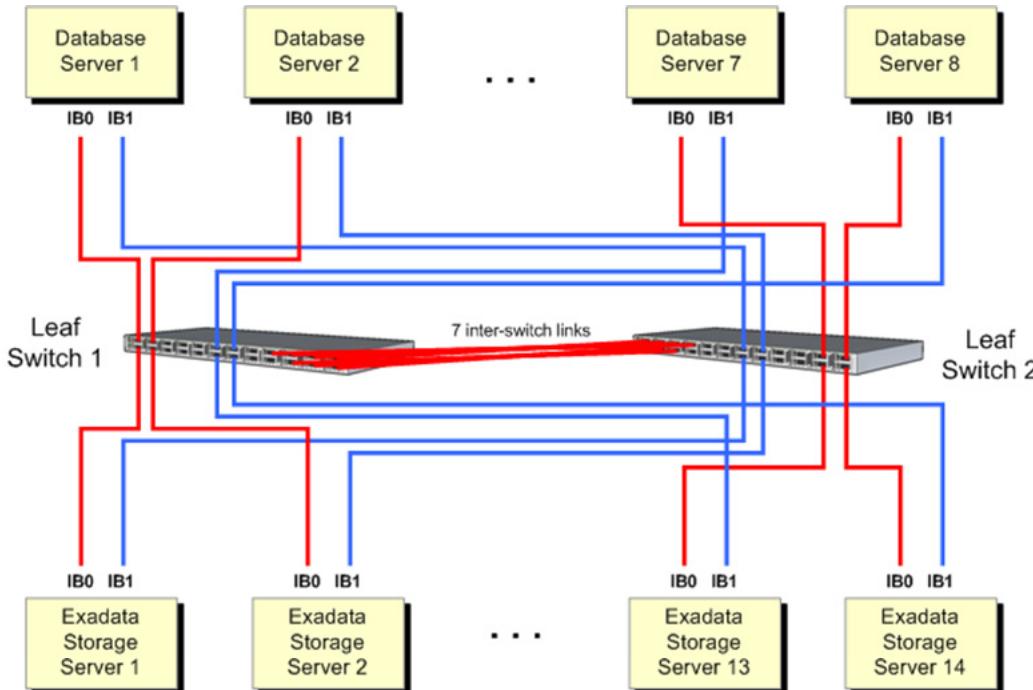
In early generation Exadata Database Machines, each pair of HCA ports was configured to use active-passive bonding to provide high availability. In this configuration, the full bandwidth of the InfiniBand network was not used; however, this was acceptable because the InfiniBand host channel adapters (HCAs) were effectively bandwidth-limited by the underlying PCIe infrastructure.

Starting with the X4 generation of hardware, the Exadata servers are equipped with InfiniBand cards that support PCIe 3.0. These cards raise the PCIe bandwidth ceiling, allowing the combined bandwidth of both ports to be exploited.

Note that this configuration does not use a Linux bonded network interface (BONDIB0, for example). Rather, each port is configured as a separate network interface (ib0 and ib1, for example) with a separate IP address. Communication between Oracle Database and the Exadata Storage Servers is load-balanced across the available interfaces, which effectively doubles the InfiniBand network bandwidth during normal operations.

Note also that active-active connectivity (also known as Active Bonding) is not recommended on pre-X4 servers running Exadata release 11.2.3.3.0 because the hardware is not able to fully exploit the active-active port configuration. Also, Oracle Clusterware requires the same interconnect interfaces on every database server, so it is advisable to use active-passive bonding when interconnecting newer generation hardware with pre-X4 Database Machines.

Leaf Switch Topology



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide illustrates how the two leaf switches in each Exadata X6-2 Full Rack are connected to the database servers, Exadata servers, and to each other. On each server, each InfiniBand port is connected to a different leaf switch.

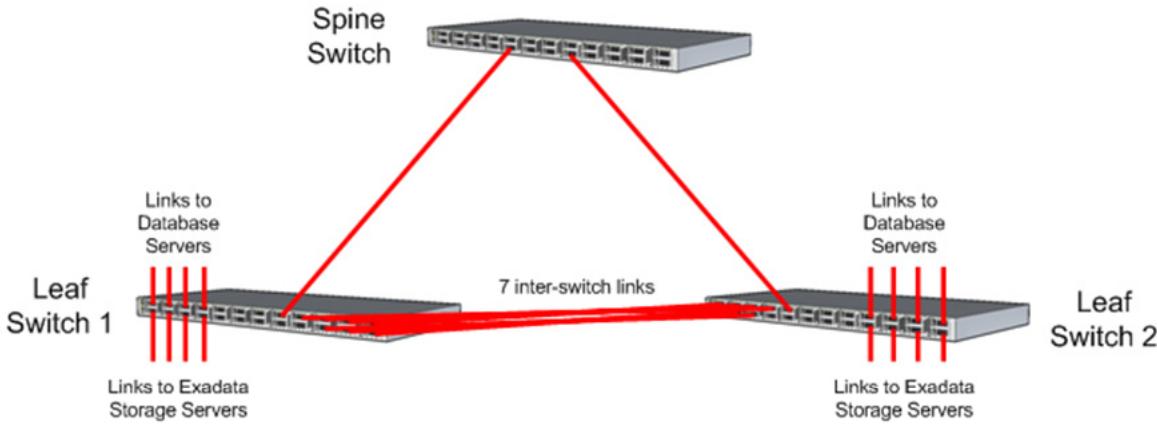
For each server, the InfiniBand ports (IB0 and IB1) are connected to the switches such that approximately half of the IB0 ports are connected to each switch. Likewise, approximately half of the IB1 ports are connected to each switch.

In addition, the leaf switches are connected to each other via seven inter-switch links. All the connections are pre-cabled in the factory.

Exadata X6-8 has essentially the same topology as illustrated in the slide, except that each of the database servers has four dual-port InfiniBand network interfaces with connections spread between the leaf switches.

Note: The switch port mappings shown in the diagram are illustrative only. For full details, refer to the cabling tables in *Oracle Exadata Database Machine System Overview*.

Spine and Leaf Topology



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide shows the addition of a spine switch. The purpose of the spine switch is to facilitate connection of multiple racks, which can be used to create a single large-scale database grid.

Spine switches were included in older Full Rack and Half Rack Exadata configurations. In those configurations, each leaf switch is nominally connected to the spine switch using a single network link.

For Exadata X6-2 or X6-8 configurations, spine switches and associated cabling must be added to interconnect multiple racks.

Note: Quarter Racks can be interconnected with other racks in limited situations that are discussed later in this lesson.

Scale Performance and Capacity Beyond a Single Rack



Redundant and Fault Tolerant

- Failure of any component is tolerated.
- Data is mirrored across storage servers.

Scalable

- Scale to 18 racks by adding cables.
- Scale to hundreds of storage servers to support multi-petabyte databases.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Database Machines can be interconnected to scale beyond the performance or capacity of a single rack configuration. You can scale up to 18 racks by simply adding some cabling between them.

Typical Scaling Scenarios

- Large monolithic system:
 - Multiple Exadata X6-2 racks
 - Possibly in conjunction with Exadata Storage Expansion Racks
 - Multiple Exadata X6-8 racks
 - Possibly in conjunction with Exadata Storage Expansion Racks
- Platform consolidation:
 - Multiple Exadata X6-2 or X6-8 racks
 - Possibly in conjunction with Exadata Storage Expansion Racks
- Maximum capacity:
 - An Exadata X6-2 or X6-8 rack in conjunction with Exadata Storage Expansion Racks
- Tiered storage:
 - One or more Exadata X6-2 or X6-8 racks with Extreme Flash drives in conjunction with one or more Exadata Storage Expansion Racks with High Capacity disks



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

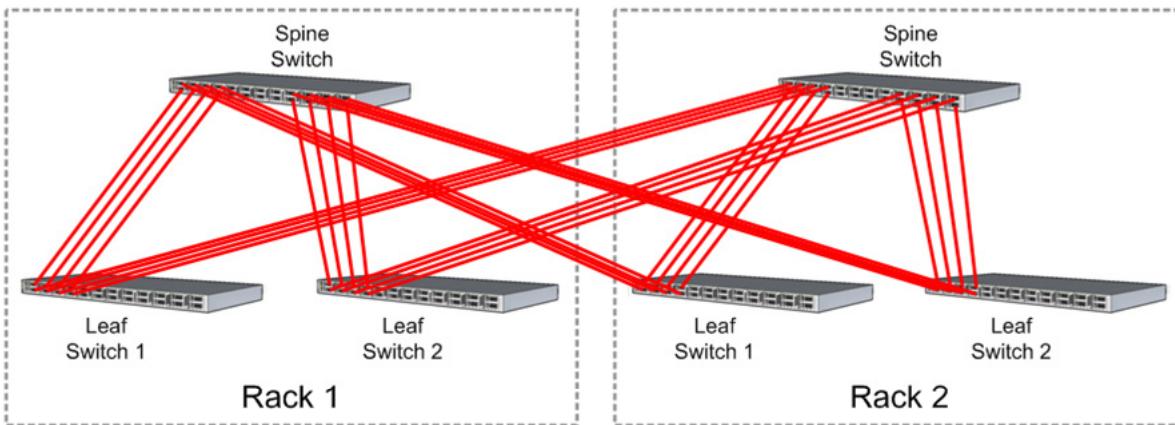
Four typical scaling scenarios are presented here. This is by no means an exhaustive list; rather it provides examples of common situations requiring multiple interconnected racks:

- Large monolithic systems may require more computing power, more storage capacity, or more I/O throughput than a single rack can deliver. In these cases, multiple Exadata racks are typically deployed. The deployment may also use Exadata Storage Expansion Racks to further boost storage capacity. For a large monolithic system, a mixture of X6-2 and X6-8 Database Machines is not considered because it is not permitted to have a single database environment that includes database servers from both the X6-2 and X6-8 racks.
- Platform consolidation may bring together multiple systems that together require more computing power, more storage capacity or more I/O throughput than a single rack can deliver. Multiple Exadata racks, possibly in conjunction with Exadata Storage Expansion Racks can be deployed in these cases. Also, it is possible to interconnect Exadata X6-2 and X6-8 models so that databases running on the different models can share a single consolidated storage grid spanning both rack types. In other words, you can have a single storage grid between the two models of Exadata Database Machine, but not a single database grid spanning both models.

- The processing requirements of many systems can be satisfied using a single Exadata rack. However, the system may simply run out of storage capacity. In these cases, the Exadata rack can be interconnected with one or more Exadata Storage Expansion Racks to deliver the required storage capacity.
- In many situations, customers use Extreme Flash drives on their Database Machines but desire High Capacity disks for backup and online data archiving purposes. This requirement is easily satisfied by adding Exadata Storage Expansion Racks to the Database Machines. Then the storage can be configured in a tiered manner so that the Extreme Flash drives on the Database Machines are allocated to data disk groups, while the High Capacity disks on the storage expansion racks can be allocated to disk groups supporting the Fast Recovery Area (FRA) or another area dedicated to online archiving.

Scaling Out to Eight Racks

- Single InfiniBand network based on a Fat Tree topology
 - Database and storage server cabling unchanged
- Two rack example:



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

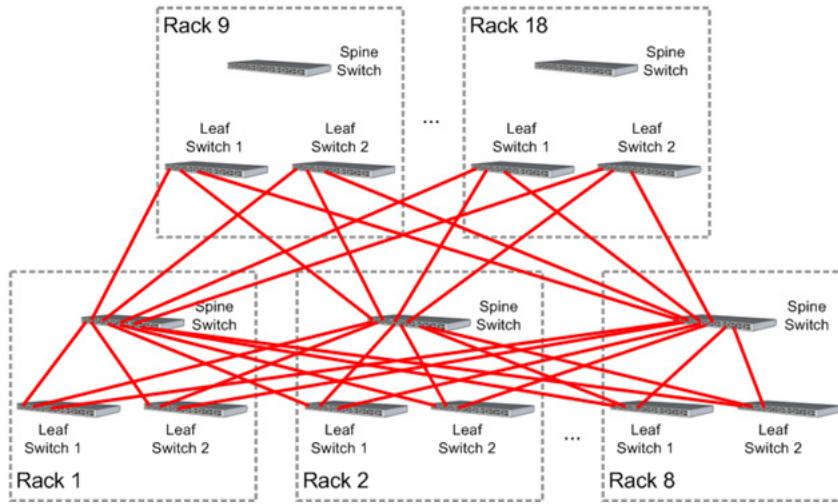
When up to eight racks are connected to form a large-scale database grid, the InfiniBand network is modified to resemble a Fat Tree topology by adding a spine switch to each rack. The database server and storage server connections to the leaf switches remain unchanged.

In a stand-alone Exadata rack, each leaf switch has seven links that connect it to the other leaf switch in the rack. When multiple racks are interconnected, these seven links plus one spare link are redistributed to the spine switches. The diagram in the slide illustrates this using a two rack example. As shown in the diagram, the seven inter-switch links that originally connected the leaf switches, plus one spare link on each switch, have been redistributed so that each leaf switch has four links to each spine switch. There are no direct links between spine switches and there are no direct links between leaf switches.

To interconnect eight racks, the eight leaf-to-spine links on each leaf switch are configured to provide exactly one link to each spine switch. For configurations between two and eight racks, the eight leaf-to-spine links on each leaf switch are distributed as evenly as possible amongst the total number of spine switches.

Scaling Out Between 9 and 18 Racks

- Single InfiniBand network based on a Fat Tree topology
 - Up to 18 racks supported with existing switches
 - Database and storage server cabling unchanged
- Topology:



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

To interconnect between 9 and 18 racks, start by connecting the first eight racks as described on the previous page.

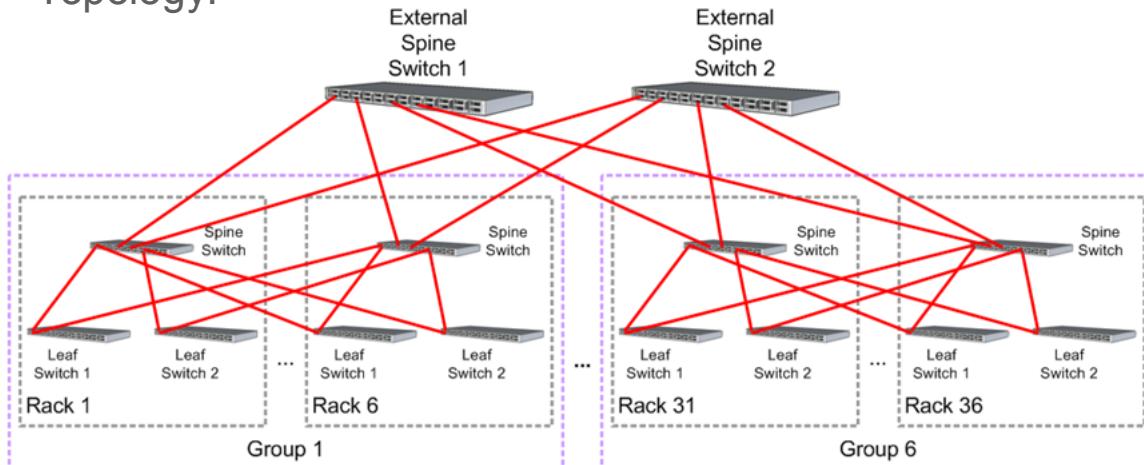
Then on each rack from 9 through 18:

1. Remove the seven inter-switch connections between the leaf switches.
2. Use the seven newly available ports, along with one spare port on each switch, to connect each leaf switch to every spine switch in racks 1 through 8.

The diagram in the slide illustrates the resulting network topology. Note that the InfiniBand connections inside racks 1 through 8 are not changed when adding racks 9 through 18.

Scaling Out Between 19 and 36 Racks

- Single InfiniBand network based on a Fat Tree topology
 - Scale out to 36 racks by adding two external spine switches
 - Database and storage server cabling is unchanged
 - One level is added to the Fat Tree topology
- Topology:



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Configurations containing up to 36 racks can be built by adding two 36-port external spine switches. The result is a single InfiniBand network which is still built on a Fat Tree topology. The external spine switches simply add another level to the tree.

The extended network topology is built by grouping the racks so that each group contains between two and eight racks. Within each group, the racks are interconnected as described on page 13. The groups of racks are further interconnected by adding a link between the spine switch in each rack and each of the two external spine switches. An example is shown in the diagram in the slide.

Because of the extra network hops that are required to communicate between the groups of racks, it is recommended that the racks should be grouped in a way that minimizes cross-group communications.

Interconnecting Quarter Racks and Eighth Racks

Quarter Racks and Eighth Racks only can be interconnected without a spine switch in the following limited situations:

- Interconnect two Quarter Racks, or two Eighth Racks
 - Connect each leaf switch in each rack to both leaf switches in the other rack using two links for each connection
- Connect one Quarter Rack or one Eighth Rack to a larger rack
 - Connect each leaf switch in the Quarter Rack or Eighth Rack to both leaf switches in the other rack using two links for each connection
- Connect one Quarter Rack or Eighth Rack to a group of up to eight other interconnected racks
 - Remove the seven inter-switch links between the leaf switches within the Quarter Rack or Eighth Rack
 - Connect each leaf switch in the Quarter Rack or Eighth Rack to each spine switch in the other racks
 - Use two links for each connection if there are four or less other racks
 - Use one link for each connection if there are more than four other racks



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Quarter Racks and Eighth Racks only can be interconnected without the addition of a spine switch in the following limited situations. Note that in this context a Quarter Rack can either be a Quarter Rack Database Machine or a Quarter Rack Storage Expansion Rack:

- **Interconnect two Quarter Racks or two Eighth Racks:** This is achieved by connecting each leaf switch in each rack to both of the leaf switches in the other rack. Two links are used for each inter-switch connection between the racks. The existing seven inter-switch links between the leaf switches are left unchanged.
- **Connect one Quarter Rack or one Eighth Rack to one Half Rack or one Full Rack:** This is also achieved by connecting each leaf switch in the Quarter Rack or Eighth Rack to both of the leaf switches in the other rack. Two links are used for each inter-switch connection between the racks. The existing seven inter-switch links between the leaf switches are left unchanged. If present, the spine switch in the larger rack and its links within the rack are also left unchanged.

Note that when you connect a Quarter Rack or Eighth Rack to a Full Rack, you will use some of the spare leaf switch ports on the Full Rack that are nominally reserved for external connectivity.

- **Connect one Quarter Rack or Eighth Rack to a group of up to eight other interconnected racks:** For this configuration, the group of other interconnected racks must meet the following requirements:
 - The group must be all Half Racks or all Full Racks; no mix of Half Racks and Full Racks.
 - The group of other racks must be interconnected using the Fat Tree topology described earlier in the lesson on Page 13.

Assuming the requirements are met, the Quarter Rack or Eighth Rack can be interconnected with the group using the following procedure:

- Remove the seven inter-switch links between the leaf switches within the Quarter Rack or Eighth Rack.
- Connect each leaf switch in the Quarter Rack or Eighth Rack to each spine switch in the other racks. Use two links for each connection if the Quarter Rack or Eighth Rack is being connected to a group of four racks or less. Use one link for each connection if there are more than four other racks.

InfiniBand Network External Connectivity

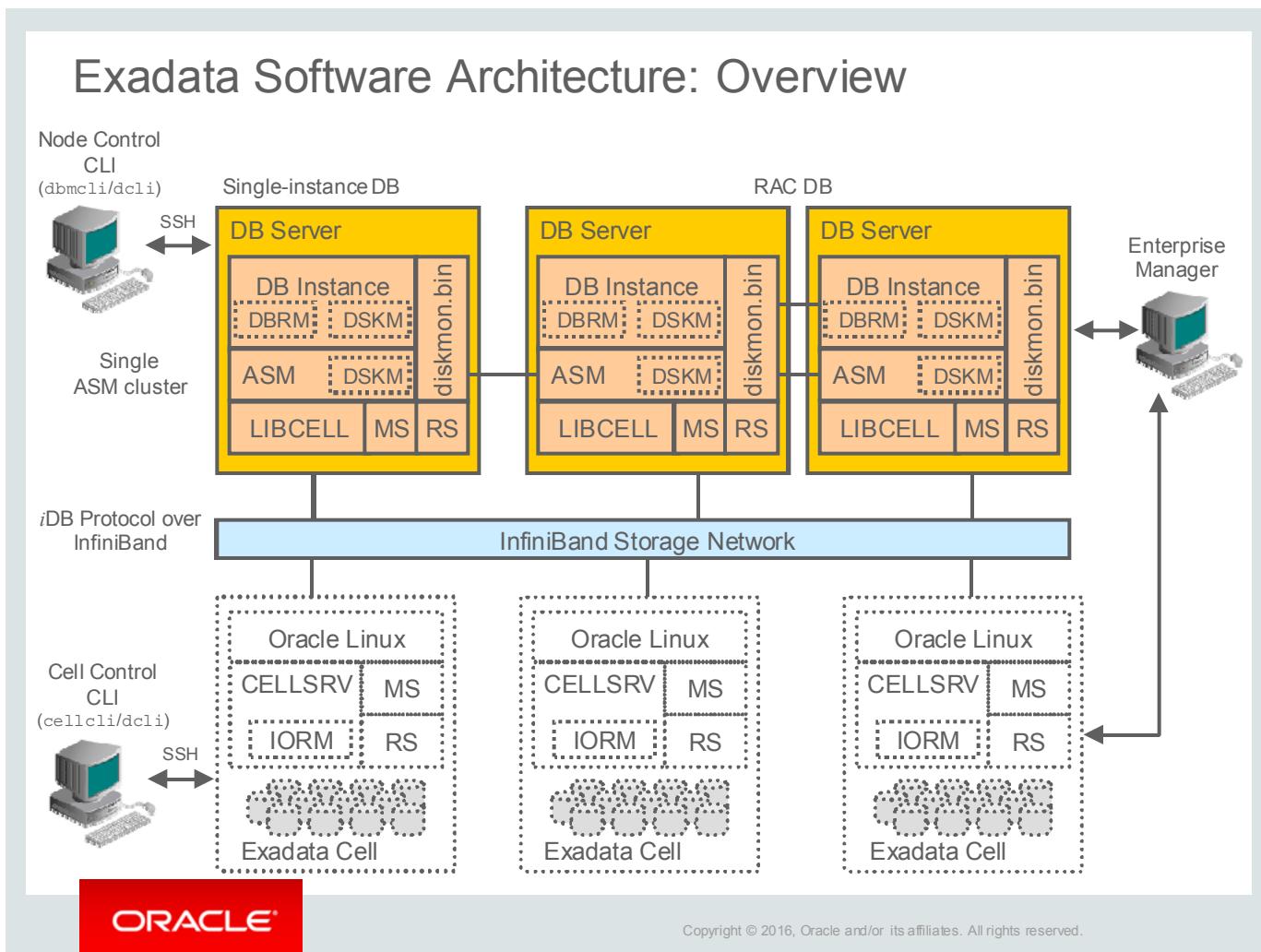
- Six ports on each leaf switch are reserved for external connectivity.
- External connectivity ports can be used for:
 - Connecting to media servers for disk or tape backup
 - Such as Oracle ZFS Storage Appliance
 - Connecting to external ETL servers
 - Client or application access
 - Such as Oracle Exalytics In-Memory Machine
- Use bonded network interfaces from the external device for high availability.
 - Connect each of the bonded links to separate leaf switches.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Six ports on each leaf switch are reserved for external connectivity. These ports can be used to connect Database Machines directly to the media servers used to manage offline storage devices such as tape libraries. The InfiniBand network can also be leveraged to provide high performance connectivity to ETL servers associated with a data warehouse, or middleware servers associated with performance-sensitive business application.

It is recommended to use bonded network interfaces on the external client or server with each bonded interface connecting to a separate leaf switch. Where possible, external clients and servers should connect to the InfiniBand network in the same way as the Exadata database servers and Exadata Storage Servers. At least, active-passive bonding should be used to provide continuity in the case of a switch or port failure; however, the external client or server can be configured with active bonding if that is supported.

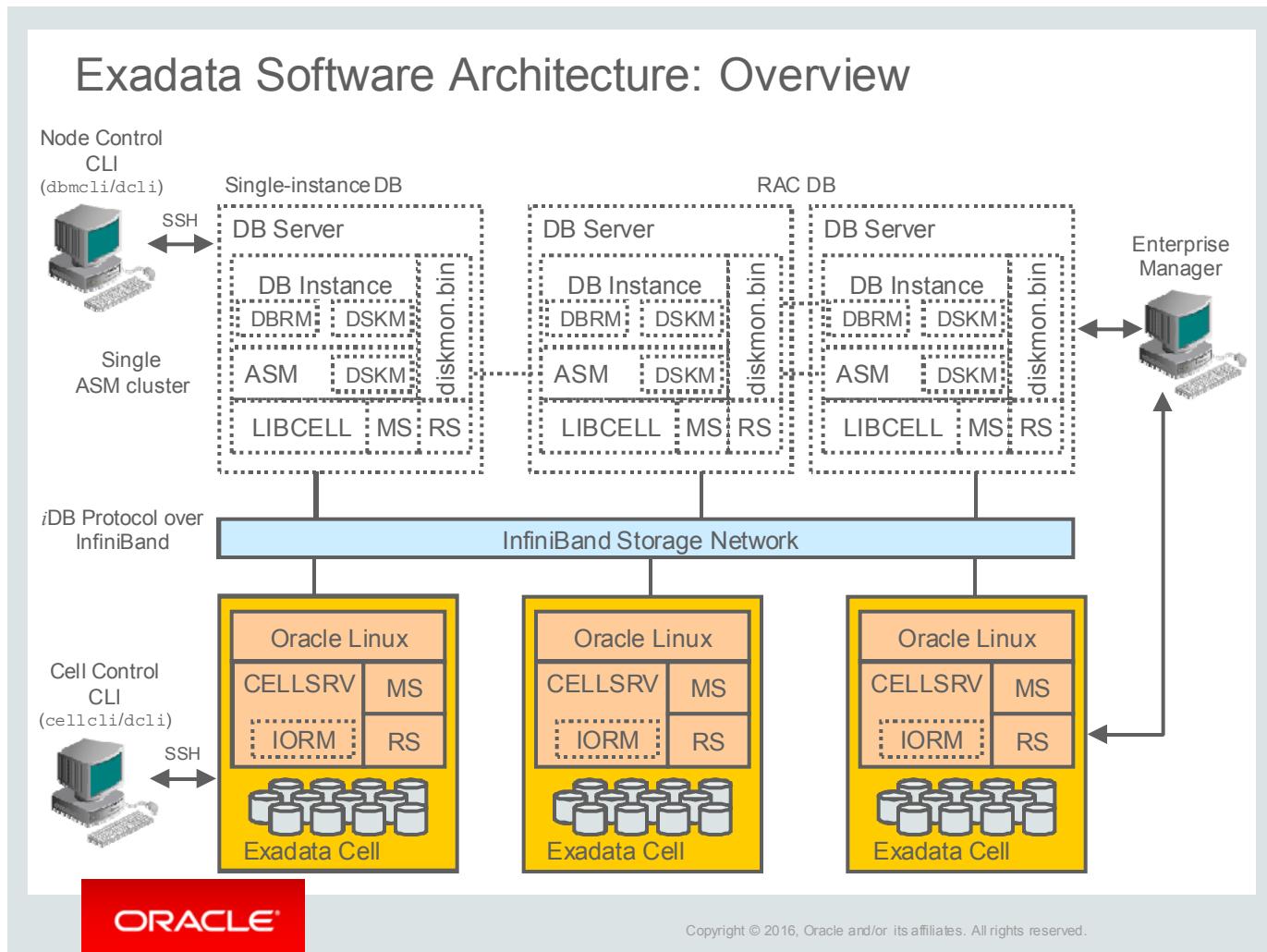


The Exadata Database Machine software architecture includes components on the database servers and on the Exadata cells. The overall architecture is illustrated in the slide. The following components reside on each database server:

- Exadata database servers use Oracle Linux x86_64 as the operating system. Beginning with the X5 generation of systems, customers can choose to run Oracle Virtual Machine (OVM) on 2-socket database servers, along with a series of virtual machines (VMs) that use Oracle Linux x86_64. The ability to run Oracle Linux without OVM remains, and is the only option for 8-socket database servers. Earlier generations of Exadata supported Oracle Solaris as an operating system option; however, this option no longer exists.
- Exadata database servers can run Oracle Database 11g Release 2 or Oracle Database 12c Release 1. The database servers can be configured with both Oracle Database software releases; however, in this case, version 12.1 of Oracle Grid Infrastructure is required. The precise patch release of Oracle software must be compatible with the Exadata Storage Server software and other Exadata software components. My Oracle Support note 888828.1 contains an up-to-date list of supported versions for the Exadata software components.
- Automatic Storage Management (ASM) is required and provides a file system and volume manager optimized for Oracle Database.

- Oracle Database communicates with Exadata cells by using a special library called LIBCELL (`$ORACLE_HOME/lib/libcell11.so` or `$ORACLE_HOME/lib/libcell12.so`). In combination with the database kernel and ASM, LIBCELL transparently maps database I/O operations to Exadata Storage Server enhanced operations. LIBCELL communicates with Exadata cells by using the Intelligent Database protocol (iDB). iDB is a unique Oracle data transfer protocol, built on Reliable Datagram Sockets (RDS), which runs on industry standard InfiniBand networking hardware. LIBCELL and iDB enable ASM and database instances to utilize Exadata Storage Server features, such as Smart Scan and I/O Resource Management.
- The Database Resource Manager (DBRM) is integrated with Exadata Storage Server I/O Resource Management (IORM). DBRM and IORM work together to ensure that I/O resources are allocated based on administrator-defined priorities.
- Diskmon checks the storage network interface state and cell liveness. It also performs DBRM plan propagation to Exadata cells. Diskmon uses a node-wide master process (`diskmon.bin`) and one slave process (DSKM) for each RDBMS or ASM instance. The master performs the monitoring and propagates state information to the slaves. The slaves use the SGA to communicate with the RDBMS or ASM processes. If there is a failure in the cluster, Diskmon performs I/O fencing to protect data integrity. Cluster Synchronization Services (CSS) decides what to fence. The master Diskmon starts with the clusterware processes. The slave Diskmon processes are background processes that are started and stopped in conjunction with the associated RDBMS or ASM instance.
- The Management Server (MS) on a database server provides a set of management and configuration functions. It works in cooperation with the DBMCLI command-line interface. Each database server is individually managed with DBMCLI. DBMCLI can be used only from within a database server to manage that server. However, you can run the same DBMCLI command remotely on multiple nodes with the `dcli` utility. In addition, MS is responsible for sending alerts, and collects some statistics.
- The Restart Server (RS) is used to start up or shut down MS and monitors MS to automatically restart it if required.

Note: The slide illustrates a typical configuration where a single ASM cluster is used to consolidate storage for all of your databases. Alternatively, you could configure multiple, separate ASM clusters, each with separate disk groups.



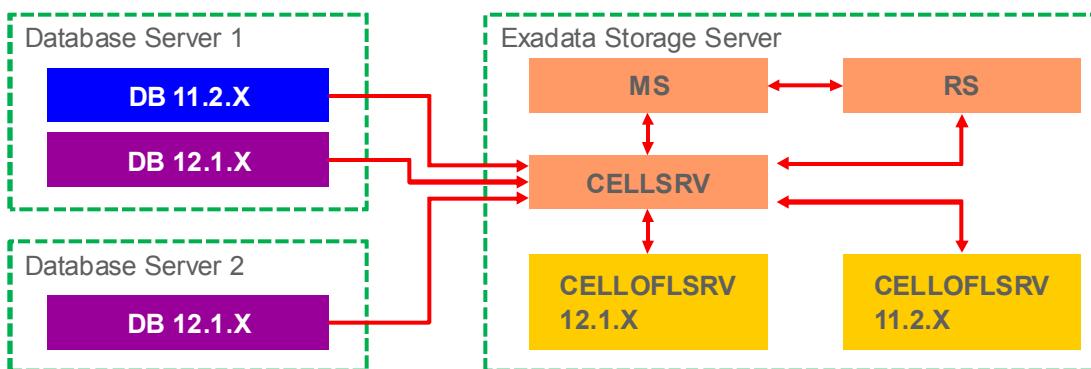
The software components that reside in each Exadata cell include the following:

- Oracle Linux x86_64 provides the Exadata Storage Server operating system.
- Cell Server (CELLSRV) is the primary Exadata Storage Server software component and provides the majority of Exadata storage services. CELLSRV is a multithreaded server. It serves simple block requests, such as database buffer cache reads, and Smart Scan requests, such as table scans with projections and filters. CELLSRV also implements IORM, which works in conjunction with DBRM to meter out I/O bandwidth to the various databases and consumer groups that are issuing I/Os. Finally, it collects numerous statistics relating to its operations. Oracle Database and ASM processes use LIBCELL to communicate with CELLSRV, and LIBCELL converts I/O requests into messages that are sent to CELLSRV by using the iDB protocol.
- The Management Server (MS) provides Exadata cell management and configuration functions. It works in cooperation with the Exadata cell command-line interface (CellCLI). Each cell is individually managed with CellCLI. CellCLI can be used only from within a cell to manage that cell. However, you can run the same CellCLI command remotely on multiple cells with the `dcli` utility. In addition, MS is responsible for sending alerts, and collects some statistics in addition to those collected by CELLSRV.
- The Restart Server (RS) is used to start up or shut down the CELLSRV and MS services, and monitors these services to automatically restart them if required.

Support for Mixed Database Versions

CELLSRV architecture to support mixed database versions:

- Separate offload servers for each major database release
- Requests are sent to the appropriate offload server
- Fully automatic, no configuration or maintenance

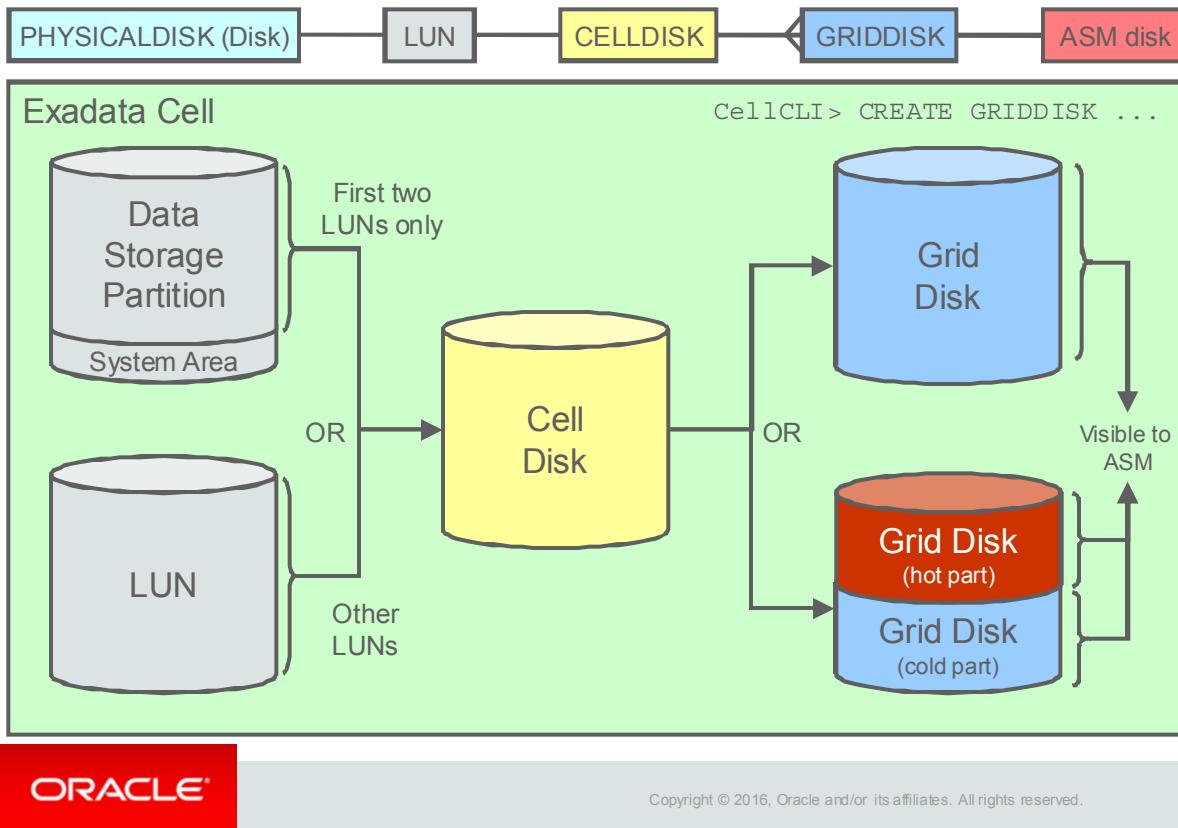


ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Beginning with version 12.1.1.1.0, Exadata Storage Server contains separate offload servers for each major database release so that it can fully support all offload operations. Offload requests coming from Oracle Database 11g Release 2 (11.2) are handled automatically by a release 11g offload server, and offload requests coming from Oracle Database 12c Release 1 (12.1) database are handled automatically by a 12c offload server. The offload servers are automatically started and stopped in conjunction with CELLSRV, so there is no additional configuration or maintenance procedures associated with them.

High Capacity Storage Server: Disk Storage Entities and Relationships



Each Exadata High Capacity Storage Server contains 12 hard disk drives. Exadata cell software automatically senses the physical disks in each storage server. As a cell administrator you can only view a predefined set of physical disk attributes. Each physical disk is mapped to a logical abstraction called a Logical Unit (LUN). A LUN exposes additional predefined metadata attributes to a cell administrator. You cannot create or remove a LUN; they are automatically created.

Each of the first two disks contains a system area that spans multiple disk partitions. The two system areas are mirror copies of each other which are maintained using software mirroring. The system areas consume approximately 29 GB on each disk. The system areas contain the OS image, swap space, Exadata cell software binaries, metric and alert repository, and various other configuration and metadata files.

A cell disk is a higher level abstraction that represents the data storage area on each LUN. For the two LUNs that contain the system areas, Exadata cell software recognizes the way that the LUN is partitioned and maps the cell disk to the disk partition reserved for data storage. For the other disks, Exadata cell software maps the cell disk directly to the LUN.

After a cell disk is created, it can be subdivided into one or more grid disks, which are directly exposed to ASM.

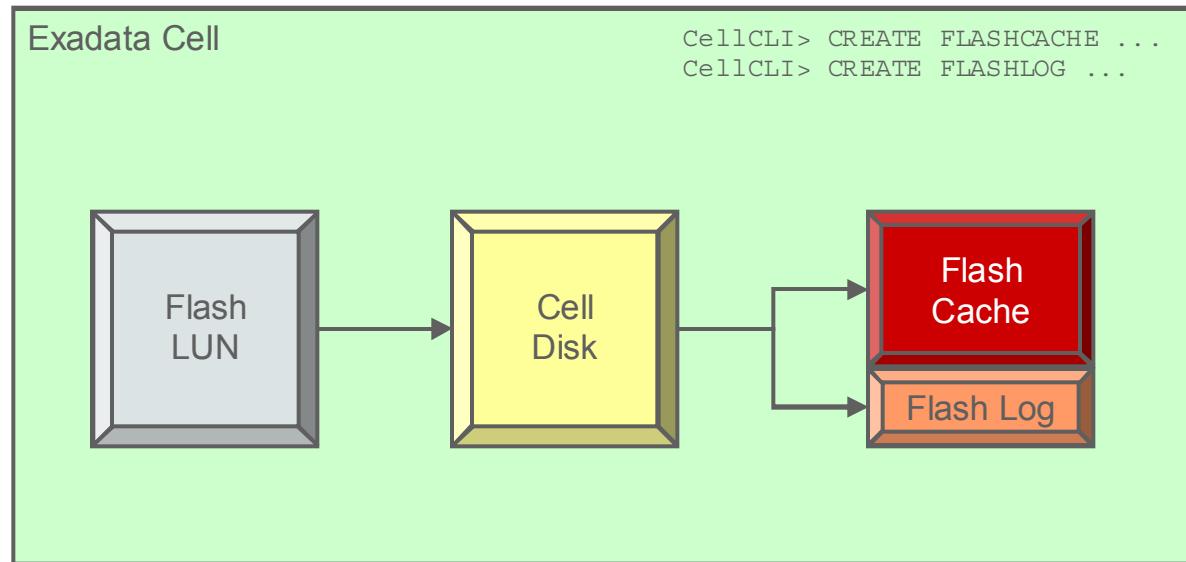
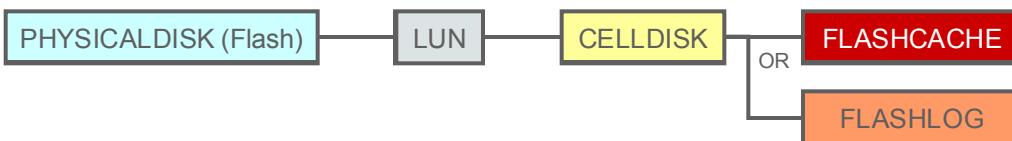
Placing multiple grid disks on a cell disk allows the administrator to segregate the storage into pools with different performance characteristics. For example, a cell disk could be partitioned so that one grid disk resides on the highest performing portion of the disk (the outermost tracks on a hard disk drive), whereas a second grid disk could be configured on the lower performing portion of the disk (the inner tracks). The first grid disk might then be used in an ASM disk group that houses highly active (hot) data, while the second grid disk might be used to store less active (cold) data files.

Placing multiple grid disks on a cell disk also allows the administrator to segregate the storage into separate pools that can be assigned to different databases.

In cases where the entire cell capacity is required for a single database or where it is difficult to clearly define hot and cold data sets, an Exadata cell administrator will usually define a single grid disk containing all the space on each cell disk.

Note: The diagram in the slide shows the cases where one or two grid disks are created from the space on a cell disk. However this does not imply a limit of two grid disks on a cell disk. In fact, when Exadata is initially configured, three grid disks are defined on most of the cell disks. Details of the initial configuration options and outcomes for Exadata are described later in the course.

High Capacity Storage Server: Flash Storage Entities and Relationships



ORACLE®

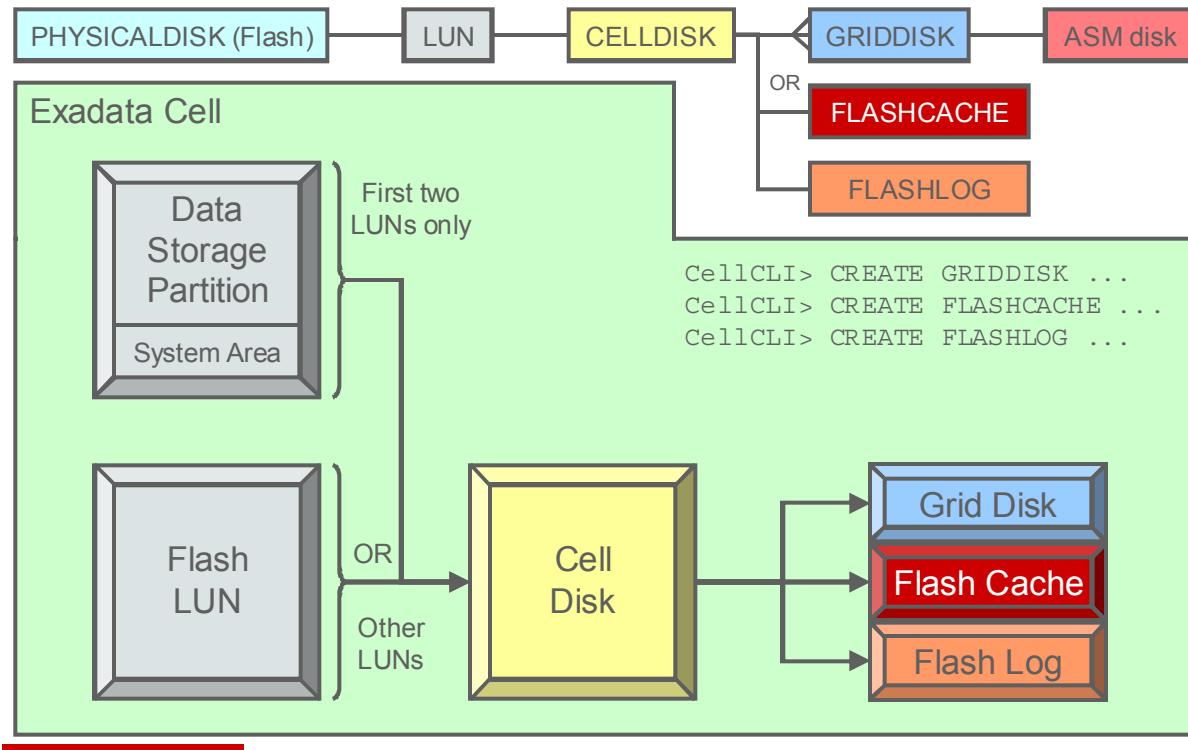
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Each Exadata X6-2 High Capacity Storage Server contains 12.8 TB of high performance flash memory distributed across four PCI flash memory cards. Therefore, each flash card has a capacity of 3.2 TB.

Essentially, each flash device is like a physical disk in the storage hierarchy. Each flash device is visible to the Exadata cell software as a LUN and the initial cell configuration process creates flash-based cell disks on all the flash devices.

By default, nearly all the available space on each flash-based cell disk is allocated to Exadata Smart Flash Cache and only a small portion (128 MB on each flash-based cell disk, 512 MB in total) is allocated to Exadata Smart Flash Log.

Extreme Flash Storage Server: Flash Storage Entities and Relationships

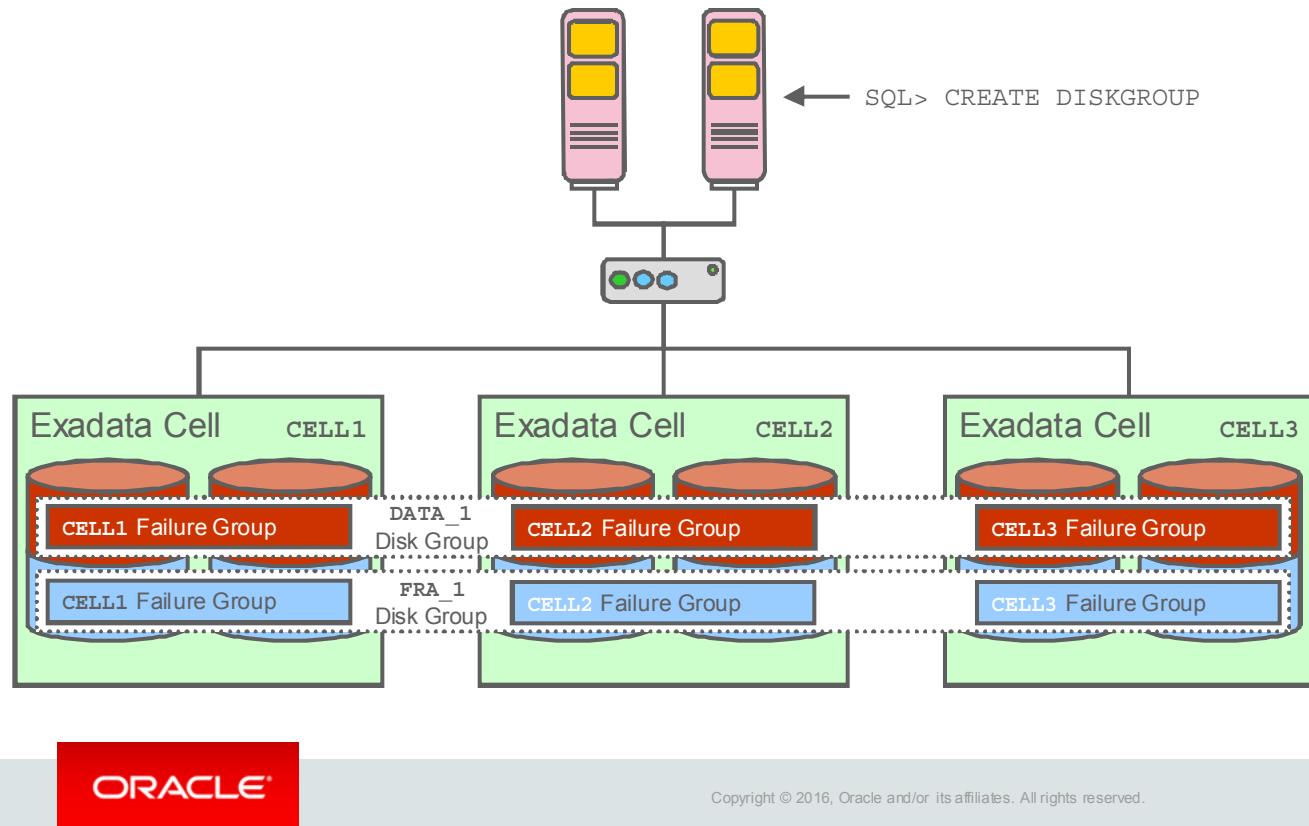


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata X6-2 Extreme Flash Storage Servers contain 8 high-performance flash drives. Each flash drive has a capacity of 3.2 TB. Like Exadata High Capacity Storage Server disks and flash devices, each flash drive is visible to the Exadata cell software as a LUN, and each LUN can contain a cell disk. Each of the first two flash drives also contains a system area, which is essentially the same as that found on the first two disks in each Exadata High Capacity Storage Server.

By default, 95% of the available space on each Extreme Flash cell disk is used for data storage. This space is allocated to grid disks that are exposed to ASM in the same way as the disk space from a high capacity storage server. Nearly all of the remaining 5% is allocated to Exadata Smart Flash Cache, and a small amount (64 MB on each flash-based cell disk, 512 MB in total) is allocated to Exadata Smart Flash Log.

Disk Group Configuration



After grid disks are configured, ASM disk groups can be defined to consume Exadata storage. The slide illustrates an example where two ASM disk groups are defined. The `DATA_1` disk group is defined across all of the red grid disks, and the `FRA_1` disk group is defined across the blue grid disks. When data is loaded into each disk group, ASM evenly distributes the data across all of the grid disks in each disk group.

Using `NORMAL` or `HIGH` ASM redundancy in conjunction with at least 3 failure groups for each disk group is recommended on Exadata. This ensures that at least two copies of data are maintained to protect from storage failure. `EXTERNAL` redundancy disk groups are not supported because they provide no protection from storage failures and prohibit the use of online storage maintenance procedures, such as rolling patches for example.

Note that the initial configuration process for Exadata creates three disk groups using `NORMAL` or `HIGH` ASM redundancy. The configuration of these disk groups and the associated redundancy options are discussed in detail later in the course.

To protect against the failure of an entire Exadata cell, separate ASM failure groups are automatically associated with each cell, which ensures that mirrored ASM extents are placed on different Exadata cells. This is also illustrated in the slide. By default, when failure groups are automatically created, their names correspond to the cell name. So, different disk groups can have the same failure group names. For example, the `DATA_1` disk group has a failure group named `CELL1`, and the `FRA_1` disk group also has a failure group named `CELL1`.

Quiz



Which are the three main Exadata services?

- a. OMS
- b. MS
- c. GMON
- d. CELLSRV
- e. RS

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b, d, e

Quiz



In which of the following scenarios will you maintain data availability if you use NORMAL ASM redundancy for all of your disk groups in conjunction with ASM failure groups spread across two or more Exadata cells?

- a. A single disk failure in a single cell
- b. Simultaneous failure of multiple disks in a single cell
- c. Simultaneous failure of one disk in each of two cells
- d. Complete failure of a single cell

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, d

The prescribed configuration may provide protection against simultaneous failure of one disk in each of two cells if there are no data extents mirrored to both of the failed disks. To guarantee data availability in cases where simultaneous failures affect two cells, you must use HIGH ASM redundancy in conjunction with failure groups spread across at least three Exadata cells.

Quiz



Active bonding is used for the InfiniBand network interfaces on Exadata servers to provide:

- a. Fault tolerance and high availability
- b. Increased bandwidth and performance
- c. Both of the above

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: c

Quiz



Which of the following scalability options are supported?

- a. Upgrading a Quarter Rack by adding more database servers
- b. Upgrading a Quarter Rack by adding more storage servers
- c. Interconnecting two Quarter Racks
- d. Interconnecting two or more Full Racks

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c, d

Summary

In this lesson, you should have learned how to:

- Describe the Exadata network architecture
- Describe the Exadata software architecture
- Describe the Exadata Storage Server storage entities and their relationships
- Describe how multiple Exadata racks can be interconnected



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 3 Overview: Introducing Exadata Cell Architecture

In these practices, you will be familiarized with the Exadata cell architecture. You will:

- Examine the Exadata processes
- Exercise Exadata high availability
- Examine the hierarchy of cell objects
- Examine Exadata Smart Flash Cache



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

4

Key Capabilities of Exadata Database Machine

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

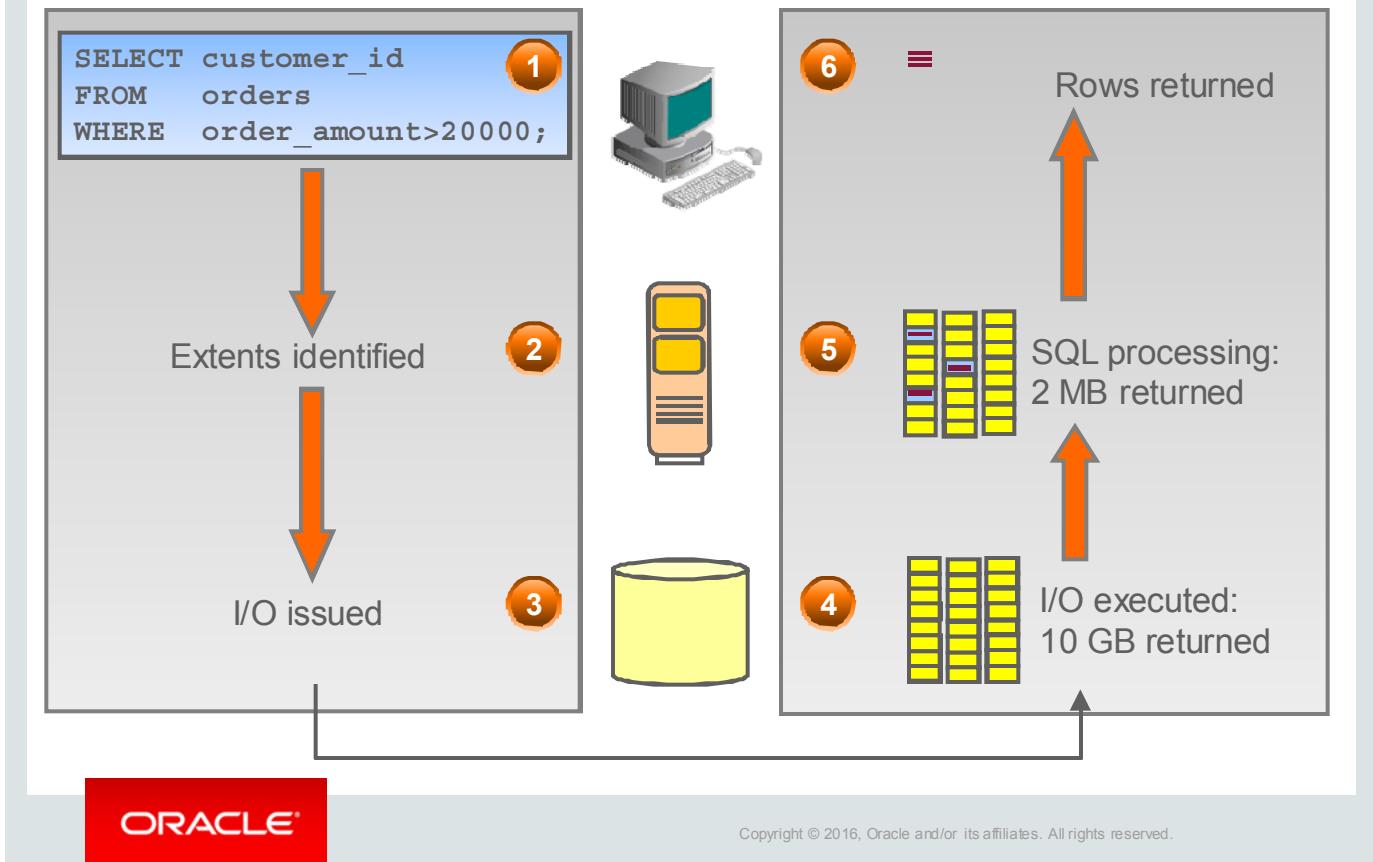
After completing this lesson, you should be able to describe the key features of Exadata Database Machine.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Classic Database I/O and SQL Processing Model

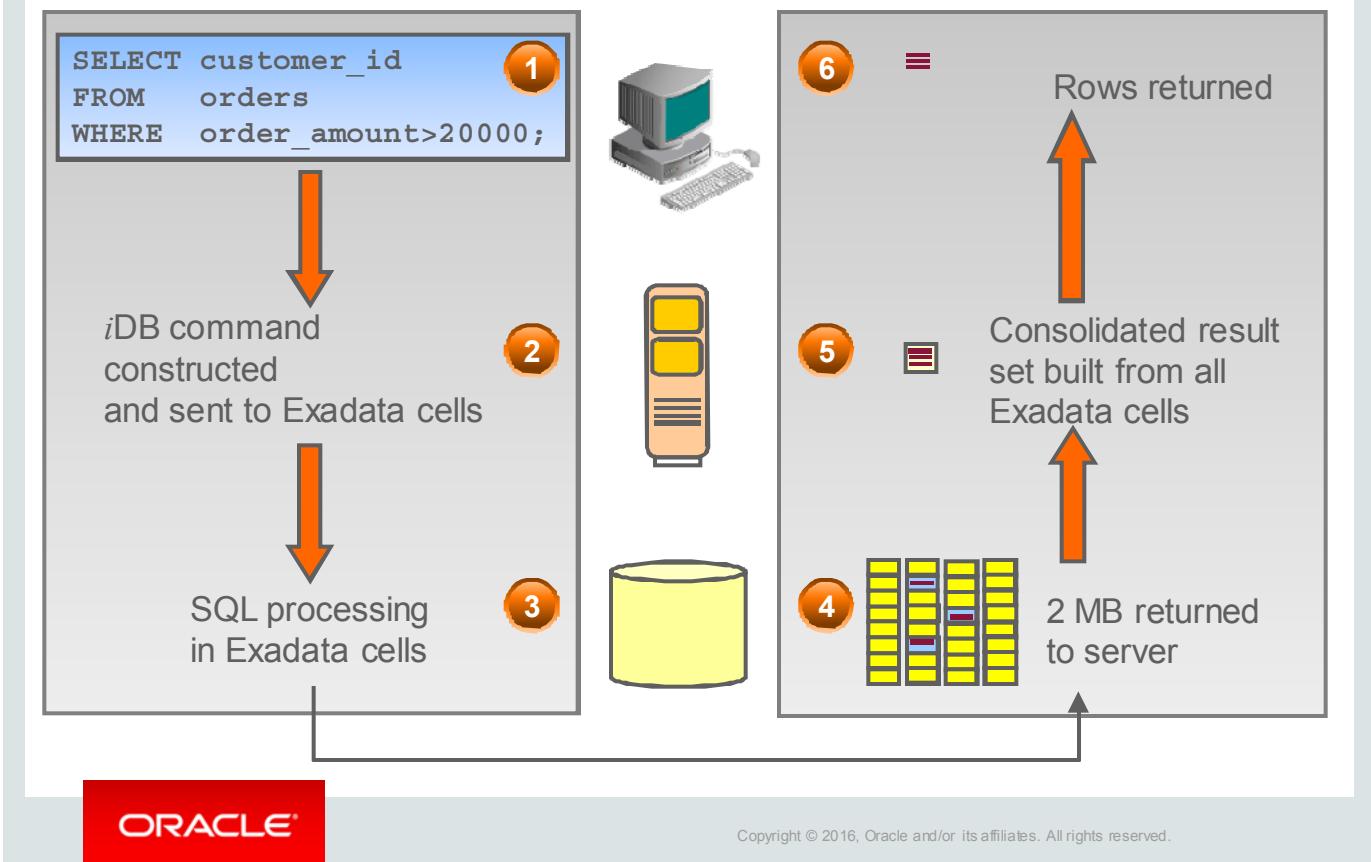


With traditional storage, all the database intelligence resides in the software on the database server. To illustrate how SQL processing is performed in this architecture, an example of a table scan is shown in the graphic in the slide.

1. The client issues a `SELECT` statement with a predicate to filter a table and return only the rows of interest to the user.
2. The database kernel maps this request to the file and extents containing the table.
3. The database kernel issues the I/Os to read all the table blocks.
4. All the blocks for the table being queried are read into memory.
5. SQL processing is conducted against the data blocks searching for the rows that satisfy the predicate.
6. The required rows are returned to the client.

As is often the case with the large queries, the predicate filters out most of the rows in the table. Yet all the blocks from the table need to be read, transferred across the storage network, and copied into memory. Many more rows are read into memory than required to complete the requested SQL operation. This generates a large amount of unproductive I/O, which wastefully consumes resources and impacts application throughput and response time.

Exadata Smart Scan Model



On Exadata, database operations are handled differently. Queries that perform table scans can be processed within Exadata cells and return only the required subset of data to the database server. Row filtering, column filtering, some join processing, and other functions can be performed within Exadata cells. Exadata Storage Server uses a special direct-read mechanism for Smart Scan processing. The above graphic illustrates how a table scan operates with Exadata cell storage:

1. The client issues a `SELECT` statement to return some rows of interest.
2. The database kernel determines that the data is stored on Exadata cells so an *iDB* command representing the `SQL` command is constructed and sent to the Exadata cells.
3. The Exadata Storage Server software scans the data blocks to extract the relevant rows and columns which satisfy the `SQL` command.
4. Exadata cells return to the database instance *iDB* messages containing the requested rows and columns of data. These results are not block images, so they are not stored in the buffer cache.
5. The database kernel consolidates the result sets from across all the Exadata cells. This is similar to how the results from a parallel query operation are consolidated.
6. The rows are returned to the client.

Moving SQL processing off the database server frees server CPU cycles and eliminates a massive amount of unproductive I/O transfers. These resources are free to better service other requests. Queries run faster, and more of them can be processed.

Exadata Smart Storage Capabilities

- Predicate filtering:
 - Only the requested rows are returned to the database server rather than all the rows in a table.
- Column filtering:
 - Only the requested columns are returned to the database server rather than all the columns in a table.
 - Example:

```
SQL> SELECT name FROM employees WHERE LENGTH(name) > 5;
```

- With predicate and column filtering, only the employee names that are longer than five characters are sent to the database servers.
- Without filtering, the entire `employees` table must be sent from storage to the database server.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The following database functions are integrated within Exadata Storage Server:

- Exadata Storage Server enables predicate filtering for table scans. Rather than returning all the rows for the database to evaluate, Exadata Storage Server returns only the rows that match the filter condition. The conditional operators that are supported include =, !=, <, >, <=, >=, IS [NOT] NULL, LIKE, [NOT] BETWEEN, [NOT] IN, EXISTS, IS OF type, NOT, AND, and OR. In addition, many common SQL functions can be evaluated by Exadata Storage Server during predicate filtering. For a full list of functions that can be evaluated by Exadata cell, use the following query:

```
SELECT * FROM v$sqlfn_metadata WHERE offloadable = 'YES';
```

- Exadata Storage Server provides column filtering, also called column projection, for table scans. Only the requested columns are returned to the database server rather than all columns in a table. For tables with many columns, or columns containing LOBs, the I/O bandwidth saved by column filtering can be very large.

When used together, the combination of predicate and column filtering dramatically improves performance and reduces I/O bandwidth consumption.

Exadata Smart Storage Capabilities

- Join processing:
 - Star join processing is performed within Exadata Storage Server.
- Scans on encrypted data
- Scans on compressed data
- Scoring for Data Mining
 - Example:

```
SELECT cust_id
  FROM customers
 WHERE region = 'US'
 AND prediction_probability(churnmod, 'Y' using *) > 0.8;
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

- Exadata Storage Server performs join processing for star schemas (between large tables and small lookup tables). This is implemented using Bloom Filters, which is a very efficient probabilistic method to determine whether an element is a member of a set.
- Exadata Storage Server performs Smart Scans on encrypted tablespaces and encrypted columns. For encrypted tablespaces, Exadata Storage Server can decrypt blocks and return the decrypted blocks to Oracle Database, or it can perform row and column filtering on encrypted data. Significant CPU savings can be made within the database server by offloading the CPU-intensive decryption task to Exadata cells.
- Smart Scan works in conjunction with Hybrid Columnar Compression so that column projection, row filtering and decompression can be executed at the storage level to save CPU cycles on the database servers.
- Exadata Storage Server can perform scoring functions, such as PREDICTION_PROBABILITY, for data mining models. This accelerates analysis while reducing database server CPU consumption and the I/O load between the database server and storage servers.

Exadata Smart Storage Capabilities

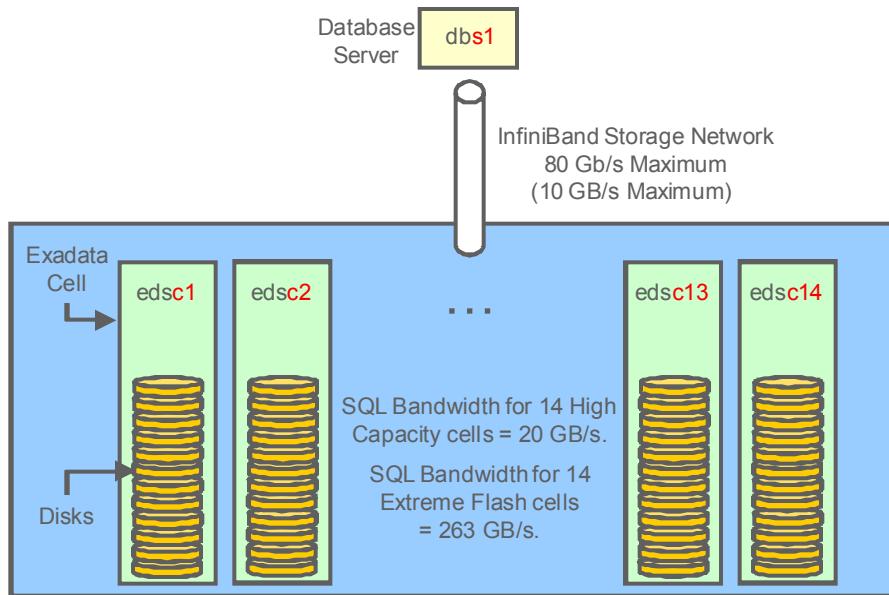
- Create/extend tablespace:
 - Exadata formats database blocks.
- Backup and recovery:
 - I/O for incremental backups is much more efficient because only changed blocks are returned to the database server.
 - Exadata performs RMAN file restoration.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

- With Exadata storage, the create/extend tablespace operation is executed more efficiently. Instead of formatting empty blocks in database server memory and writing them to storage, a single iDB command is sent to Exadata Storage Server instructing it to format the blocks. Database server memory usage is reduced and I/O associated with the creation and formatting of the database blocks is eliminated with Exadata storage.
- The speed and efficiency of incremental database backups is enhanced with Exadata Storage Server. With Exadata storage, changes are tracked at the individual Oracle block level rather than for a larger group of blocks. This results in less I/O bandwidth being consumed for backups and faster running backups.
- With Exadata storage, the RMAN file restore operation is offloaded to the cells. This is essentially the same as the optimization associated with creating or extending a tablespace (described earlier).

Exadata Smart Scan Scale-Out: Example



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The example in the next three slides illustrates the power of Smart Scan in a quantifiable manner using a typical case in which multiple Exadata cells scale-out to share a workload.

The database server, depicted in the upper portion of the slide, is connected to the InfiniBand storage network, which can deliver a maximum of 40 gigabits per second (Gb/s) on each network channel. Therefore, assuming that the server is configured with active bonding, the storage network bandwidth available to the server is 80 Gb/s (or 10 gigabytes per second [GB/s]). To keep the example clear and simple, assume that the InfiniBand storage network can deliver data at 10 GB/s. This is a generous assumption because it assumes that active bonding scales perfectly and that there is no messaging overhead in the network. We will also assume that a single database server has access to the full I/O bandwidth of all the Exadata cells.

In this scenario, there are 14 Exadata cells, the same as in a Full Rack Database Machine. Based on published specifications, the maximum SQL bandwidth for a Full Rack Database Machine using High Capacity cells is 25 GB/s. However, this jumps to 263 GB/s for a Full Rack Database Machine using Extreme Flash storage servers.

Exadata Smart Scan Scale-Out: Example

Without Smart Scan

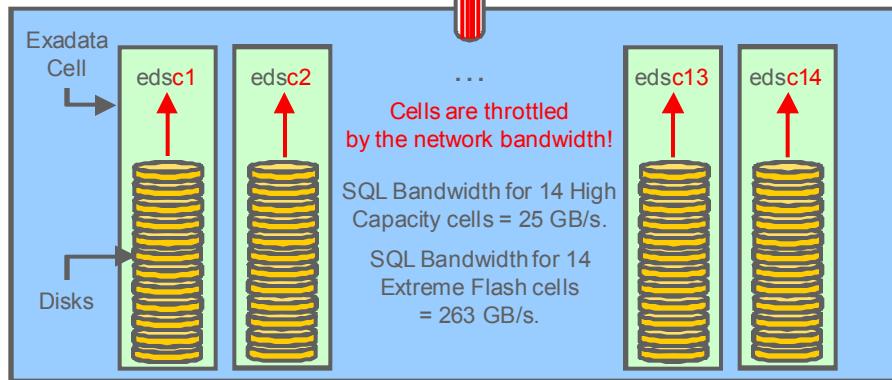
```
select /*+ full(lineitem) */ count(*)
from lineitem
where l_orderkey < 0;
```

Database Server
dbs1

Database asks to retrieve all blocks by doing a full table scan, and then filters matching rows.

If the table is 4800 GB in size, the complete scan would take approximately **8 minutes**.

InfiniBand Storage Network
80 Gb/s Maximum
(10 GB/s Maximum)



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Now assume a 4800 gigabyte table is evenly spread across the 14 Exadata cells and a query is executed that requires a full table scan. As is commonly the case, assume that the query returns a small set of result records.

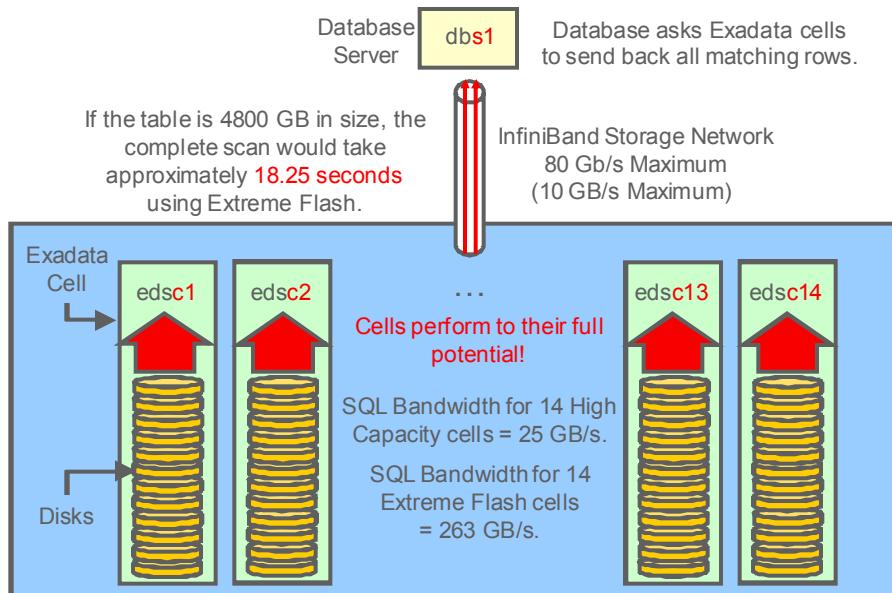
Without Smart Scan capabilities, each Exadata Storage Server behaves like a traditional storage server by delivering database blocks to the client database.

Because the storage network link to the database server is bandwidth-limited to 80 gigabits per second (that is 10 gigabytes per second), it is not possible for the Exadata cells to deliver all their power. In this case, even making generous assumptions about the power of the storage network, it would take approximately 8 minutes to scan the whole table.

Exadata Smart Scan Scale-Out: Example

With Smart Scan

```
select /*+ full(lineitem) */ count(*)
from lineitem
where l_orderkey < 0;
```



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Now consider if Smart Scan is applied to the same query. The same storage network bandwidth limits apply. However, this time, the entire 4800 GB is not transported across the storage network; only the matching rows are transported back to the database server. So each Exadata cell can process its part of the table at full speed. In this case, the entire table scan would be completed in approximately 3 minutes and 12 seconds on a Database Machine with High Capacity cells, and in as little as 18.25 seconds using Extreme Flash cells.

Hybrid Columnar Compression: Overview

Warehouse Compression

Optimized for Speed

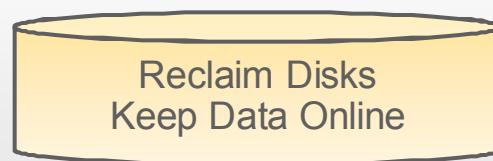
- 10x average storage savings
- 10x scan I/O reduction
- Optimized for query performance



Archival Compression

Optimized for Space

- 15x average storage savings
 - Up to 50x on some data
- Greater access overhead
- For cold or historical data



Can mix compression types by partition for Information Lifecycle Management

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In addition to the basic and OLTP compression capabilities of Oracle Database 11g, Exadata Storage Server includes Hybrid Columnar Compression.

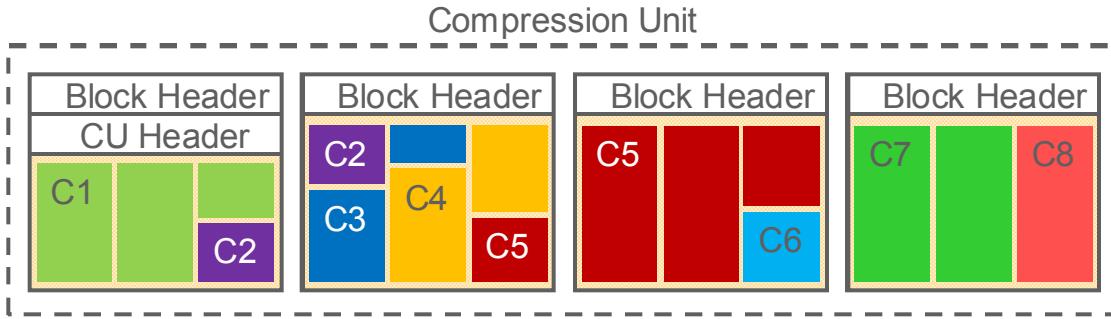
You can specify Hybrid Columnar Compression at the table, partition, and tablespace level. You can also choose between two types of Hybrid Columnar Compression, to achieve the proper trade-off between disk usage and CPU consumption, depending on your requirements:

- Warehouse compression is optimized for query performance, and is intended for data warehouse applications.
- Online archival compression optimized for maximum compression ratios, and is intended for data that rarely changes.

You can use Hybrid Columnar Compression on complete tables or in combination with basic and OLTP compression by using partitioning.

Note: A compression advisor, provided by the DBMS_COMPRESSION package, helps you determine the expected compression ratio for a particular table and compression method.

Hybrid Columnar Compression: Data Organization



- A compression unit is a logical structure spanning multiple database blocks.
- Each row is self-contained within a compression unit.
- Data is organized by column during data load.
- Each column is compressed separately.
- Smart Scan is supported.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Using Hybrid Columnar Compression, data is organized into sets of rows called compression units. Within a compression unit, data is organized by column and then compressed. The column organization of data brings similar values close together, enhancing compression ratios. Each row is self-contained within a compression unit.

The size of a compression unit is determined automatically by Oracle Database based on various factors in order to deliver the most effective compression result while maintaining excellent query performance. Although the diagram in the slide shows a compression unit with four data blocks, do not assume that a compression unit always contains four blocks.

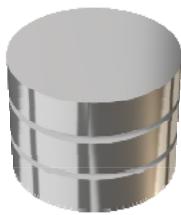
In addition to providing excellent compression, Hybrid Columnar Compression works in conjunction with Smart Scan so that column projection and row filtering can be executed along with decompression at the storage level to save CPU cycles on the database servers.

While aiding compression, columnar organization is not ideally suited to DML operations because a seemingly insignificant change to one row may force an entire compression unit to be reorganized. Because of the high cost of reorganizing columnar organized data, such operations are avoided by Hybrid Columnar Compression. Rather, deletes are logical in nature, whereas updates and conventional path inserts result in new rows being inserted into single-block compression units. Hence, Hybrid Columnar Compression provides optimal compression ratios for direct path loaded data and is recommended for data that is not updated frequently.

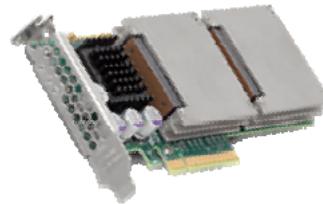
Exadata Smart Flash Cache: Overview

- High-performance cache for frequently accessed objects
- Write-through and write-back modes available
 - Write-through mode is excellent for absorbing repeated random reads.
 - Write-back mode is best for write intensive workloads.
- Allows optimization by application table

Hundreds of I/Os per Sec



Tens of thousands of I/Os per second



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

For many years, a constraining factor for storage performance has been the number of random I/Os per second (IOPS) that a disk can deliver. To compensate for the fact that even a high-performance disk can deliver only a few hundred IOPS, large storage arrays with hundreds of disks are required to deliver in excess of 100,000 IOPS.

Exadata Storage Server provides Exadata Smart Flash Cache, a caching mechanism for frequently accessed data. Using Exadata Smart Flash Cache, an Exadata cell can support approximately 400,000 read IOPS, two cells can support approximately 800,000 read IOPS, and so on.

Commencing with Exadata Storage Server release 11.2.3.2.0, Exadata Smart Flash Cache can operate as either a write-through cache or a write-back cache. Write-through mode is best suited to the random repeated reads commonly found in OLTP applications. Write-back mode is best suited to write-intensive applications. Both modes of operation are covered in greater detail later in the lesson.

Exadata Smart Flash Cache focuses on caching frequently accessed data and index blocks, along with performance critical information such as control files and file headers. In addition, DBAs can influence caching priorities by using the `CELL_FLASH_CACHE` storage attribute for specific database objects.

Exadata Smart Flash Cache Intelligent Caching: Overview

Exadata Smart Flash Cache understands different types of database I/O:

- Frequently accessed data and index blocks are cached.
 - Control file reads and writes are cached.
 - File header reads and writes are cached.
 - DBA can influence caching priorities.
-
- Backup-related I/O is not cached.
 - Data Pump I/O is not cached.
 - Table scans do not thrash the cache.
 - I/Os to mirror copies are managed intelligently.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In more recent times, nonvolatile memory caches have become commonplace in storage arrays to improve performance. However, these caches know nothing about the applications using them, so their efficiency is limited.

With Exadata storage, each database I/O is tagged with metadata indicating the I/O type. Exadata Smart Flash Cache uses this information to make intelligent decisions about how to use the cache. This cooperation ensures the efficient use of Exadata Smart Flash Cache.

For example, with traditional storage arrays, backups and exports typically cause all the data to be loaded into the cache even though the operation does not read the data repeatedly. Exadata Storage Server knows that there is no need to fill the cache with backup and export data.

Exadata Storage Server carefully manages caching in association with full table scans to avoid cache thrashing, a condition which renders the cache ineffective because it is constantly being overwritten.

Also, depending on the caching mode used, there may be no need to cache mirror copies of data.

Exadata Smart Flash Cache

Intelligent Caching Details

- Caching is automatically governed by:
 - A cache hint, based on the reason for the I/O:
 - CACHE indicates that the I/O should be cached.
 - NOCACHE indicates that the I/O should not be cached.
 - EVICT calls for data to be removed from the cache.
 - Object size and frequency of access, versus the frequency of access for other cached objects.
 - The most useful data is prioritized and the least useful data is evicted.
- Automatic caching priorities can be manually adjusted by setting the `CELL_FLASH_CACHE` storage attribute.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Smart Flash Cache automatically works in conjunction with Oracle Database to intelligently optimize the efficiency of the cache. Each database I/O is tagged with a cache hint, which is assigned by the database based on the reason for the I/O:

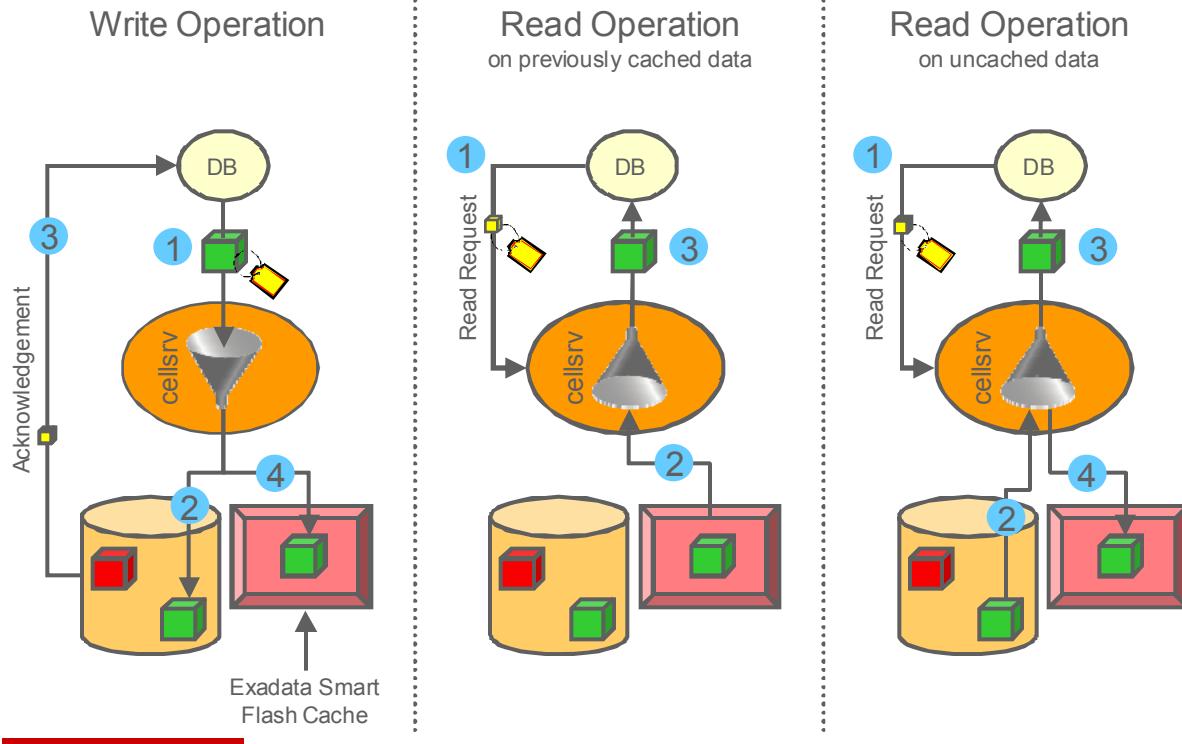
- CACHE indicates that the I/O should be cached. For example, the I/O is for an index lookup.
- NOCACHE indicates that the I/O should not be cached. For example, the I/O is for a mirrored block of data or is associated with a backup.
- EVICT indicates that data should be removed from the cache. For example, when an ASM rebalance operation moves data between different disks, the cached copies that correspond to the original location are removed from the cache.

Exadata Smart Flash Cache also uses internal statistics and other measures to determine whether or not an object (table or index) should be cached. It takes into account the size of the object and the frequency of access to the object, and weighs that against the frequency of access to other cached objects to determine which objects to prioritize. Objects can be fully or partially cached, depending on the object size, flash cache size, and the other concurrent workloads. This behaviour is fully automatic and designed to intelligently maximize the benefits associated with Exadata Smart Flash Cache.

In addition to the automatic caching algorithms, administrators can manually influence caching priorities by setting the `CELL_FLASH_CACHE` storage attribute for specific tables and indexes. The options are:

- `DEFAULT` specifies that the object is cached normally.
- `KEEP` specifies that the object has a higher than default caching priority, making it more likely that the object will be cached.
- `NONE` ensures that the object is never cached in Exadata Smart Flash Cache. This allows flash cache space to be reserved for other objects.

Using Exadata Smart Flash Cache: Write-Through Cache



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In write-through mode, Exadata Smart Flash Cache works as follows:

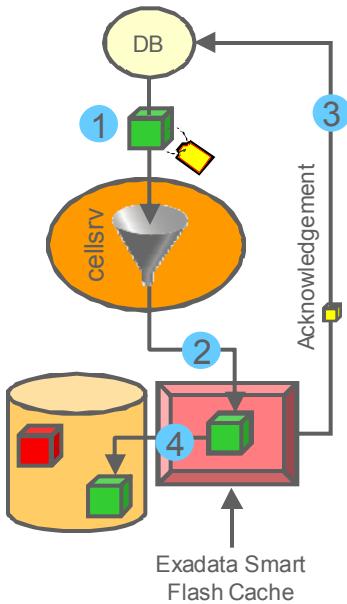
- For write operations, CELLSRV writes data to disk and sends an acknowledgement to the database so it can continue without any interruption. Then, if the data is suitable for caching, it is written to Exadata Smart Flash Cache. Write performance is not improved or diminished using this method. However, if a subsequent read operation needs the same data, it is likely to benefit from the cache. When data is inserted into a full cache, a prioritized least recently used (LRU) algorithm determines which data to replace.
- For read operations, CELLSRV must first determine if the request should use the cache. This decision is based on various factors including the reason for the read, the CELL_FLASH_CACHE setting for the associated object, and the current load on the cell. If it is determined that the cache should be used, CELLSRV uses an in-memory hash table, to quickly determine if the data resides in Exadata Smart Flash Cache. If the requested data is cached, a cache lookup is used to satisfy the I/O request.
- For read operations that cannot be satisfied using Exadata Smart Flash Cache, a disk read is performed and the requested information is sent to the database. Then if the data is suitable for caching, it is written to Exadata Smart Flash Cache.

Note that in write-through mode, multiple copies of each data block are written to disk to maintain ASM redundancy. As a result, there is usually no need to cache the secondary block copy because ASM will read the primary copy if it is available.

By default, Smart Flash Cache operates in write-through mode on high capacity storage servers.

Using Exadata Smart Flash Cache: Write-Back Cache

Write Operation



- How it works:
 - Suitable writes go to flash only.
 - Data is automatically written to disk as it ages out of the cache.
 - Active data blocks can reside in flash indefinitely.
 - Reads are handled the same way as in write-through mode.
- Characteristics:
 - Ideal for write-intensive applications.
 - For many applications, most I/O is serviced by flash.
 - If a problem is detected, I/O operations transparently fail over to mirrored copies of data also on flash.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing with Exadata Storage Server release 11.2.3.2.0, Exadata Smart Flash Cache can operate in write-back mode. In this mode, write operations work as follows:

1. CELLSRV receives the write operation and uses intelligent caching algorithms to determine if the data is suitable for caching.
2. If the data is suitable for caching, it is written to Exadata Smart Flash Cache only. If the cache is full, CELLSRV determines which data to replace using the same prioritized least recently used (LRU) algorithm as in write-through mode.
3. After the data is written to flash, an acknowledgment is sent back to the database.
4. Data is only written back to disk when it is aged out of the cache.

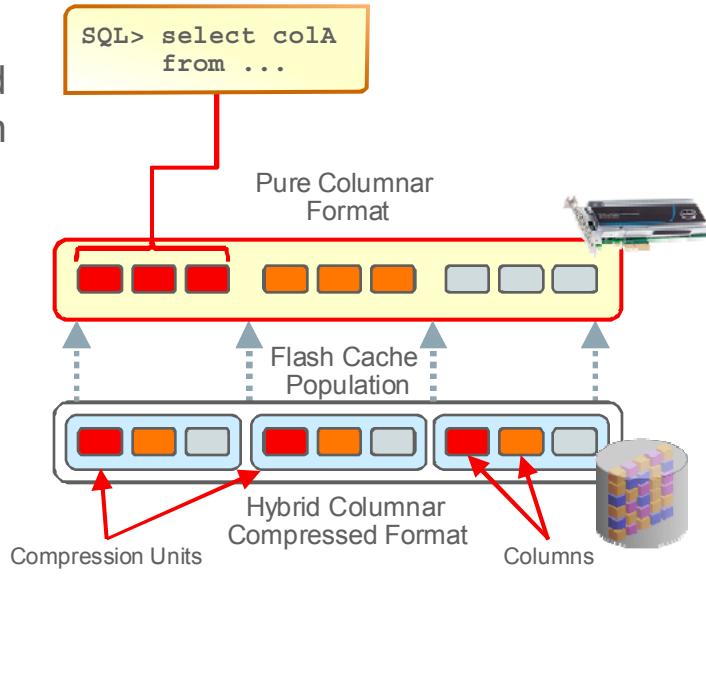
Note the following regarding write-back flash cache:

- Write-back flash cache is ideal for write-intensive applications that would otherwise saturate the disk controller write cache.
- The large flash capacity of Exadata Storage Server means that for many applications a high proportion of all I/O can be serviced by flash.
- An active data block can remain in write back flash cache for months or years. Also, Exadata Smart Flash Cache is persistent through power outages, shutdown operations, cell restarts, and so on.

- With write-back flash cache, data redundancy is maintained by writing primary and secondary data copies to cache on separate storage servers.
- Secondary block copies are aged out of the cache (and written to disk) more quickly than primary copies. Hence, blocks that have not been used recently only keep the primary copy in cache, which optimizes the utilization of flash space.
- If there is a problem with the flash cache on one storage server, then operations transparently fail over to the mirrored copies (on flash or disk) on other storage servers. No user intervention is required. The unit for mirroring is the ASM allocation unit. This means that the amount of data affected is proportional to the lost cache size, not the disk size.
- With write-back flash cache, read operations are handled the same as with the write-through flash cache.

Columnar Flash Caching

- Frequently scanned hybrid columnar compressed data is automatically transformed into pure columnar format in Smart Flash Cache.
- The same data may be cached twice in mixed workload situations.
- Analytics runs faster while maintaining excellent single row lookup performance.
- Enabled by default and no user configuration



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing with Exadata release 12.1.2.1.0, Smart Flash Cache can automatically transform frequently scanned hybrid columnar compressed (HCC) data into pure columnar format during flash cache population. Smart scans operating on pure columnar data in flash run faster because they read only the selected columns, which reduces the required number of flash I/Os and amount of storage server CPU that is consumed. In particular, this benefits analytic queries that operate on a small proportion of table columns.

Depending on the nature of the system workload the same region of data can be cached in both the hybrid columnar compressed format and the columnar format. As a result, columnar flash caching accelerates reporting and analytic queries while maintaining excellent performance for OLTP style single row lookups.

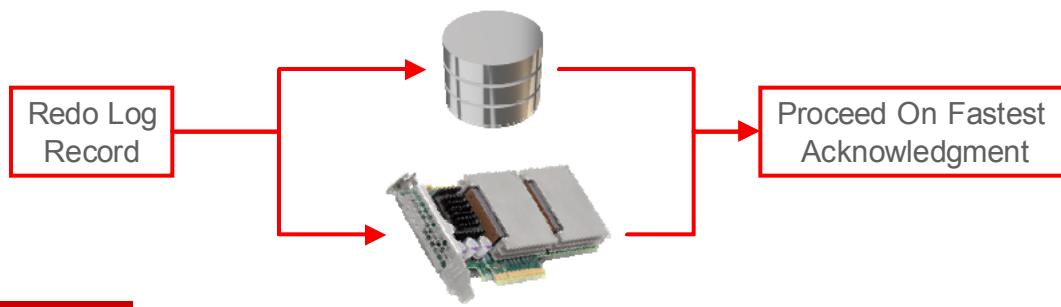
Columnar flash caching is enabled by default, and does not need configuration by the user.

Exadata cell software release 12.1.2.1.0 and Oracle Database 12c release 12.1.0.2.0 are the minimum software requirements for columnar flash caching.

Exadata Smart Flash Log: Overview

Exadata Smart Flash Log provides a high-performance, low-latency, reliable temporary store for redo log writes:

- Log writes are directed to disk and Exadata Smart Flash Log.
- Processing continues after fastest acknowledgment.
- Conceptually similar to multiplexed redo logs.
- Exadata Storage Server automatically manages Smart Flash Log and ensures all log entries are persisted to disk.



ORACLE®

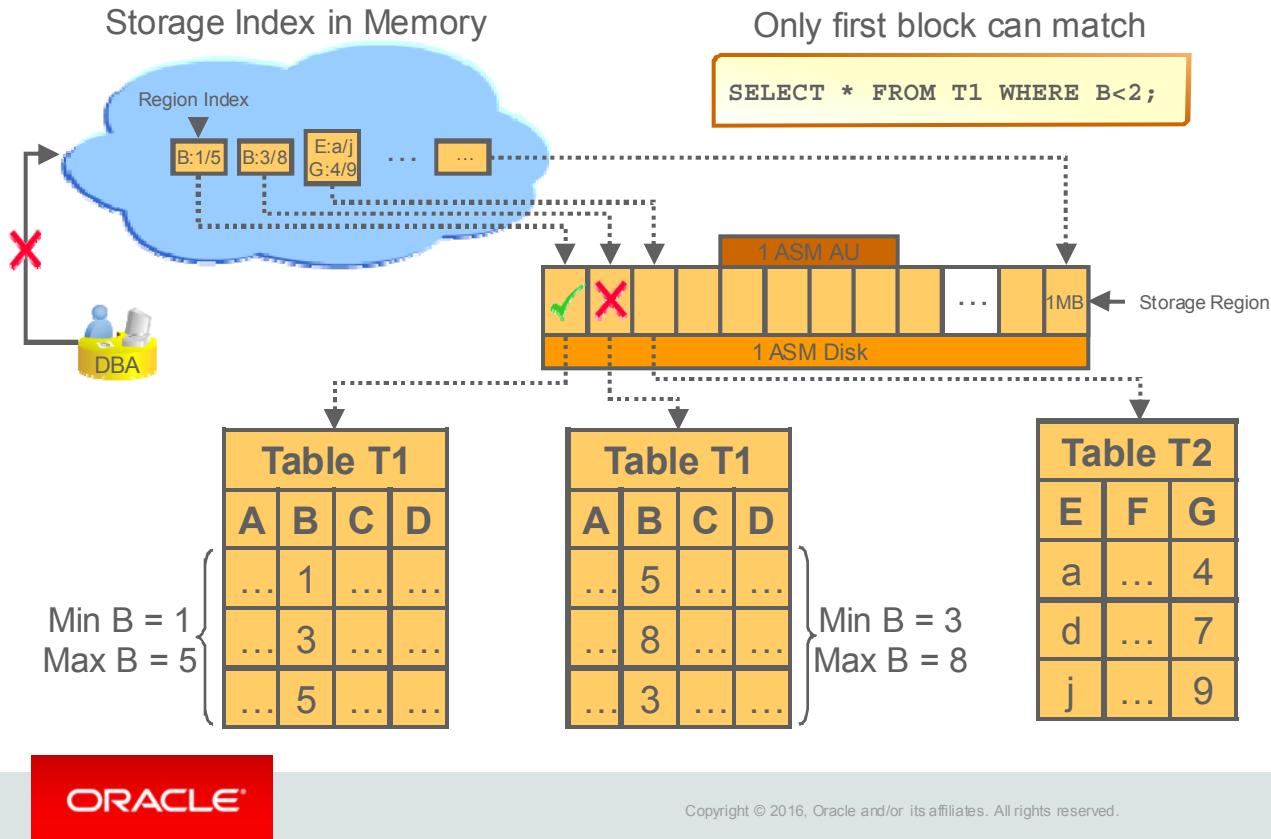
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Smart Flash Log provides a mechanism for improving the latency of database redo log write operations. Exadata Smart Flash Log uses a small portion of the high-performance flash memory on Exadata Storage Server as temporary storage to provide low latency redo log writes.

Using Exadata Smart Flash Log, log records are written simultaneously to the disk-based online redo log files and Exadata Smart Flash Log, and processing continues upon acknowledgment of the faster write operation, whether it is to hard disk or Exadata Smart Flash Log. When the hard disks containing the online redo log files experience temporary performance issues, such as heavy I/O loads, the write operations to Exadata Smart Flash Log will typically complete faster and provide consistently good response times.

Exadata Smart Flash Log is conceptually similar to multiplexed online redo log files, which have been available in Oracle Database for many years. The main difference is that Exadata Smart Flash Log uses flash memory only for temporary storage of redo log data. By default, Exadata Smart Flash Log uses 512 MB of flash storage on each Exadata Storage Server.

Exadata Storage Index: Overview



A storage index is a memory-based structure that reduces the amount of physical I/O performed in an Exadata cell. The storage index keeps track of minimum and maximum column values and this information is used to avoid useless I/Os.

For example, the slide shows table T1 which contains column B. Column B is tracked in the storage index so it is known that the first half of T1 contains values for column B ranging between 1 and 5. Likewise it is also known that the second half of T1 contains values for column B ranging between 3 and 8. Any query on T1 looking for values of B less than 2 can quickly proceed without any I/O against the second part of the table.

Given a favorable combination of data distribution and query predicates, a storage index can drastically speed up a query by quickly skipping much of the I/O. For another query, the storage index may provide little or no benefit. In any case, the ease of maintaining and querying the memory-based storage index means that any I/O saved through its use effectively increases the overall I/O bandwidth of the cell while consuming very few cell resources.

The storage space inside each cell disk is logically divided into 1 MB chunks called storage regions. The boundaries of ASM allocation units (AUs) are aligned with the boundaries of storage regions. For each of these storage regions, data distribution statistics are held in a memory structure called a region index. Each region index contains distribution information for up to eight columns. The storage index is a collection of the region indexes.

The storage statistics maintained in each region index represent the data distribution (minimum and maximum values) of columns that are considered well clustered. Exadata Storage Server contains logic that transparently determines which columns are clustered enough to be included in the region index. Different parts of the same table can potentially have different column sets in their corresponding region indexes.

The storage index works best when the following conditions are true:

- The data is roughly ordered so that the same column values are clustered together.
- The query has a predicate on a storage index column checking for =, <, >, or a combination of these.

It is important to note that the storage index works transparently with no user input. There is no need to create, drop, or tune the storage index. The only way to influence the storage index is to load your tables using presorted data.

Also, because the storage index is kept in memory, it disappears when the cell is rebooted. The first queries that run after a cell is rebooted automatically cause the storage index to be rebuilt.

The storage index works for data types whose binary encoding is such that byte-wise binary lexical comparison of two values of that data type is sufficient to determine the ordering of those two values. This includes data types like NUMBER, DATE, and VARCHAR2. However, NLS data types are an example of data types that are not included for storage index filtering.

Storage Index with Partitions: Example

ORDER#	ORDER_DATE (Partition Key)	SHIP_DATE	ITEM
1	2007	2007	
2	2008	2008	
3	2009	2009	

- Queries on SHIP_DATE do not benefit from ORDER_DATE partitioning:
 - However SHIP_DATE is highly correlated with ORDER_DATE.
- Storage index enhances performance for queries on SHIP_DATE:
 - Takes advantage of the ordering created by partitioning.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

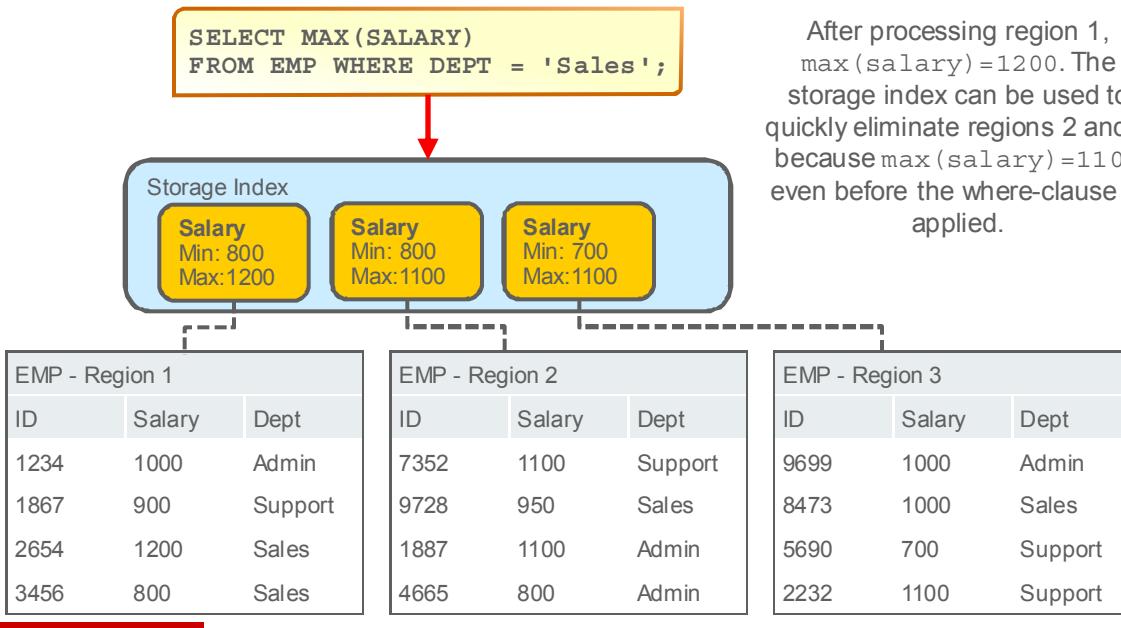
The example in the slide contains correlated columns. ORDER_DATE is highly correlated with SHIP_DATE. The dates are generally correlated because usually a ship date is close to an order date.

If your table is partitioned by ORDER_DATE, and you execute a query using ORDER_DATE as a filter, then partition pruning is used to read only the relevant partitions. However, if you do a query using only SHIP_DATE in the WHERE clause, partition pruning cannot be used to optimize the query.

However, if SHIP_DATE is part of the storage index, the storage index is used to skip all the blocks that do not correspond to your query. This filtering takes place at the storage level. The storage index helps the SHIP_DATE query to take advantage of the natural ordering implied by the ORDER_DATE partitioning and the natural correlation that exists between the ORDER_DATE and SHIP_DATE columns.

Performance Optimization for SQL Queries with Minimum or Maximum Functions

The Exadata storage index can be used to optimize queries that contain MIN and MAX functions:



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing in Exadata version 12.1.2.1.0, the storage index can also be used to satisfy queries that contain MIN and MAX functions. As the query is processed, running minimum and maximum values are tracked. Before issuing an I/O, the minimum and maximum values cached in the storage index for the data region are checked against the running values to decide whether or not the I/O is required.

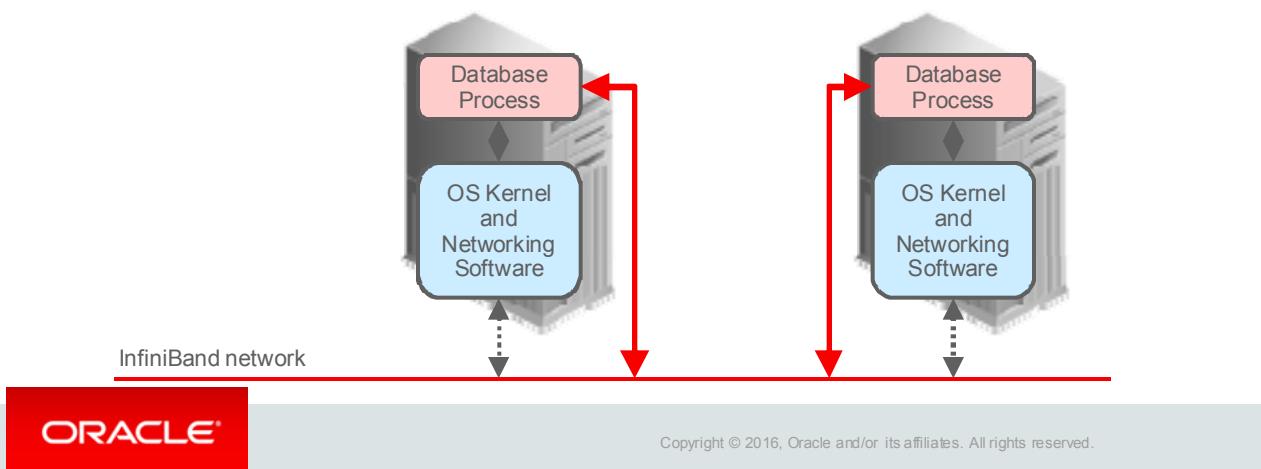
Overall, this optimization can result in significant I/O pruning during the course of a query, which improves query performance. The slide shows an example that benefits from this optimization.

Business intelligence tools that get the shape of a table by querying the minimum or maximum value for each column benefit greatly from this optimization.

Note that Oracle Database release 12.1.0.2 is the minimum required database software version.

Exafusion Direct-to-Wire Protocol

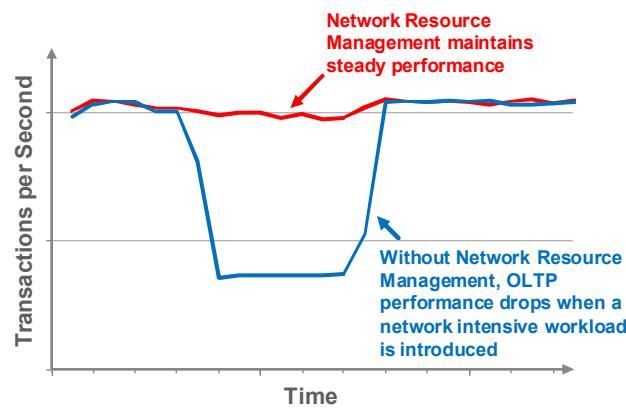
- Allows database processes to send and receive Oracle RAC messages directly over the InfiniBand network:
 - Bypasses the OS kernel and some networking layers
 - Reduces messaging latency and resource usage
 - Improves response times and transaction rates
 - Especially useful for OLTP environments



Exafusion Direct-to-Wire protocol allows database processes to read and send Oracle Real Applications Cluster (RAC) messages directly over the InfiniBand network, bypassing the overhead of entering the OS kernel, and running through the normal networking software stack. This improves the response time and scalability of the Oracle RAC environment on Exadata. Exafusion is especially useful for OLTP applications since messaging overhead is particularly apparent in such application, which are typically characterized by a very large number of small messages.

Exadata Network Resource Management

- Automatically and transparently prioritizes latency-sensitive messages:
 - Log writes:
 - Compliments Exadata Smart Flash log
 - Avoids issues caused when log writes are queued behind numerous large writes
 - Cache fusion:
 - Ensures consistent high performance for Oracle RAC



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Starting with Exadata release 11.2.3.3.0, Exadata Network Resource Management automatically and transparently prioritizes critical database network messages through the InfiniBand fabric ensuring fast response times for latency-critical operations. It leverages an InfiniBand quality of service mechanism that allows clients to prioritize messages based on how latency-sensitive they are.

Exadata Network Resource Management gives highest priority to log writes to ensure low latency for transaction processing. This capability complements Exadata Smart Flash Log, which also aims to minimize log write latencies.

Smart Flash Log mitigates high latency outliers that are seen when the disk controller cache is flushed to disk. However, even with Smart Flash Log, outliers can still be seen if the log buffer writes get queued behind a series of large (1 MB) transfers. This can affect OLTP performance, as illustrated in the performance chart example shown in the slide.

By using Exadata Network Resource Management, the issue is avoided because the log buffer writes go to the head of the queue and are processed before the large data transfers.

Exadata Network Resource Management also prioritizes Oracle RAC Cache Fusion messages over batch, reporting, and backup messages. This ensures consistent high performance for Oracle RAC users.

Exadata Network Resource Management is enabled by default, and requires no configuration or management.

Snapshot Databases for Test and Development

- Exadata snapshot databases for test and development:
 - Are based on a read-only copy of an existing database
 - Changes are written to a sparse disk group
- Creating sparse grid disks and a sparse disk group:

```
CellCLI> create griddisk all harddisk prefix=SPARSE, size=100G,  
virtualsize=1000G
```

```
SQL> create diskgroup SPARSE high redundancy  
disk 'o/*/*SPARSE_*' attribute  
'compatible.asm'='12.1.0.2', 'compatible.rdbms'='12.1.0.2',  
'appliance.mode' = 'TRUE', 'cell.smart_scan_capable'='true',  
'cell.sparse_dg'='allsparse', 'au_size'='4M';
```

- Creating a snapshot database based on a pluggable database:

```
SQL> CREATE PLUGGABLE DATABASE <snapshot name>  
FROM <source pluggable database>  
CREATE_FILE_DEST='+SPARSE'  
SNAPSHOT COPY;
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Using Exadata snapshot databases, space-efficient database snapshots can be quickly provisioned for test and development purposes. Snapshots databases support all Exadata features, such as smart scan. Snapshots start with a shared read-only copy of a database. As changes are made, each snapshot writes changed blocks to a sparse disk group.

A sparse disk group is composed of sparse grid disks. A sparse grid disk has a virtual size attribute as well as physical size. You must create sparse grid disks before you can create a sparse disk group. The required attribute settings are highlighted in the command examples in the slide.

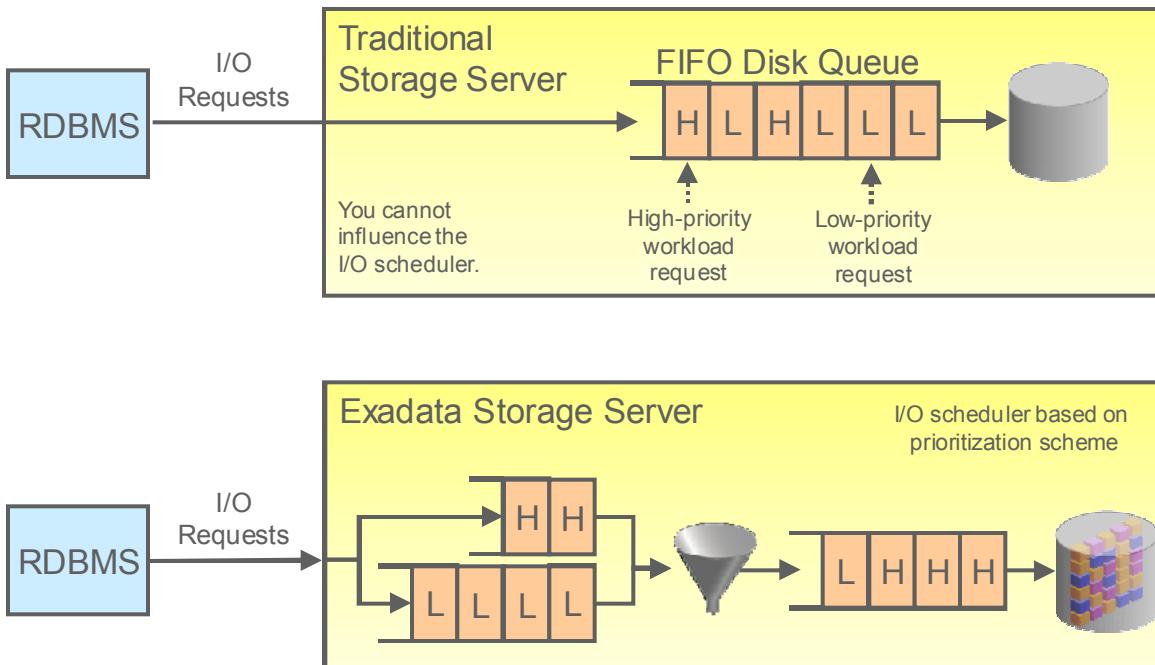
Multiple users can create independent snapshots from the same base database. Therefore multiple test and development environments can share space while maintaining independent databases for each task.

Exadata database snapshots are integrated with the Multi-tenant Database Option to provide an extremely simple interface for creating new PDB snapshots. The process for creating a snapshot database based on a PDB is reasonably simple, with most of the work performed through the CREATE PLUGGABLE DATABASE ... SNAPSHOT COPY command. The command example in the slide shows an outline of the required command.

Creating snapshot databases based on a non-PDB requires a more lengthy manual process.

Both procedures are documented in the *Oracle Exadata Storage Server Software User's Guide*.

I/O Resource Management: Overview



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

With traditional shared storage, balancing the work of multiple databases sharing the storage subsystem is inherently difficult. This issue is illustrated by the graphic at the top of the slide, which shows how traditional storage servers handle I/O requests. In essence, they queue I/O requests in a first-in, first-out (FIFO) order, which makes no distinction between high-priority and low-priority requests.

Exadata Storage Server enables allocation of I/O resources based on user-specified priorities and policies. This is illustrated in the graphic at the bottom of the slide where the Exadata Storage Server I/O scheduler executes I/O requests based on a prioritization scheme. It does that by internally queuing I/O requests to prevent a low-priority but intensive workload from flooding the disks.

I/O resource management is covered in more detail in the lesson titled “I/O Resource Management.”

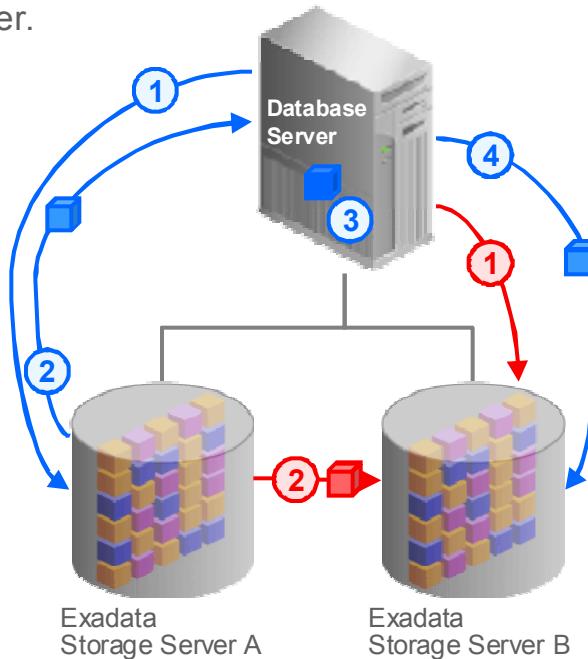
Cell-to-Cell Data Transfer

Cell-to-Cell transfer in 11.2.X:

1. The database server sends a read request to Cell A.
2. Cell A sends data to the database server.
3. Data is stored in the database server memory.
4. The database server sends data to Cell B.

Cell-to-Cell transfer in 12.1.X:

1. The database server sends a transfer request to Cell B.
 2. Cell B reads data from Cell A.
- Lower network bandwidth consumption
 - Lower database server resource usage
 - Used by ASM resynchronization, resilver, and rebalance operations



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In earlier releases, Exadata cells did not directly communicate with each other. Any data movement between the cells was done through the database servers. Data was read from the source cell into the database server memory, and then written out to the destination cell.

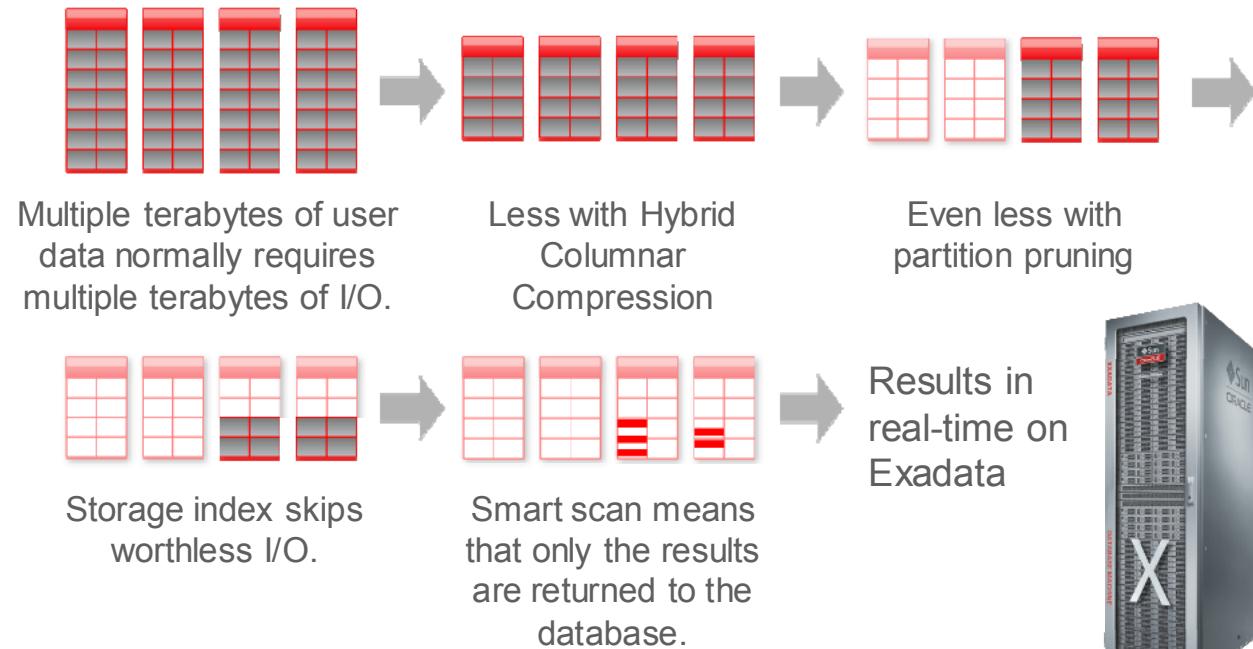
Starting with Exadata Storage Server version 12.1.1.1.0, database server processes can offload data transfer. That is, a database server can instruct the destination cell to read the data directly from the source cell. This halves the amount of data transferred across the fabric, reducing database server InfiniBand bandwidth consumption and memory requirements.

Oracle Automatic Storage Management (Oracle ASM) resynchronization, resilver, and rebalance operations use this feature to offload data movement. This provides improved bandwidth utilization at the InfiniBand fabric level and improved memory utilization in the Oracle ASM instances.

The minimum software requirement for this feature is Oracle Database 12c Release 1 (12.1) or later, and Exadata Storage Server version 12.1.1.1.0 or later. No additional configuration is needed to use this feature.

Commencing with Exadata release 12.1.2.3.0, storage index entries can be rebalanced and moved along with the data during a cell-to-cell offloaded rebalance. This optimization provides significant performance improvement compared to earlier releases in application performance during a rebalance due to disk failure. The minimum software required for this optimization is Exadata Storage Server version 12.1.2.3.0 in conjunction Oracle Grid Infrastructure release 12.1.0.2.160119 with patch 22682752.

Multiplied Benefits



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This is an example that shows you how the features that were introduced in this lesson can work together to multiply the benefits of Exadata Database Machine.

Assume you have a multi-terabyte table and somebody runs a query that is interested in a small subset of the data, but causes a full table scan. Traditionally, the system would have to scan the terabytes of data.

However, using Hybrid Columnar Compression could reduce the size of the table.

If the table is partitioned, the database optimizer could use partition pruning to eliminate a substantial proportion of the data.

Using storage indexes, Exadata Storage Server might further reduce the amount of physical I/O that is required.

Finally, because of Smart Scan, the only data returned to the database is the data of interest to the query, some of which may have been cached inside Exadata Smart Flash Cache.

This example shows how many features can work in harmony to improve the performance of a single operation by using Exadata Database Machine.

Exadata Benefits for Data Warehousing and Analytics

- Modular storage cell building blocks are organized into a parallel storage grid providing large I/O throughput.
- The InfiniBand storage network is much faster than traditional SAN storage networks, which helps to deliver the potential of the storage grid.
- Query processing is moved into storage to dramatically reduce data sent to servers while unloading server CPUs.
- Hybrid Columnar Compression reduces the number of physical I/Os for large table scans.
- In-memory parallel query provides a powerful alternative query strategy that complements Exadata Storage Server.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

One of the key benefits of Exadata is extreme performance for data warehousing and analytics. Most of the performance gains are derived through the architecture and unique features associated with the Exadata Storage Server grid inside Exadata Database Machine.

The Exadata Storage Server grid addresses three key dimensions of database I/O that can affect analytics performance:

- Exadata Storage Server inherently supports a parallel architecture, which provides numerous connections to deliver more data faster between the storage servers and the database servers. Each Exadata Database Machine rack contains between 3 and 18 Exadata cells, each with 12 hard disk drives or 8 flash disk drives, which can deliver the large I/O throughput required to satisfy large queries.
- The Exadata storage network is built using wide network pipes that provide extremely high bandwidth between the storage servers and the database servers. Exadata uses InfiniBand as the storage network, which provides a throughput of 40 Gb/sec with very low latency. This is many times the bandwidth provided by traditional SAN storage networks.
- Exadata Storage Server is database-aware and can transport just the data required to satisfy SQL requests resulting in much less data being sent between the storage servers and the database servers.

In essence, the Exadata Storage Server grid reduces the volume of data transported and moves data faster compared with other systems based on traditional SAN storage.

Exadata Storage Server provides additional capabilities that can further enhance performance.

Hybrid Columnar Compression provides very high levels of data compression implemented inside Exadata Storage Server. Hybrid Columnar Compression benefits large scale scans, commonly used in data warehousing and analytics, by efficiently scanning vast volumes of data using a fraction of I/Os. Compression ratios of 10 to 1 are common, which means that a 10 TB table can be scanned using 1 TB of disk I/O.

The tight integration between Oracle Database and Exadata Storage Server results in an intelligent platform for data warehousing and analytics. The complete solution uses a range of technologies to deliver the best result, not just relying on one approach to the problem. An example of this is the in-memory parallel query feature introduced in Oracle Database 11g Release 2.

Normally, a Smart Scan would be used to execute portions of a query inside Exadata Storage Server and return the minimum amount of data to the database server. In some cases, however, it may be more efficient to read all the required data into the memory on the database servers and process the query that way.

In-memory parallel query enhances query performance by minimizing or even completely eliminating additional physical I/O for a particular query. Oracle automatically decides if an object being accessed by using parallel execution benefits from being cached in the database buffer cache. The decision to cache an object is based on a well-defined set of heuristics including size of the object and the frequency that it is accessed.

In-memory parallel query harnesses the aggregated memory across a database cluster for parallel operations, enabling it to scale-out as the number of nodes in a cluster increases. In an Oracle RAC environment, Oracle maps fragments of the object into each of the buffer caches on the active instances. By creating this mapping, Oracle knows which buffer cache to access to find a specific part or partition of an object. Using this information, Oracle Database will prevent multiple instances from reading the same information from disk over and over again, thus maximizing the amount of memory that can be used to cache the objects.

In-memory parallel query nicely complements Exadata Storage Server. Using this combination, some queries can be efficiently executed with little or no additional I/O by pinning tables in the database buffer cache whereas others can harness the power of Smart Scan inside Exadata Storage Server.

Exadata Benefits for OLTP

- Modular storage cell building blocks are organized into a parallel storage grid providing large I/O throughput.
- The InfiniBand storage network is much faster than traditional SAN storage networks, which helps to deliver the potential of the storage grid.
- Powerful well-balanced database servers with large amounts of RAM help to support large user populations and large database buffer caches.
- Exadata Smart Flash Cache provides a high-performance secondary cache for frequently accessed objects, which is excellent for absorbing repeated random reads.

Hundreds of I/Os per sec



Tens of thousands of I/Os per second



- Exadata Smart Flash Logging keeps commit latency low.
- Exadata Network Resource Management automatically and transparently prioritizes latency-sensitive messages.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Some of the fundamental architectural characteristics of Exadata that are beneficial for analytics are equally relevant and beneficial for online transaction processing (OLTP). The high-performance, low-latency InfiniBand network that is used in conjunction with the massively parallel architecture of the Exadata storage grid is ideal for supporting many thousands of simultaneous users.

The Exadata database servers are designed to be powerful and well-balanced. They contain generous amounts of RAM, up to 6 TB per server in Exadata Database Machine X6-8, to support large numbers of client connections and large memory caches, which is beneficial to OLTP performance.

The introduction of Exadata Smart Flash Cache is of particular benefit to OLTP performance. Exadata Smart Flash Cache allows each Exadata cell to deliver approximately 300,000 IOPS. For the repeated random reads often associated with OLTP applications, Exadata Smart Flash Cache provides a secondary cache so that much faster lookups can be performed on data that is not in the database server caches. In addition, Oracle Database and Exadata Smart Flash Cache work closely with each other. This cooperation optimizes the usage of Exadata Smart Flash Cache so that only the most frequently accessed and performance-sensitive data is cached. Users have additional control over which database objects should be cached more aggressively than others, and which ones should not be cached at all.

Other features, such as Exadata Smart Flash Logging and Exadata Network Resource Management, benefit OLTP performance by ensuring consistently low processing latencies.

Quiz



When processing a query, which of the following Exadata features help to reduce the amount of data that is transported between the Exadata Storage Servers and the database servers?

- a.** Smart Scan
- b.** Smart Flash Cache
- c.** Smart Flash Log
- d.** Storage Index
- e.** I/O Resource Management

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a, d

Summary

In this lesson, you should have learned how to describe the key features of Exadata Database Machine.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 4 Overview: Introducing Exadata Features

In these practices, you are introduced to four major capabilities of Exadata, namely:

- Smart Scan
- Hybrid Columnar Compression
- Exadata Smart Flash Cache
- Storage Index



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

5

Exadata Database Machine Initial Configuration

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to describe:

- Installation and configuration process for Exadata
- Default configuration for Exadata
- Supported and unsupported customizations for Exadata



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Implementation: Overview

Four phases:

1. Pre-installation

- Various planning and scheduling activities including:
 - Site planning: space, power, cooling, logistics
 - Configuration planning: host names, IP addresses, databases
 - Network preparation: DNS, NTP, cabling
- Oracle and customer engineers can work together

2. Installation and configuration

- Hardware and software installation and configuration
- Result is a working system configured by using the desired configuration settings
- Recommended to be performed by Oracle engineers



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The process of successfully implementing Exadata requires cooperation and coordination between various Oracle and customer representatives and engineers. At a high level the process involves four main phases:

1. Pre-installation involves various planning and scheduling activities including those listed in the slide. Oracle can assist the customer through this phase by collaborating with them to complete the Oracle Exadata Deployment Assistant configuration tool (Exadata configuration tool). The information captured in the Exadata configuration tool is used to create a set of configuration files which drive the Oracle Exadata Deployment Assistant deployment tool (Exadata deployment tool) in phase two.
2. Installation and configuration is the process of taking a virgin Database Machine and installing it into the customer environment. It involves numerous steps conducted by hardware and software engineers. It is highly recommended that Oracle hardware and software engineers are used to conduct this phase to ensure that Exadata is configured in a standard and supportable way. The key step that is performed during this phase is running the Exadata deployment tool, which automates much of the configuration process. The result of this phase is a working system which is configured according to the information specified in the Exadata configuration tool.

Exadata Implementation: Overview

3. Additional configuration

- Additional activities for production readiness including:
 - Reconfigure storage using non-default settings.
 - Create additional databases.
 - Configure backup and recovery.
 - Configure Oracle Data Guard.
 - Connect Oracle Exalogic Elastic Cloud.
- Conducted by customer or Oracle services engagement

4. Post-installation

- Ongoing monitoring and maintenance
- Conducted by customer or Oracle services engagement



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

3. After installation and configuration, Exadata is configured on the customer network. At this point customers can commence the process of loading or migrating data and prepare Exadata for production use. However, customers may wish to vary the configuration of Exadata in order to create customized databases or to organize the available storage in a non-default way. Such additional configuration tasks are typically conducted by the customer or Oracle services are engaged to conduct the work. Other examples of additional configuration tasks include configuring Oracle Data Guard, configuring the backup and recovery environment, or connecting Oracle Exalogic Elastic Cloud. Common supported and unsupported additional configuration activities are discussed later in the lesson.
4. During post-installation the focus of activities shifts to on-going monitoring and maintenance of Exadata. This is usually conducted by the customer.

The rest of this lesson contains topics that will help you to understand the Exadata implementation process, particularly the installation and configuration phase (Phase 2). The aim is not to replace the available documentation, rather the material is presented to assist you to better understand the available configuration options and to better appreciate the results of making different selections.

Key Documentation

Oracle Exadata Database Machine Installation and Configuration Guide

- Important reference document covering:
 - Site planning
 - Network planning
 - Installation
 - Initial configuration



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The *Oracle Exadata Database Machine Installation and Configuration Guide* is intended primarily for those responsible for data center site planning, installation, and configuration of Exadata.

Exadata Site Preparation

- Use the following sections in the Installation and Configuration Guide to direct site preparation activities:
 - Chapter 1: Site Requirements for Oracle Exadata Database Machine
 - General Environmental Requirements
 - Space Requirements
 - Flooring Requirements
 - Electrical Power Requirements
 - Temperature and Humidity Requirements
 - Ventilation and Cooling Requirements
 - Network Connection and IP Address Requirements
 - Appendix A: Site Checklists



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The *Database Machine Installation and Configuration Guide* documents the site requirements for each Exadata model. This information should be used to direct pre-installation site preparation activities. The guide also contains a series of useful site checklists that can be used as an aid to ensure site readiness.

Exadata Configuration Tool: Overview

The Exadata configuration tool:

- Captures site-specific configuration settings including:
 - Host and domain names
 - IP addresses
 - Region and time zone information
 - Name servers and NTP time servers
 - Exadata cell notification settings
- Generates files that drive the Exadata deployment tool
- Is a Java-based program that is part of the Oracle Exadata Deployment Assistant patch bundle
 - My Oracle Support note 888828.1 lists the latest version



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

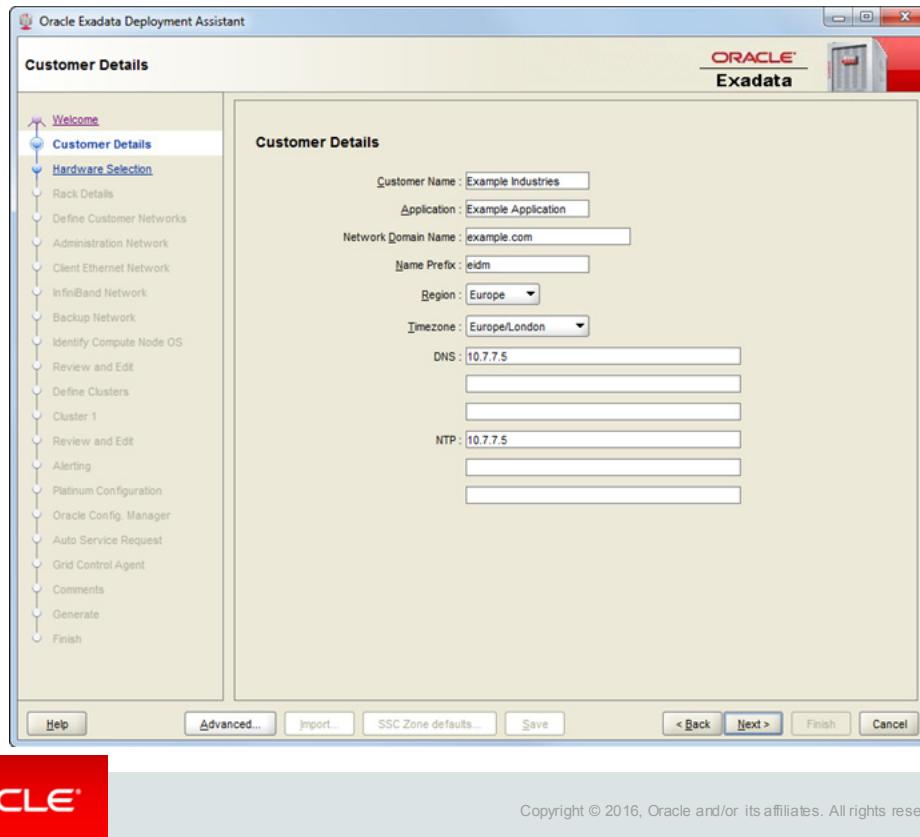
In conjunction with site preparation, one of the most important Exadata pre-installation tasks is to run the Exadata configuration tool. Completing the Exadata configuration tool is often a collaborative effort involving database, network and system administrators. Oracle engineers may also participate to provide assistance. The information gathered by the Exadata configuration tool is used to configure Exadata by driving the Exadata deployment tool through a variety of configuration tasks. The settings captured while running the assistant also direct aspects of site preparation, such as the network names and addresses that need to be configured within your network before installing Exadata.

Both the Exadata configuration tool and the Exadata deployment tool are included in the Oracle Exadata Deployment Assistant patch bundle. It is recommended to always check My Oracle Support note 888828.1 for details regarding the latest available version.

Chapter 2 of the Installation and Configuration Guide titled *Understanding the Network Requirements for Oracle Exadata Database Machine* should be considered in conjunction with running the Exadata configuration tool.

The following section introduces the Exadata configuration tool and shows a set of example configuration settings.

Exadata Configuration Tool: Customer Details

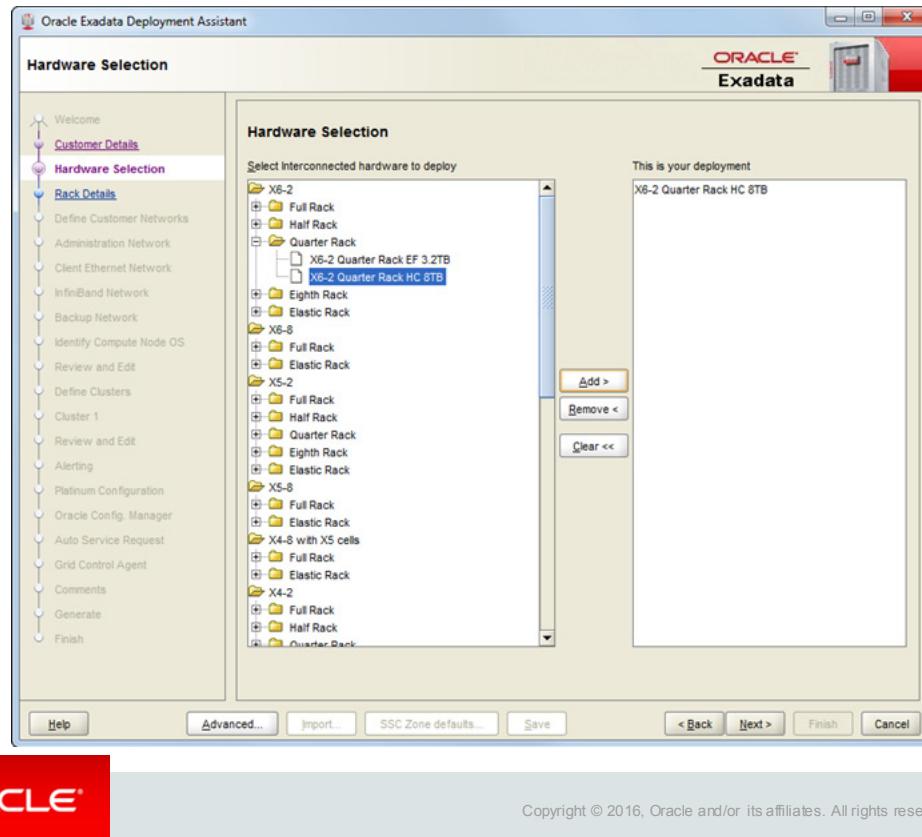


The slide shows an example of a completed Customer Details page. Entries are required for all the fields. When filling out this page, note the following items:

- **Network Domain Name:** This field must be used to specify the domain name for the Exadata servers.
- **Name Prefix:** The Name Prefix is an identifier used to generate host names for all the Exadata servers and other components such as the network switches. Indirectly, this value is also used in other identifiers such as the names of cell disks, grid disks and disk groups. Oracle recommends that this value should be 4 characters or less to avoid possible issues with names that are too long. Customers implementing multiple Database Machines often embed a numeric identifier within the Name Prefix. In this example, `eimd` is used (as an abbreviation for Example Industries Database Machine). Customers should choose a setting that reflects their own naming conventions.
- **DNS:** Use these fields to identify up to three DNS (Domain Name System) servers accessible from the Database Machine. DNS is the preferred naming service for Exadata. Grid Naming Service (GNS), available in Oracle Grid Infrastructure, is not configured.

- **NTP:** Use these fields to identify up to three NTP (Network Time Protocol) servers accessible from the Database Machine. NTP services are a mandatory requirement for Exadata. NTP provides coordinated timing which synchronizes services across Exadata. Without NTP, a lack of coordination can lead to database nodes being evicted from the cluster or Exadata cells being excluded from the storage pool. NTP services also ensure that the timestamps written to the various log files are coordinated across Exadata. Note that the Cluster Time Synchronization Service (CTSS) that was introduced in Oracle Clusterware11g Release 2 cannot be used to provide time services to Exadata cells; therefore, CTSS is not used for Exadata.

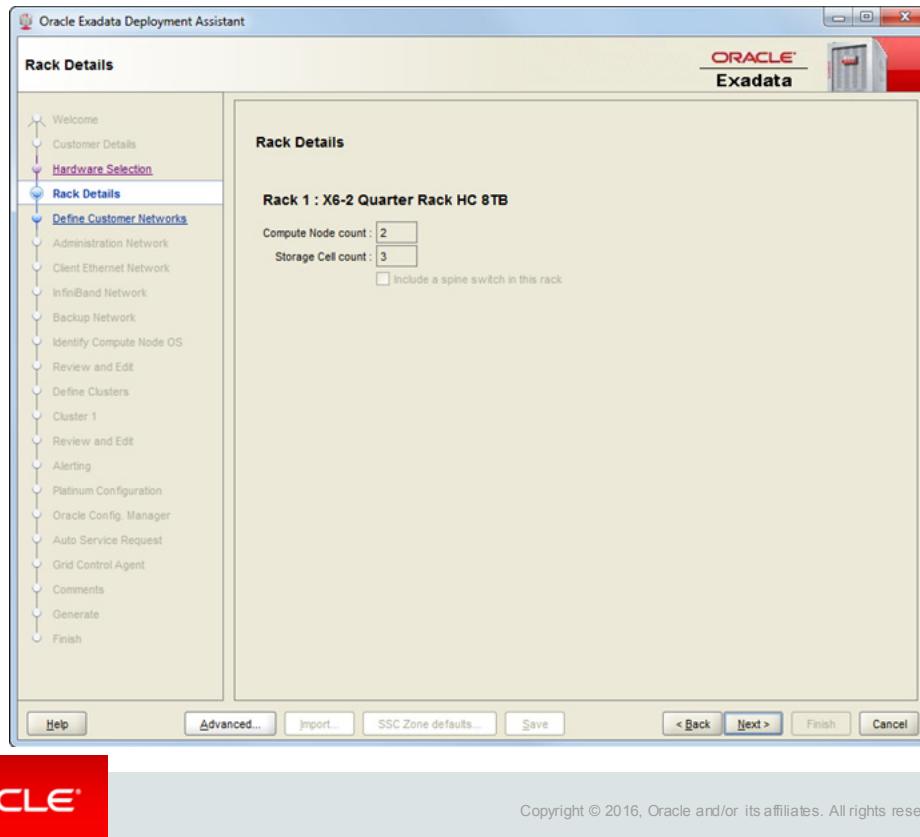
Exadata Configuration Tool: Hardware Selection



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

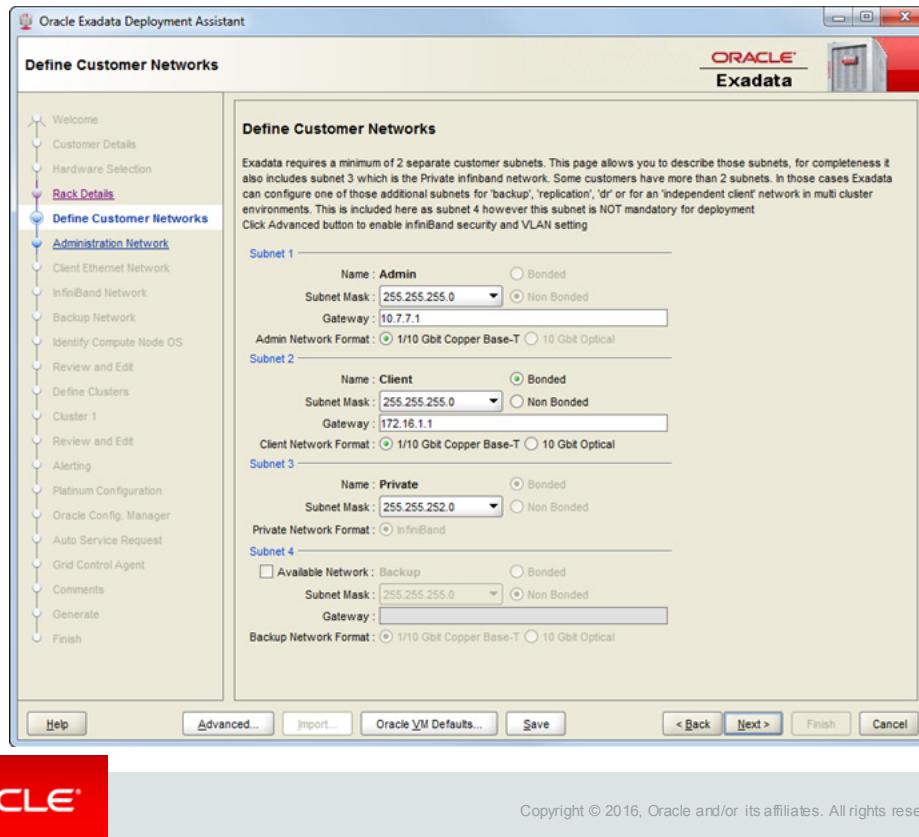
The slide shows an example of the Hardware Selection page. Use this page to specify the Exadata model being deployed. This page may be used to specify multiple interconnected Database Machines. Note that if multiple racks are specified on this page, all the server hostnames generated by the deployment assistant will be based on the Name Prefix value specified previously on the Customer Details page. If a different naming convention is desired for each rack, this may be achieved by multiple invocations of the deployment assistant, or by making manual adjustments later in the deployment assistant.

Exadata Configuration Tool: Rack Details



The slide shows an example of the Rack Details page. This page confirms the rack details based on the selections made previously on the Hardware Selection page. No inputs are required on this page.

Exadata Configuration Tool: Define Customer Networks

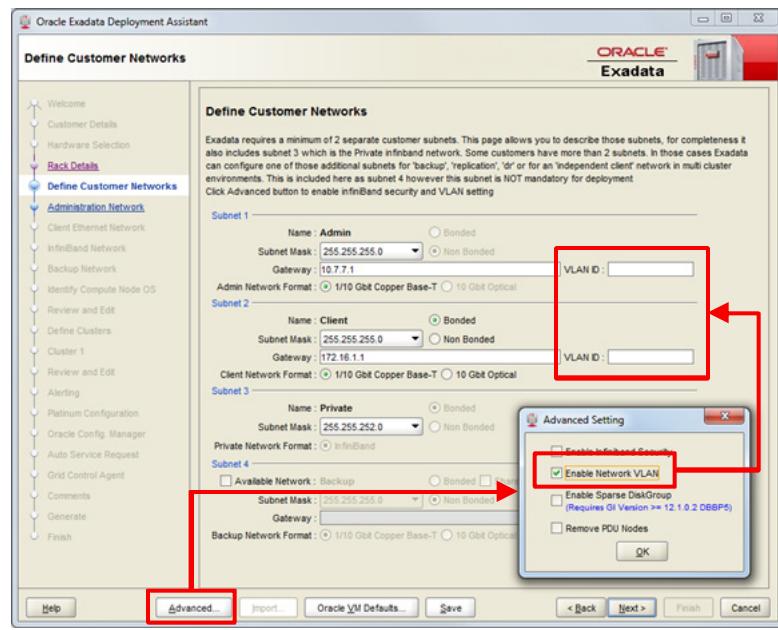


The slide shows an example of the Define Customer Networks page. Use this page to specify attributes for each network type. Depending on the network type, settings may include the subnet mask, gateway address, interface type and bonding selection. Various options may be disabled depending on whether or not they are applicable to the Exadata model being configured.

In addition to the administration (subnet 1), client (subnet 2) and private (subnet 3) networks, you can use this page to specify the creation of a secondary client network, known in the deployment assistant as the backup network (subnet 4).

VLAN Support

- Previously, VLAN tagging had to be manually implemented after the initial configuration of Exadata.
- With Exadata release 12.1.2.3.0, you can use the Oracle Exadata Deployment Assistant to configure VLANs.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

With Exadata software release 12.1.2.3.0, Oracle Exadata Deployment Assistant now supports creating VLANs on compute nodes and storage servers for the administration, ILOM, client access, and backup networks. Note the following general requirements for VLAN support:

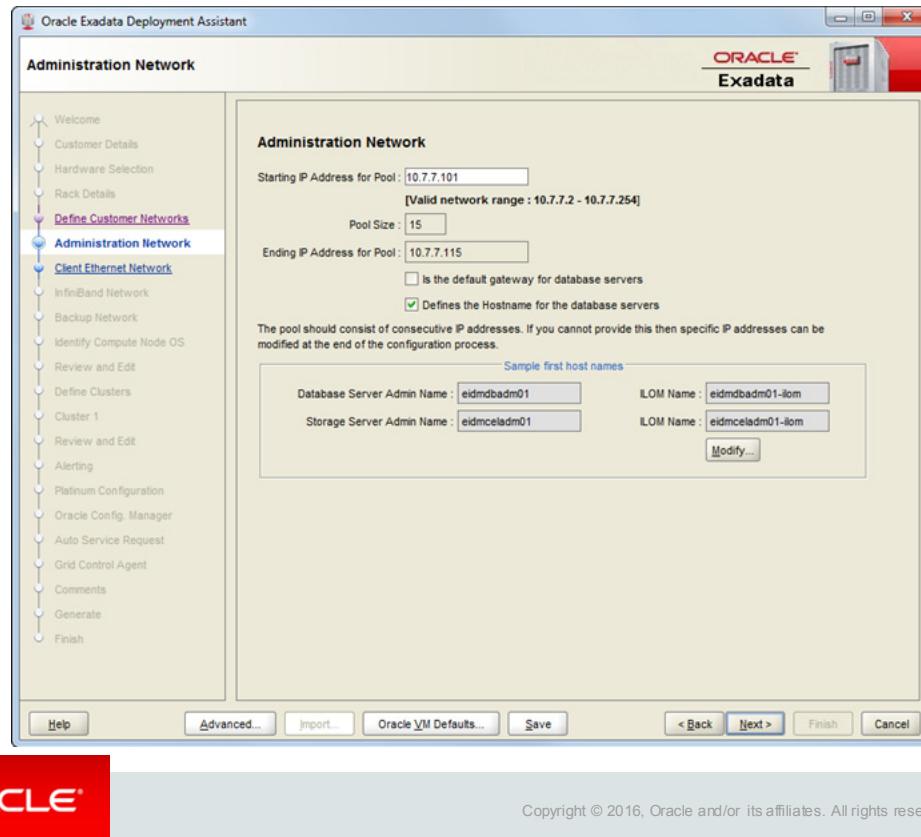
- Except for the administration network, which is never bonded, all other VLAN networks must be bonded.
- If the backup network is on a tagged VLAN network, the client access network must also be on a separate tagged VLAN network.
- The backup and client access networks can share the same network cables.
- IPv6 VLANs are not supported on the administration network.
- Virtual deployments do not support IPv6 VLANs.

Also, VLAN support is dependent on the Exadata system and Oracle Database version in use. The following outlines the supported combinations:

- VLAN tagging on the administration network is supported with IPV4 only on all Exadata models starting with Exadata X4, and also Exadata X3-2 (but not X3-8).
- VLAN tagging on the client access and backup networks is supported with both IPV4 and IPV6 on all supported hardware models. For IPV6 in conjunction with Oracle Database 12c version 12.1.0.2 and later, patch 22289350 is also required.

See *Using Network VLAN Tagging with Oracle Exadata Database Machine* in the *Oracle Exadata Database Machine Installation and Configuration Guide* for details.

Exadata Configuration Tool: Administration Network



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of a completed Administration Network page. Entries on this page are used to configure the administration network interfaces for all the database servers, Exadata cells and administration network interfaces on other Exadata components.

As a minimum, administrators must set the “Starting IP Address for Pool” field to the first IP address in the range of IP addresses allocated for the administration network. Using this address as a starting point, the configuration programs sequentially allocate IP addresses for each network interface on the administration network.

The bottom of this page shows sample host names based on the Name Prefix value specified previously on the Customer Details page, and the default naming conventions proposed in the Exadata configuration tool. Click Modify to make modifications to the default naming convention for administration network host names.

Administration Network

IP Address Allocation: Example

Interface Type	Hostname	IP Address
Database server management interface	eidmdbadm01	10.7.7.101
Database server management interface	eidmdbadm02	10.7.7.102
Exadata cell management interface	eidmceladm01	10.7.7.103
Exadata cell management interface	eidmceladm02	10.7.7.104
Exadata cell management interface	eidmceladm03	10.7.7.105
Database server ILOM interface	eidmdbadm01-ilom	10.7.7.106
Database server ILOM interface	eidmdbadm02-ilom	10.7.7.107
Exadata cell ILOM interface	eidmceladm01-ilom	10.7.7.108
Exadata cell ILOM interface	eidmceladm02-ilom	10.7.7.109
Exadata cell ILOM interface	eidmceladm03-ilom	10.7.7.110
Ethernet switch management interface	eidmsw-adm01	10.7.7.111
InfiniBand switch management interface	eidmsw-iba01	10.7.7.112
InfiniBand switch management interface	eidmsw-ibb01	10.7.7.113
PDU management interface	eidm-pdua01	10.7.7.114
PDU management interface	eidm-pdub01	10.7.7.115



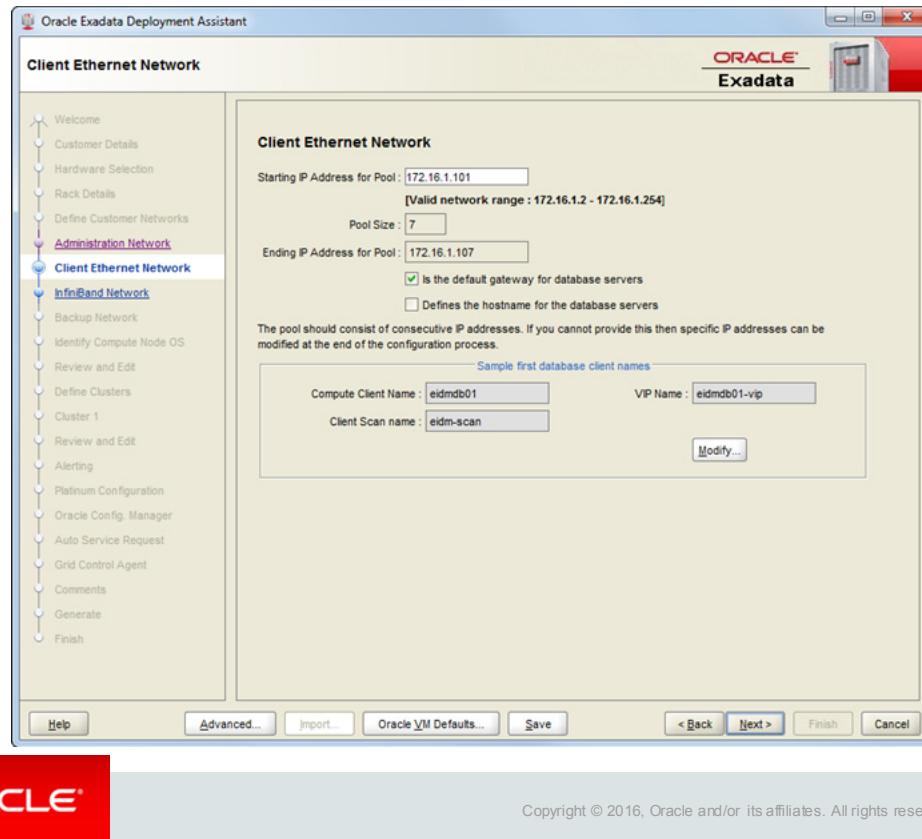
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Using the X6-2 Quarter Rack example shown so far in this lesson, IP addresses and host names would be allocated by default as shown in the table on the slide. Note that the addresses are allocated sequentially within each interface type as shown in the table.

It is possible to change the default addressing policy by setting specific values later in the Exadata configuration tool. However this is not recommended because the default allocation order is well understood by Oracle service and support engineers and any changes could lead to confusion if the defaults are later assumed. Also, beware that if the default addressing policy is modified, care must be taken to ensure that the resulting addresses are valid and there are no duplicates.

Note: The IP address allocations in this example assume that the Database Machine is configured without virtualized compute nodes. If virtualized compute nodes were configured, additional database server management interfaces would be configured for each virtual machine.

Exadata Configuration Tool: Client Ethernet Network



The slide shows an example of a completed Client Ethernet Network page. Entries in this worksheet are used to configure the client access network interfaces for all the Exadata database servers.

As a minimum, administrators must set the “Starting IP Address for Pool” field to the first IP address in the range of IP addresses allocated for the client access network. Using this address as a starting point, the configuration programs sequentially allocate IP addresses in the following order:

1. Database server client access network interfaces
2. Database server virtual network interfaces (VIPs)
3. Single Client Access Name (SCAN) addresses

The bottom of this page shows sample host names based on the Name Prefix value specified previously on the Customer Details page, and the default naming conventions proposed in the Exadata configuration tool. Click Modify to make modifications to the default naming convention for client access network host names.

Client Ethernet Network IP Address Allocation: Example

Address Type	Hostname	IP Address
Database server physical client access interface	eidmdb01	172.16.1.101
Database server physical client access interface	eidmdb02	172.16.1.103
Database server VIP	eidmdb01-vip	172.16.1.102
Database server VIP	eidmdb02-vip	172.16.1.104
SCAN address	eidm-scan	172.16.1.105
SCAN address	eidm-scan	172.16.1.106
SCAN address	eidm-scan	172.16.1.107



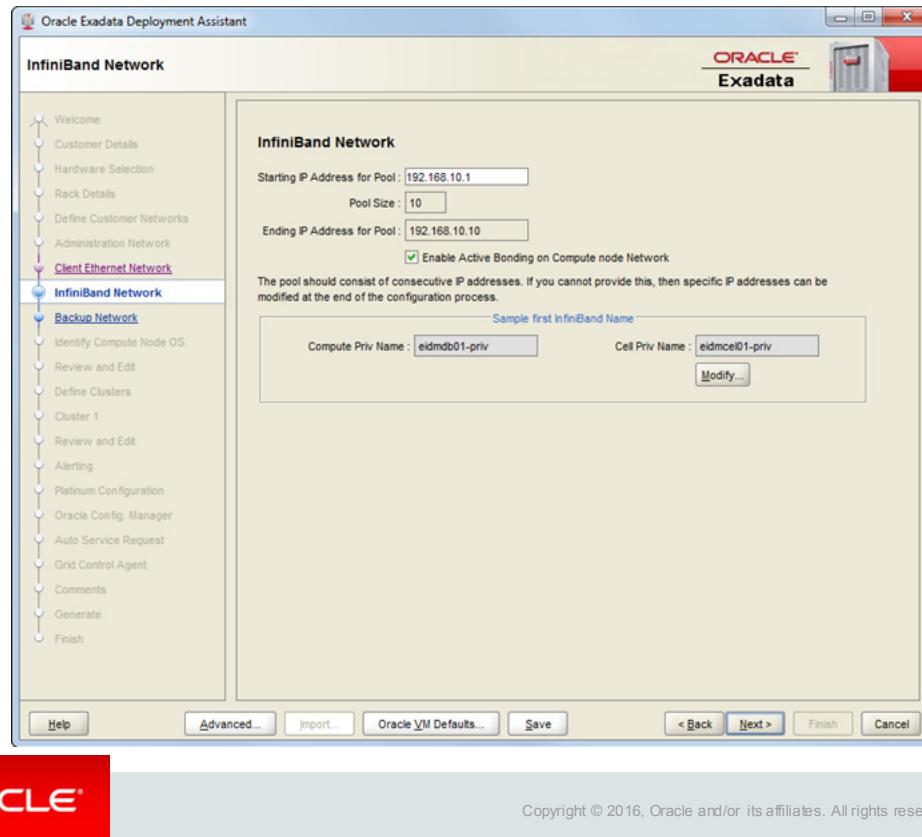
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Using the Quarter Rack example shown so far in this lesson, IP addresses and host names would be allocated by default as shown in the table on the slide. Note that the addresses are allocated sequentially from the starting address specified in the Exadata configuration tool.

Like the administration network IP addresses, it is possible to change the default addressing policy for the client access network by setting specific values later in the Exadata configuration tool. However this is not recommended since the default allocation order is well understood by Oracle service and support engineers and any changes could lead to confusion if the defaults are later assumed. Also, beware that if the default addressing policy is modified, care must be taken to ensure that the resulting addresses are valid and there are no duplicates.

Note: The IP address allocations in this example assume that the Database Machine is configured without virtualized compute nodes. If virtualized compute nodes were configured, additional database server client interfaces would be configured for virtual machine.

Exadata Configuration Tool: InfiniBand Network



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of a completed InfiniBand Network page. Entries in this worksheet are used to configure the InfiniBand network interfaces for all the Exadata database servers and Exadata cells. The default values suggested on this page are sufficient in situations where the Exadata InfiniBand network is not connected to any other systems. Otherwise, adjustments may be required to fit in with existing systems, such as backup servers or Exalogic systems.

InfiniBand Network IP Address Allocation: Example

Address Type	Hostname	IP Address
Database server InfiniBand network interface	eidmdb01-priv1	192.168.10.1
Database server InfiniBand network interface	eidmdb01-priv2	192.168.10.2
Database server InfiniBand network interface	eidmdb02-priv1	192.168.10.3
Database server InfiniBand network interface	eidmdb02-priv2	192.168.10.4
Storage server InfiniBand network interface	eidmcel01-priv1	192.168.10.5
Storage server InfiniBand network interface	eidmcel01-priv2	192.168.10.6
Storage server InfiniBand network interface	eidmcel02-priv1	192.168.10.7
Storage server InfiniBand network interface	eidmcel02-priv2	192.168.10.8
Storage server InfiniBand network interface	eidmcel03-priv1	192.168.10.9
Storage server InfiniBand network interface	eidmcel03-priv2	192.168.10.10



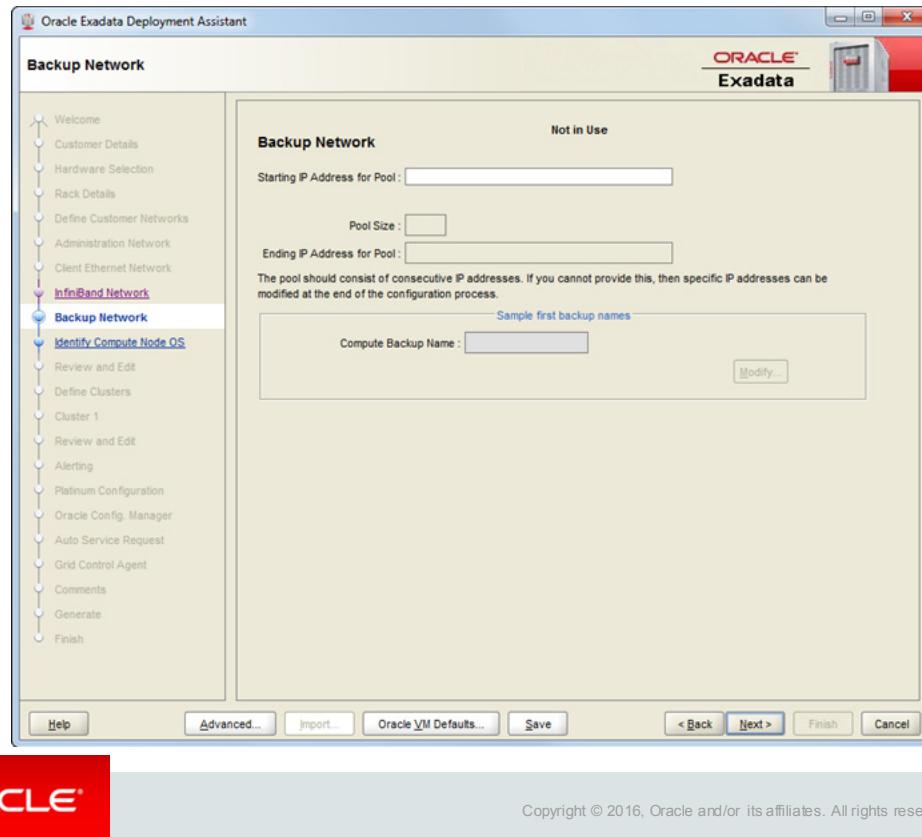
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Using the Quarter Rack example shown so far in this lesson, IP addresses and host names would be allocated by default as shown in the table on the slide. Note that the addresses are allocated sequentially from the starting address specified in the Exadata configuration tool. Note also that because active bonding was specified, each database server and each Exadata Storage Server is associated with two interfaces, each with a separate IP address.

Like the administration and client network IP addresses, it is possible to change the default addressing policy for the InfiniBand network by setting specific values later in the Exadata configuration tool. However this is not recommended since the default allocation order is well understood by Oracle service and support engineers and any changes could lead to confusion if the defaults are later assumed. Also, beware that if the default addressing policy is modified, care must be taken to ensure that the resulting addresses are valid and there are no duplicates.

Note: The IP address allocations in this example assume that the Database Machine is configured without virtualized compute nodes. If virtualized compute nodes were configured, additional database server InfiniBand interfaces would be configured for each virtual machine.

Exadata Configuration Tool: Backup Network

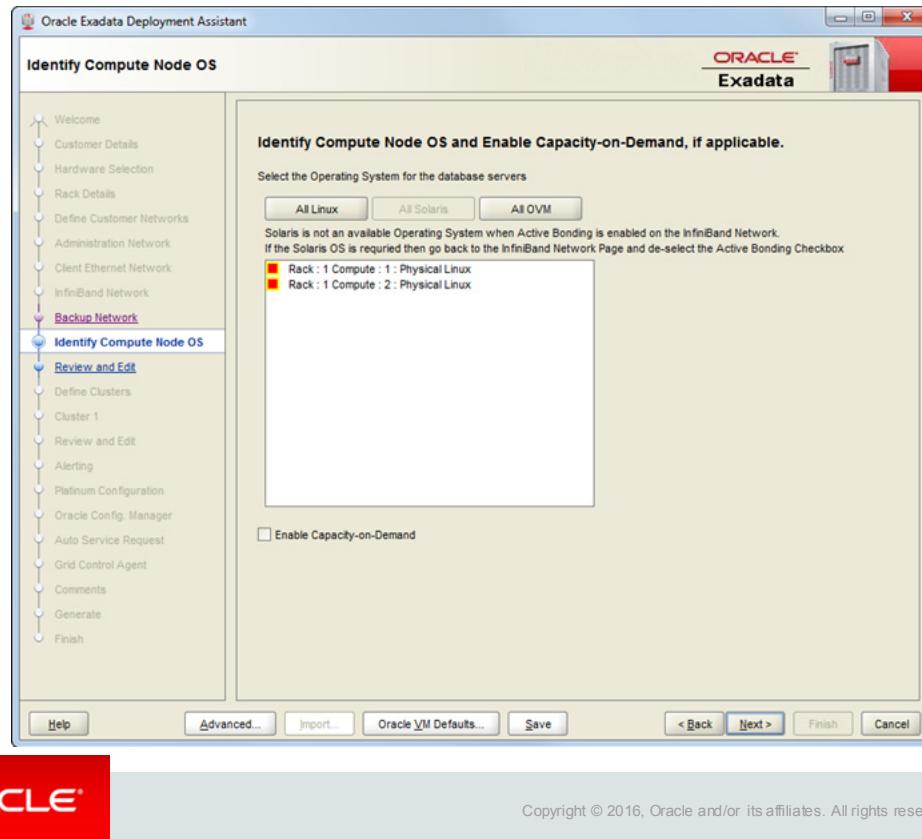


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata database servers contain additional network ports that can be optionally configured to connect to another network. For example, a secondary network could be configured to connect Exadata to a dedicated network containing a tape silo for backup purposes.

Additional network configuration can be performed using the Backup Network page, but only if the option to configure a backup network was selected back on the Define Customer Networks page. Otherwise the page contains a series of empty fields as shown by the example in the slide.

Exadata Configuration Tool: Identify Compute Node OS



Use the Identify Compute Node OS page to specify the operating system to configure on the compute nodes. By default, Oracle Linux is selected as the compute node operating system.

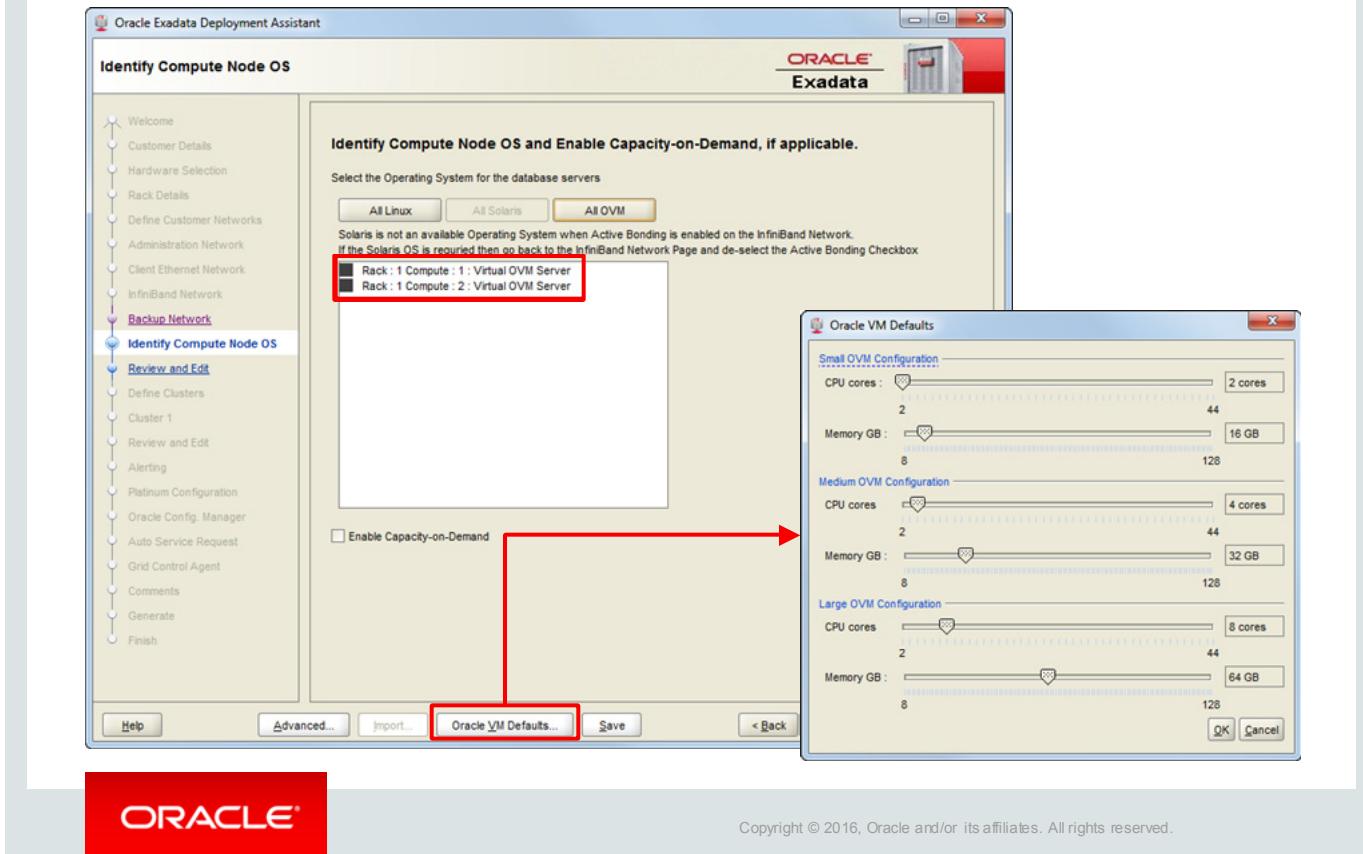
Alternatively, you can choose to configure Oracle Virtual Machine (OVM) on the compute nodes. OVM can then be used to host one or more virtual machines containing Oracle Linux and Oracle Database software. OVM enables users to use virtual machines to isolate database workloads on Exadata from the underlying server hardware and from each other, which is very useful in scenarios where you wish to consolidate databases on an Exadata platform.

Older Exadata models also provide the option to use Oracle Solaris as the database server operating system. However, this option is no longer supported starting with the Exadata X5 models.

Note that the Exadata cells always use the Linux operating system.

This page also provides the option to enable capacity-on-demand, which is a feature that allows database server cores to be disabled in order to limit the Oracle software licenses that are required for the system. If you choose to enable capacity-on-demand you are also required to specify the number of active CPU cores you wish for each database server.

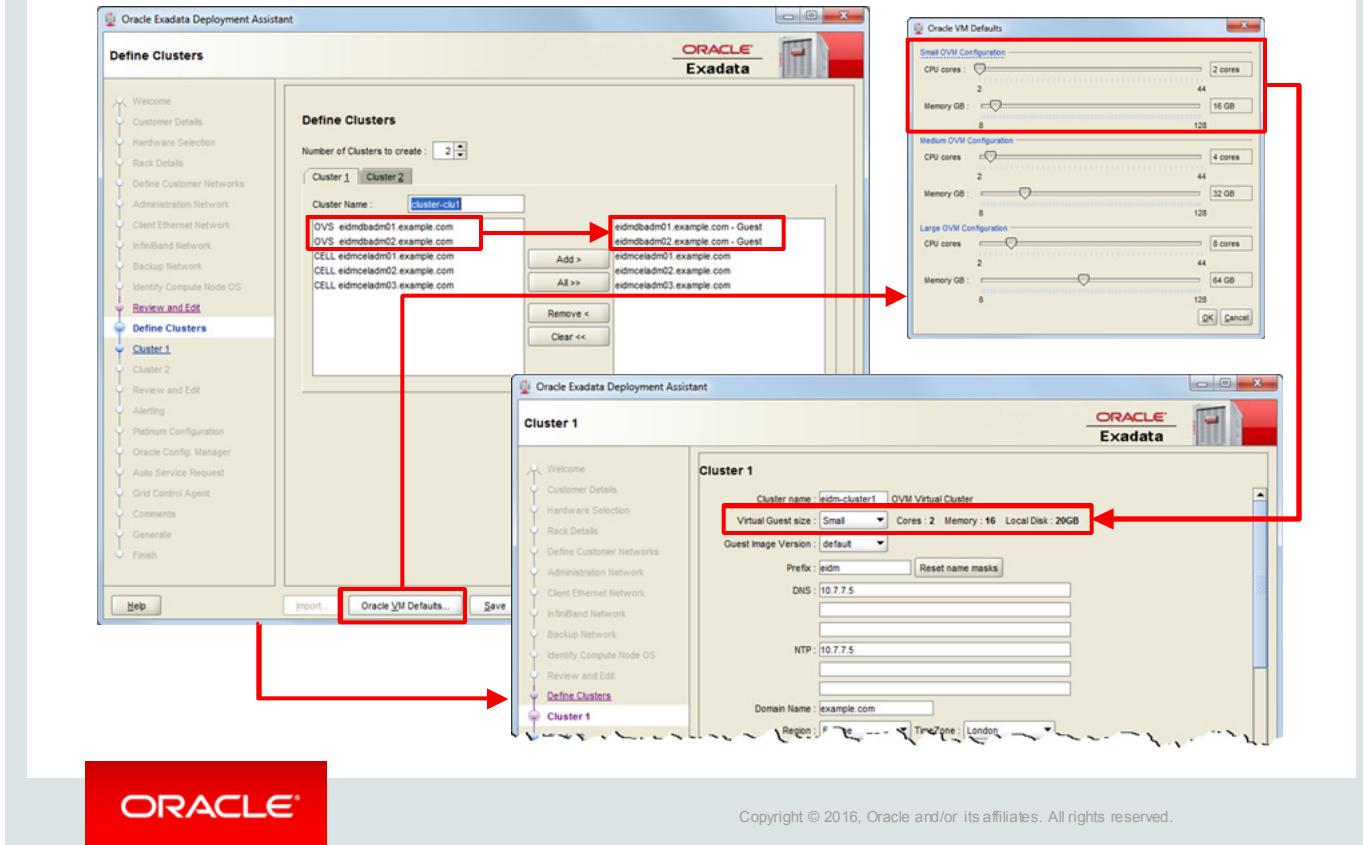
Configuring Virtualized Compute Nodes



To configure virtualized compute nodes use the Identify Compute Node OS page. Simply click each node in the list to cycle through the available physical and virtual OS implementation options. The example in the slide shows two nodes configured to run Oracle Virtual Machine server. You can configure a rack to contain a mixture of physical and virtual compute nodes; however, you cannot include a mixture of physical and virtual compute nodes in the same cluster.

Click the Oracle VM Defaults button to access a dialog that allows you to specify the default sizes for small, medium and large VMs. You will use these definitions later when you associate a VM size with the virtualized compute nodes in a cluster.

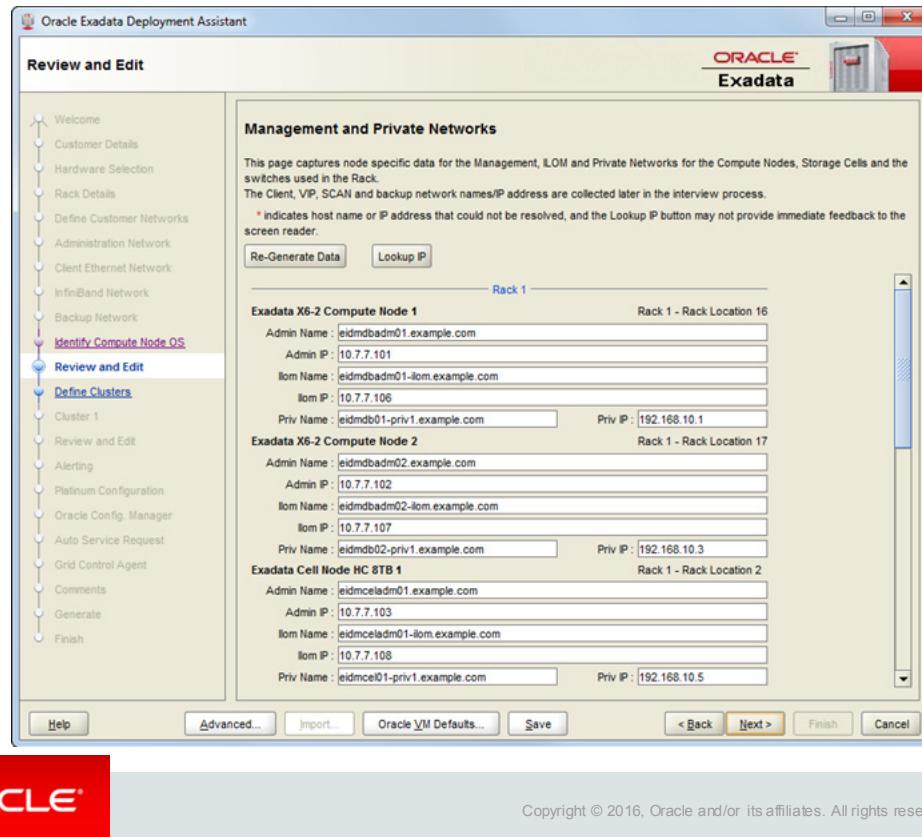
Configuring Virtualized Compute Nodes



After you use the Identify Compute Node OS page to configure compute nodes to run Oracle Virtual Machine server, the nodes later appear in the Define Clusters page and are identified as OVS nodes in the list of available nodes. When a node is added to a cluster, the node is identified as a Guest VM.

Later, when you use the Cluster page, you are invited to specify the Virtual Guest Size. The available options are small, medium and large, and these options relate to the settings in the Oracle VM Defaults dialog.

Exadata Configuration Tool: Review and Edit

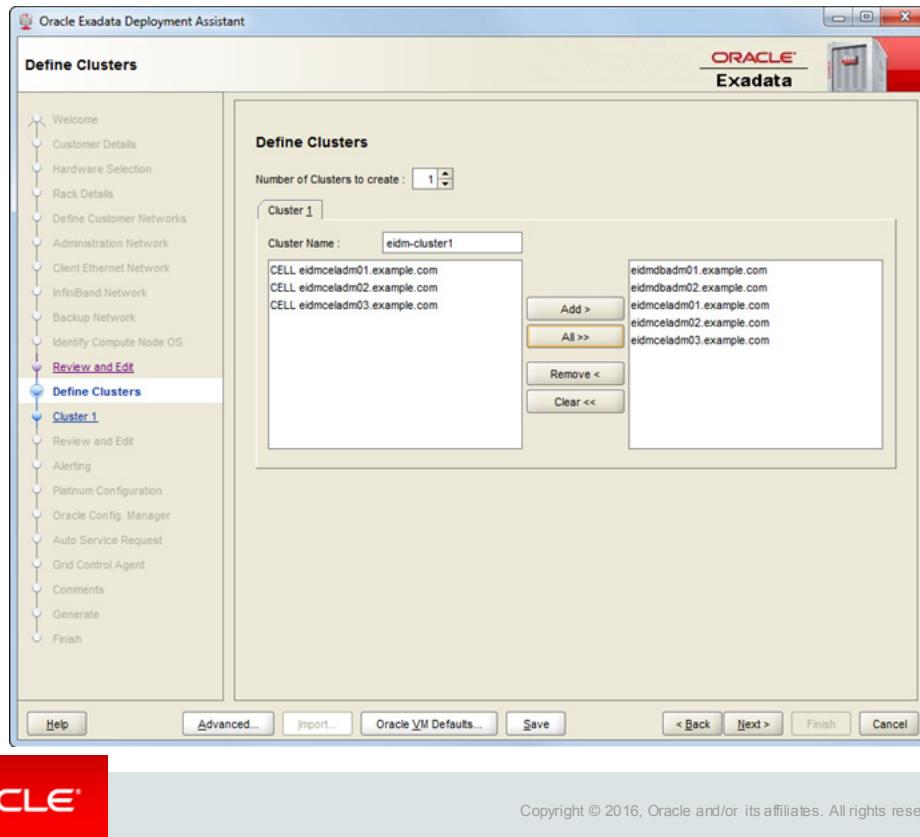


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of a completed "Review and Edit" page. The page contains management, ILOM and private network configuration information based on the settings supplied in the previous pages. You can use this page to make manual modifications to specific host names or IP addresses. However, this is not recommended since the default IP allocation order and host naming convention is well understood by Oracle service and support engineers and any changes could lead to confusion if the defaults are later assumed. Also, beware that if the default addressing policy is modified, care must be taken to ensure that the resulting addresses are valid and there are no duplicates.

You can undo any manual changes by using the Re-Generate Data button. The Lookup IP button performs a basic check to determine if any of the IP addresses listed on the page already exist in your network.

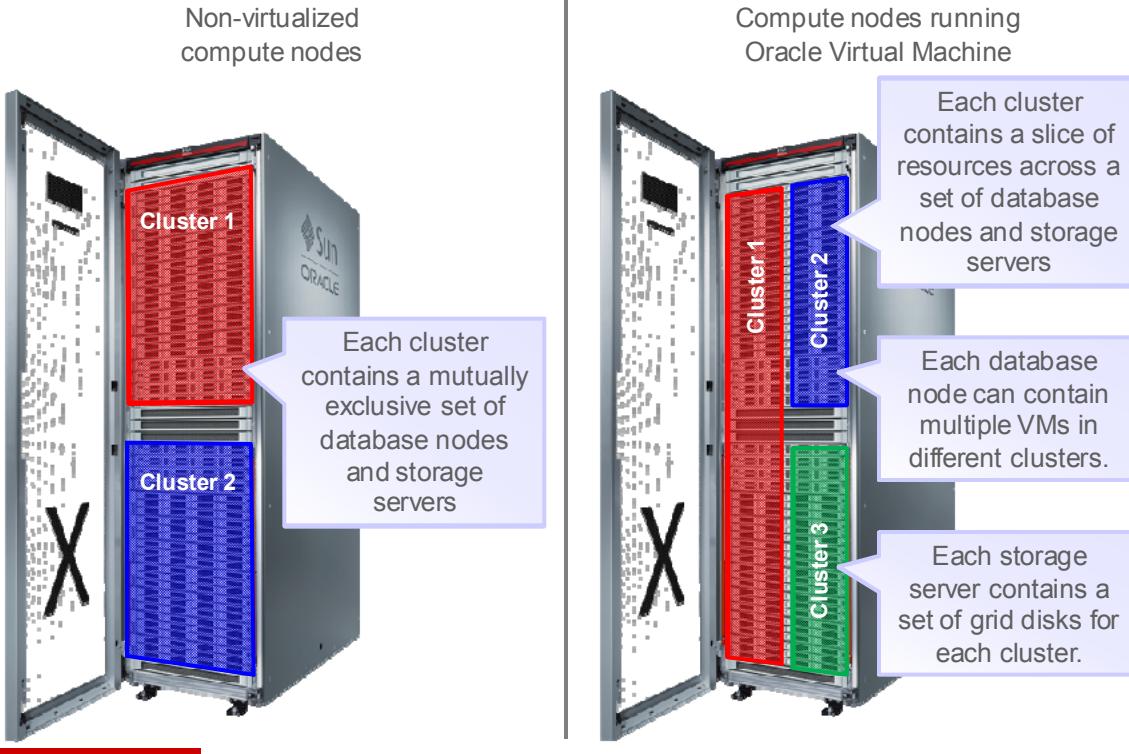
Exadata Configuration Tool: Define Clusters



The Define Clusters page allows you to define at least one, and up to 32, clusters across your Exadata environment. Use this page to first specify the number of clusters to define. Then, name each cluster and specify the associated servers. The example in the slide shows the common situation where all of the servers are associated with a single cluster.

Note: You should ensure that each cluster name is unique within your organization.

Multiple Cluster Configuration Options



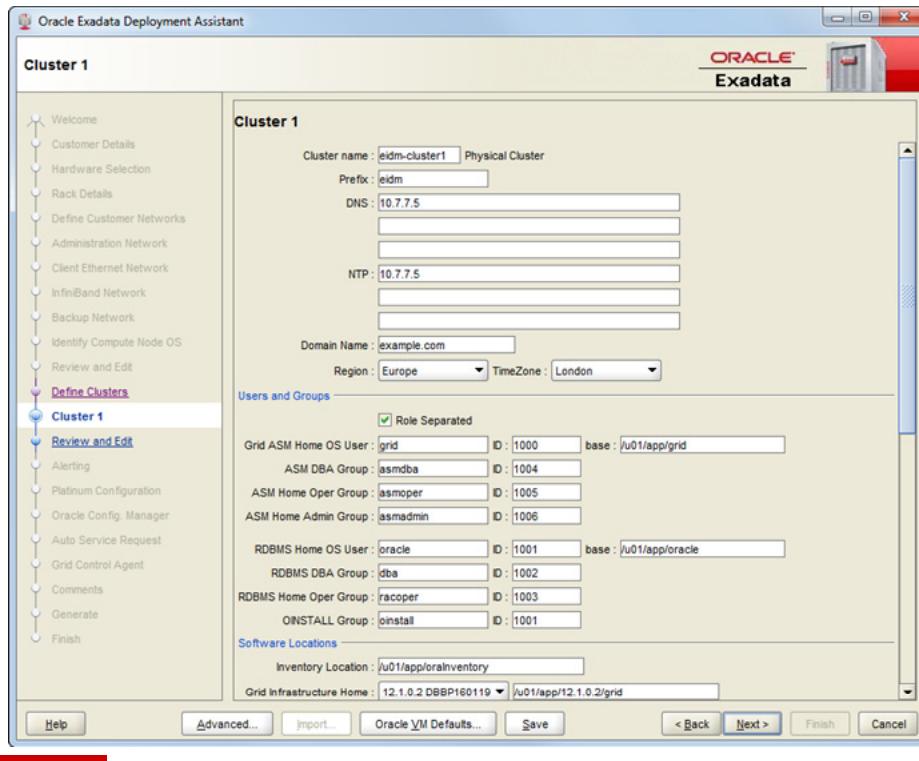
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The behavior of the Define Clusters page, and the resulting configuration, is influenced by the compute node OS settings specified in the Identify Compute Node OS page. The diagrams in the slide illustrate the fundamental differences.

As shown on the left of the slide, non-virtualized compute nodes can only belong to one cluster. Therefore, each cluster must contain a mutually exclusive set of compute nodes. Furthermore, when an Exadata Storage Server is associated with a cluster containing non-virtualized compute nodes, the Exadata configuration tool enforces a rule that does not allow the storage server to be associated with any other cluster.

For compute nodes running Oracle Virtual Machine the situation is different. As shown on the right side of the slide, a cluster containing virtualized compute nodes occupies a slice of resources that spans a set of database servers and Exadata Storage Servers. On the database servers, each cluster consists of a set of virtual machines. On the Exadata Storage Servers, each cluster is allocated a slice of storage by using a separate set of grid disks.

Exadata Configuration Tool: Cluster Configuration - Part 1



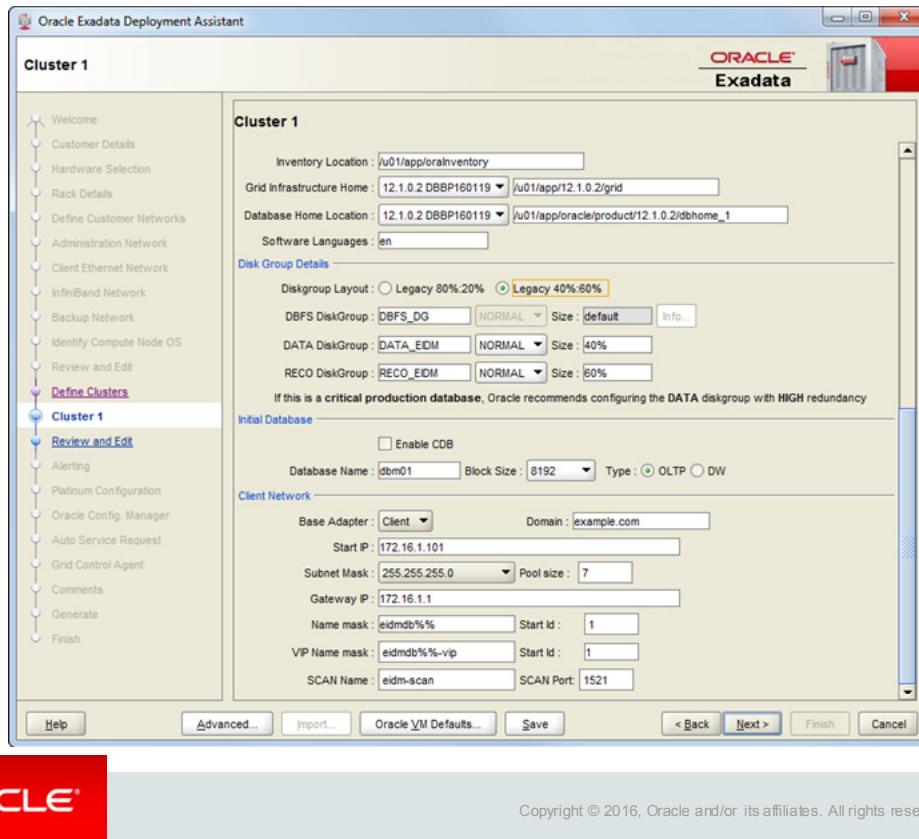
For each cluster defined in the Define Clusters page, you must complete a separate cluster configuration page. The slide shows an example of the first part of a completed cluster configuration page.

The initial section contains previously supplied cluster name, prefix, DNS, NTP, domain name, and region information.

Following this you must complete sections to specify the configuration for:

- **Users and Groups:** Use this section to specify the OS users and groups that manage the Oracle Database software. A single owner can be specified for Grid Infrastructure and Oracle Database software, or a role separated installation can be defined (as shown in the slide).
- **Software Locations:** Use this section to specify the Oracle Home locations, the installation inventory location, and the software versions to install.

Exadata Configuration Tool: Cluster Configuration - Part 2



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

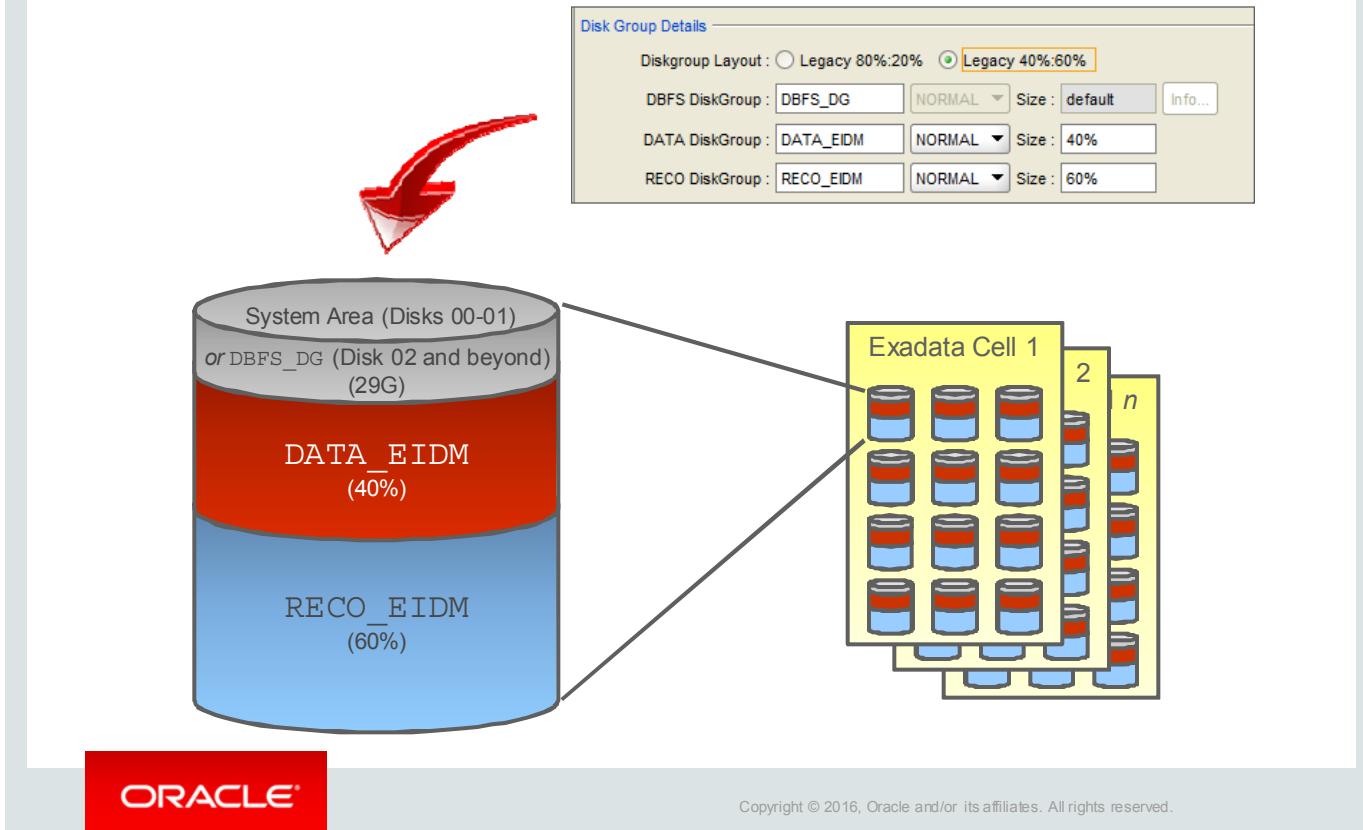
Following on from the previous slide, this slide shows an example of the second part of a completed cluster configuration page. Use the remaining sections to specify the configuration for:

- **Disk Group Details:** Use this section to name the DBFS, DATA and RECO disk groups, and specify the redundancy level for the DATA and RECO disk groups. A detailed discussion on redundancy settings is provided later in this lesson. You can also specify how the available space is allocated between the DATA and RECO disk groups. Two recommended disk group layouts are available:
 - The Legacy 80%:20% option is typically used in situations where you do not plan to store a full database backup in the RECO disk group. This setting is appropriate where external storage is used for backups, such as additional dedicated Exadata Storage Servers, an NFS server, or tape library.
 - The Legacy 40%:60% option is typically used in situations where you want to store a full database backup in the RECO disk group.

Alternatively, you can specify your own custom disk group layout by specifying custom values for the size of each disk group.

- **Initial Database:** Use this section to configure the initial database. The initial database incorporates recommended settings that are optimized for Exadata. However, the initial database offers limited configuration options, so if administrators desire a different specific configuration it is recommended to delete the initial database and create another database with the desired settings.
- **Client Network:** Use this section to provide configuration settings for the client network, including the SCAN and VIP names.

Exadata Storage Configuration Example



During the installation and configuration process, Exadata storage is configured symmetrically across the cells that are associated with each cluster. That is, the storage in each Exadata cell is configured in the same way as the other storage servers.

A cell disk is defined for each physical disk in a High Capacity cell, or each flash drive in an Extreme Flash cell. The cell disks use the default naming convention of `CD_<nn>_<cell host name>` where `<nn>` is a two-digit identifier for each disk and `<cell host name>` is the canonical host name for the storage server. For example, `CD_01_eidmceladm01`.

From there, the cell disks are carved up into a series of grid disks based upon the cluster definitions specified in the Exadata configuration tool. This slide shows an example based on the Exadata configuration tool pages shown earlier in the lesson. In this case, one cluster occupies the entire Database Machine.

In this example, a grid disk with the prefix `DBFS_DG` is created on every cell disk, apart from the first two (`<nn>=00 or 01`). The grid disks are each approximately 29 GB in size and are allocated to the `DBFS` disk group. On the first two disks (`<nn>=00 or 01`) in each cell, approximately 29 Gigabytes is used to store the system area, which contains the OS image, swap space, Exadata software binaries, metric and alert repository and various other configuration and metadata files. Because of the layout of the system areas and the `DBFS` disk group, the remaining space on every cell disk is equal.

In the remaining space on every disk, grid disks are created and allocated to the DATA and RECO disk groups. The name of each disk group is specified in the Exadata configuration tool, and is used as the prefix for each grid disk name.

The sizing of the DATA and RECO grid disks, and ultimately the sizing of the tablespaces that consume them, is determined by the disk group layout setting specified in the deployment assistant. If the Legacy 40%:60% option is selected (as shown in the screenshot on the slide), then 40% of the remaining space on each disk is allocated to the DATA disk group and the remaining 60% is allocated to the RECO disk group. If the Legacy 80%:20% option is selected, then the division of the remaining space is 80% to the DATA disk group and 20% to the RECO disk group. Alternatively, you can define your own specific disk group layout to divide the available space as required.

The final factor which impacts on the storage configuration is the ASM redundancy setting (normal or high) for the DATA and RECO disk groups. The DBFS disk group uses normal ASM redundancy.

Using the Quarter Rack example shown throughout this lesson results in the following storage configuration:

- Each 8 TB high capacity disk would have approximately 29 GB reserved for the Exadata Storage Server system area or the DBFS_DG disk group. Out of the remaining space on each disk, 40% (approximately 3.2 TB) would be allocated to the DATA_EIDM disk group, and 60% (approximately 4.8 TB) would be allocated to the RECO_EIDM disk group.
- The total space allocated to each disk group across all three Exadata cells would be:
 - Approximately 870 GB (29 GB x 10 disks per cell x 3 cells) to DBFS_DG.
 - Approximately 114 TB (3.2 TB x 12 disks x 3 cells) to DATA_EIDM.
 - Approximately 172 TB (4.8 TB x 12 disks x 3 cells) to RECO_EIDM.
- After normal ASM redundancy is applied the amount of usable space in each disk group would be:
 - Approximately 436 GB for DBFS_DG.
 - Approximately 52 TB for DATA_EIDM.
 - Approximately 78 TB for RECO_EIDM.

Configurations with multiple clusters work in essentially the same way as this example, except that each cluster is associated with a separate set of grid disks that are used to provision separate DBFS, DATA and RECO disk groups for each cluster. The way in which the storage is carved up depends on whether the cluster uses virtualized compute nodes. For clusters containing non-virtualized compute nodes, each storage server is nominally associated with only that cluster. Consequently, all of the storage on each cell is divided amongst one set of DBFS, DATA and RECO disk groups in a manner similar to the example in the slide. For clusters containing virtual compute nodes, the associated storage servers could be shared with other clusters, with each cluster receiving a slice of the available storage.

Note that the default storage configuration can be manually altered by customers after initial configuration. My Oracle Support note 1465230.1 provides guidance on how to alter a disk group without downtime on Exadata.

Choosing the Right Disk Group Redundancy Setting

- You cannot change the disk group redundancy setting without dropping and re-creating the disk group
- **HIGH** redundancy:
 - Triple mirroring across three separate storage servers
 - Provides maximum protection
 - Double mirroring still maintained if one cell is offline
 - Generally recommended for critical disk groups
 - Requires more storage capacity
 - Writes require greater I/O bandwidth.
- **NORMAL** redundancy:
 - Double mirroring across two separate storage servers
 - Provides one layer of redundancy
 - No additional protection if one cell is offline
 - Requires extra time and effort to maintain redundancy through planned maintenance
- Free space management:
 - Free space is required to preserve redundancy.
 - Consider the impact of losing a disk and the impact of losing a cell.
 - Consider also the desired protection level and the number of available cells.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Choosing the appropriate level of ASM redundancy (**NORMAL** or **HIGH**) for your disk groups is a major decision that impacts the storage capacity, fault tolerance, and even performance of Exadata. Note that you cannot change the redundancy setting for a disk group without dropping and re-creating it, which adds to the importance of making the right choice.

There is no general recommendation that suits all circumstances, rather customers must weigh the effects of each setting and select the one that best suits their circumstances. Bear in mind the following:

HIGH redundancy

HIGH redundancy results in three copies of data being written to storage on three separate Exadata Storage Servers. It provides the greatest protection, tolerating the simultaneous loss of up to two entire cells.

HIGH redundancy is generally recommended for disk groups that house critical databases. This is especially the case for disk groups based on high capacity cells because the amount of time required to restore redundancy with high capacity disks (8 TB) increases the potential for a second failure to occur before the first failure is completely dealt with.

One cost associated with this protection is that the overall storage capacity of Exadata is effectively reduced by one third compared with **NORMAL** redundancy. For example, a Full Rack Database Machine with high capacity disks has a raw disk capacity of 1344 TB. With **HIGH** redundancy applied the usable capacity becomes approximately 400 TB compared to approximately 600 TB using **NORMAL** redundancy.

Another cost is that all writes to **HIGH** redundancy disk groups must be performed on three separate storage servers. This means that 50% more I/O bandwidth is consumed by writes on **HIGH** redundancy disk groups compared with writes to **NORMAL** redundancy disk groups.

Extra I/O is not required for read operations. So the balance of application reads and writes, coupled with the I/O bandwidth requirements of the application should be considered when selecting the appropriate protection level.

NORMAL redundancy

NORMAL redundancy results in two copies of data being written to storage on two separate Exadata Storage Servers. As already highlighted, **NORMAL** redundancy has lower costs in terms of storage capacity and I/O bandwidth consumption for write operations.

Using **NORMAL** redundancy on Exadata provides the good data protection, seamlessly tolerating the loss of up to one entire cell. However, note that there are situations where data protection may be compromised using **NORMAL** redundancy.

Firstly, consider that some maintenance operations, such as cell patching for example, require that a cell is taken offline for a period of time. During that time, data availability is maintained using the remaining copy. However, what happens if a failure affects the only remaining copy? The disk group would be taken offline, disrupting or possibly even terminating database processing. This situation can be protected against by removing the cell from the disk group and rebalancing the data on to other cells prior to taking the cell offline. However, this operation requires a substantial amount of time and would consume a substantial amount of I/O bandwidth. It would also require sufficient free space to be available on the remaining cells.

Similarly, using **NORMAL** redundancy, the simultaneous loss of two disks on different storage servers may result in some portion of a database becoming unavailable. While this is a highly unlikely scenario, it is far more likely than suffering a comparable interruption using **HIGH** redundancy.

Free space management

When a failure occurs, ASM requires free space in a disk group to re-create lost data extents in order to preserve redundancy. The amount of free space required depends on the amount of storage affected by the failure. Oracle recommends that customers consider the possibility of losing an entire cell when determining the amount of free space which is usually maintained. Note that the ASM redundancy level (**HIGH** or **NORMAL**), and Exadata model being used can have a profound bearing on the amount of free space which is required to maintain ASM redundancy.

For example, on a Full Rack Database Machine, with 14 cells, the failure of a cell requires the contents of the failed cell to be redistributed across the 13 surviving cells. This means that 1/14 (a little over 7%) of the overall capacity needs to be reserved as free space in order to preserve redundancy if a cell is lost.

By contrast, on a Quarter Rack Database Machine, with only 3 cells, you require at least 1/3 of the total capacity to be free to preserve **NORMAL** redundancy if one cell becomes unavailable.

Furthermore, on a Quarter Rack Database Machine using **HIGH** redundancy, it is impossible to preserve redundancy if a cell is lost since a Quarter Rack only contains three cells to start with. However in this case you could choose to continue operations with the remaining two cells until the third one is replaced.

Quorum Disks on Database Servers

Quorum disks on database servers enable high redundancy for the cluster voting disks on smaller Exadata systems.

- Use Oracle Exadata Deployment Assistant to automatically configure new systems:

Cluster Definition:

Cluster Configuration:

Installation Template:

Disk Group	Redundancy	Volume Size	OCR/Vote	Quorum
DBFS_DG	NORMAL	436G	No	No
DATA_EIDM	HIGH	43740G	Yes	Yes
RECO_EIDM	NORMAL	65610G	No	No

- Use the `quorумdiskmgr` utility to manually manage quorum disks.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware requires 5 voting disks on different failure groups when using a high redundancy disk group to store the cluster voting disks. Consequently, in earlier releases, the voting disks are always created on a normal redundancy disk group on Exadata clusters with 4 or fewer storage servers. In such cases, if two cells become unavailable, the cluster is forced to shut down, even if the data is being protected in other high redundancy disk groups.

Quorum disks enable users to leverage the disks on database servers to achieve higher availability in smaller Exadata configurations. Quorum disks are created on the database servers and can be used to store cluster voting disks, the cluster registry (OCR), and database server parameter files (spfiles).

For new systems, Oracle Exadata Deployment Assistant automatically configures quorum disks during deployment if the required conditions are met. The main requirements are:

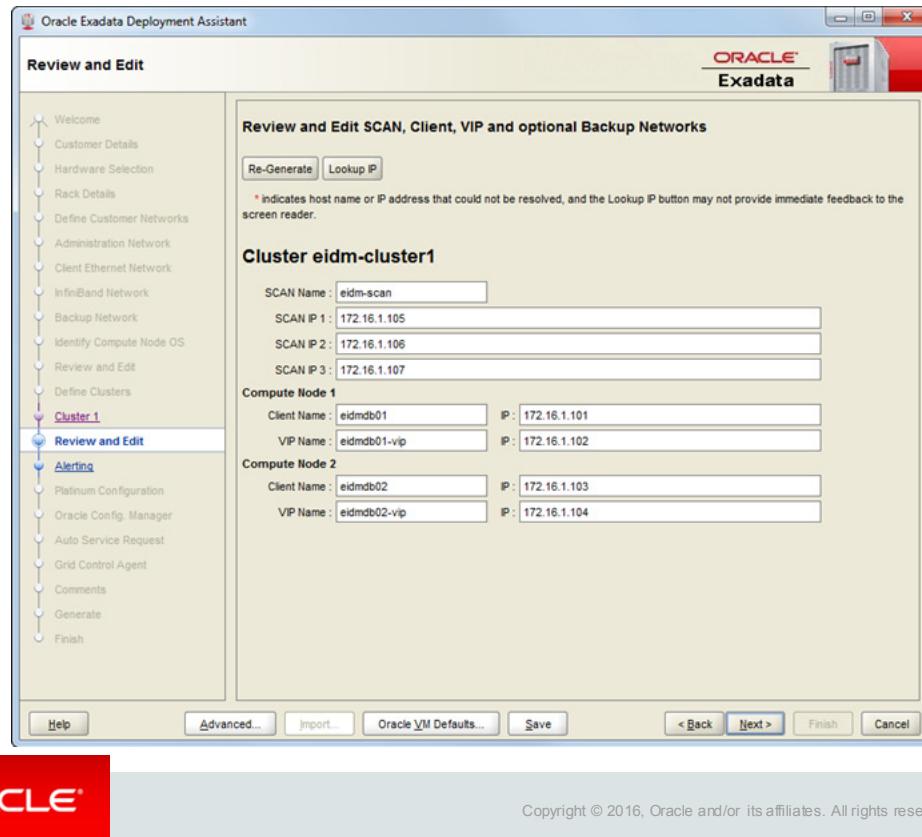
- You are configuring an Exadata cluster with 4 or fewer storage servers.
- High redundancy is specified for the DATA or RECO disk group (or both).
- The minimum required software versions are being used, including Exadata Storage Server release 12.1.2.3.0 and Grid Infrastructure version 12.1.0.2.160119.

If the required conditions for automatic configuration are met, the installation template generated by Oracle Exadata Deployment Assistant will show an indication similar to the example shown in the slide.

For existing systems, use the `quorumdskmgr` utility to manually manage quorum disks on database servers. With this utility, you can create, list, delete, and alter quorum disk configurations, targets, and devices.

See *Managing Quorum Disks Using the Quorum Disk Manager Utility* and *Adding Quorum Disks to Database Servers* in the *Oracle Exadata Database Machine Maintenance Guide* for details.

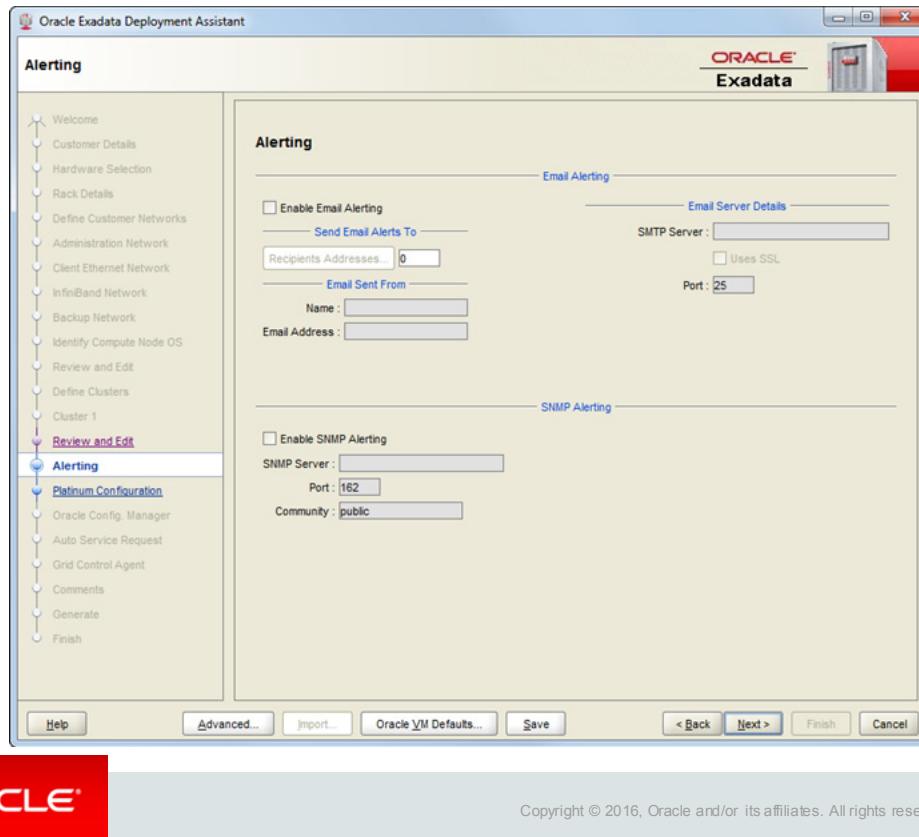
Exadata Configuration Tool: Review and Edit



The slide shows an example of a completed "Review and Edit" page. The page contains SCAN, client, VIP and backup network configuration information based on the settings supplied in the previous pages. You can use this page to make manual modifications to specific names or IP addresses. However, this is not recommended since the default IP allocation order and naming convention is well understood by Oracle service and support engineers and any changes could lead to confusion if the defaults are later assumed. Also, beware that if the default addressing policy is modified, care must be taken to ensure that the resulting addresses are valid and there are no duplicates.

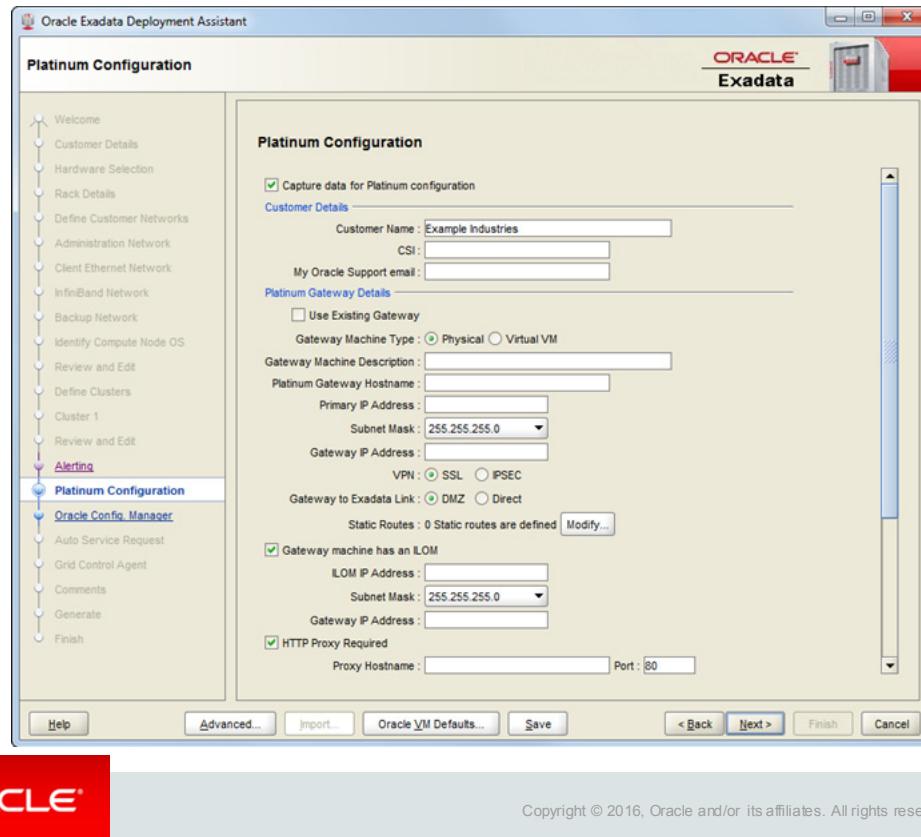
You can undo any manual changes by using the Re-Generate Data button. The Lookup IP button performs a basic check to determine if any of the IP addresses listed on the page already exist in your network.

Exadata Configuration Tool: Alerting



Exadata cell alerts can be delivered to administrators using Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP), or both. Cell alert delivery may be configured during the initial configuration process or any time afterwards. To configure cell alert delivery as part of initial configuration, complete the Alerting page shown on the slide. Configuration is optional.

Exadata Configuration Tool: Platinum Configuration

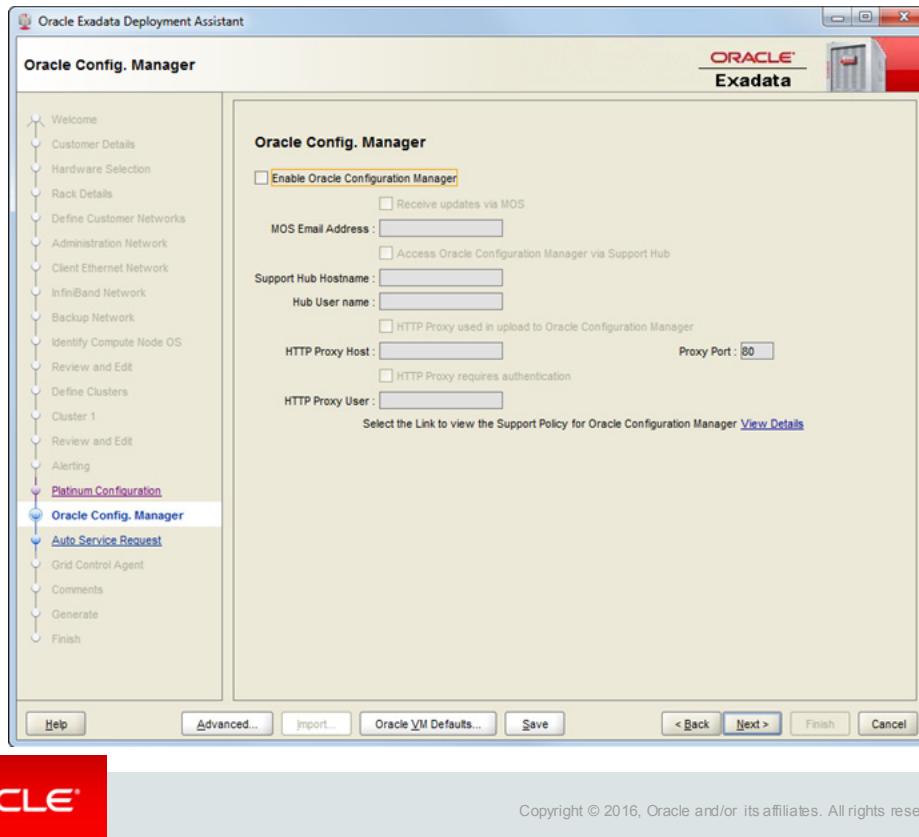


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle Platinum Services is a special support services package that provides customers with additional services such as remote fault monitoring, accelerated response times and patch deployment services. In order to use Oracle Platinum Services, customers must configure a platinum support gateway to enable remote monitoring, restoration and patching services.

Configuration of Oracle Platinum Services is optional but recommended. The slide shows a blank Platinum Configuration page. The page captures the required configuration attributes for Oracle Platinum Services.

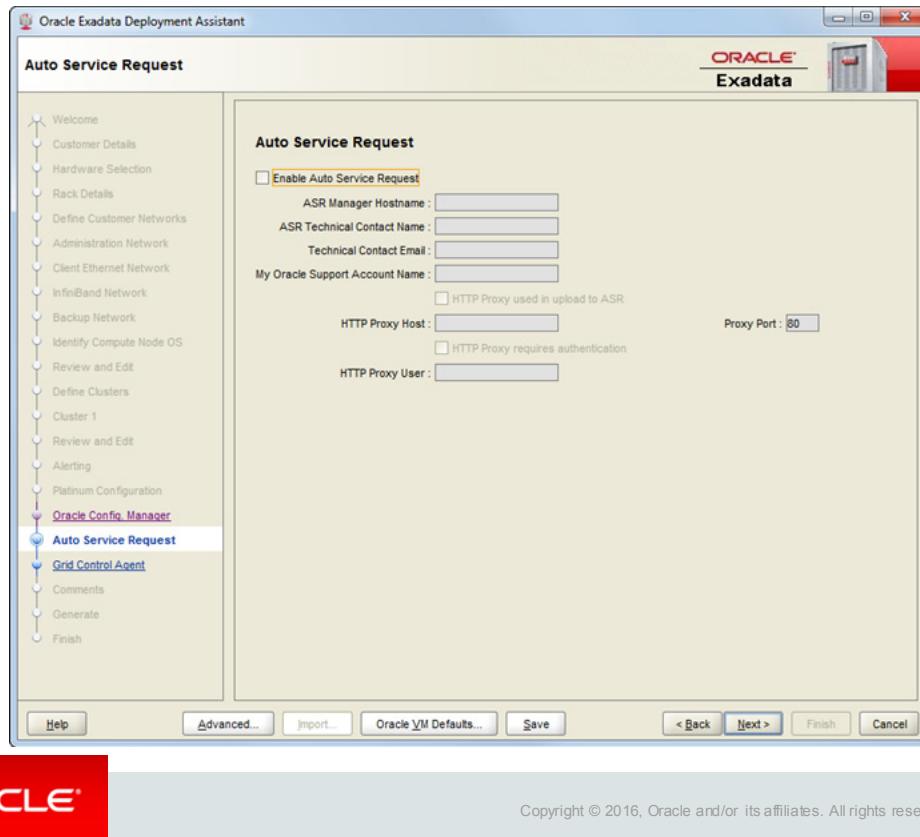
Exadata Configuration Tool: Oracle Configuration Manager



Oracle Configuration Manager (OCM) automatically collects configuration information from your environment at regular intervals. This configuration information can be uploaded to My Oracle Support. This helps Oracle to maintain up-to-date information about your environment, diagnose support issues more efficiently, and offer consistently better support outcomes.

OCM configuration in conjunction with Exadata is optional but recommended. The slide shows a blank Oracle Configuration Manager configuration page. The page captures the important configuration attributes for OCM. Further information regarding OCM configuration and operation is covered later in the course in the lesson titled “Exadata Database Machine Automated Support Ecosystem.”

Exadata Configuration Tool: Auto Service Request

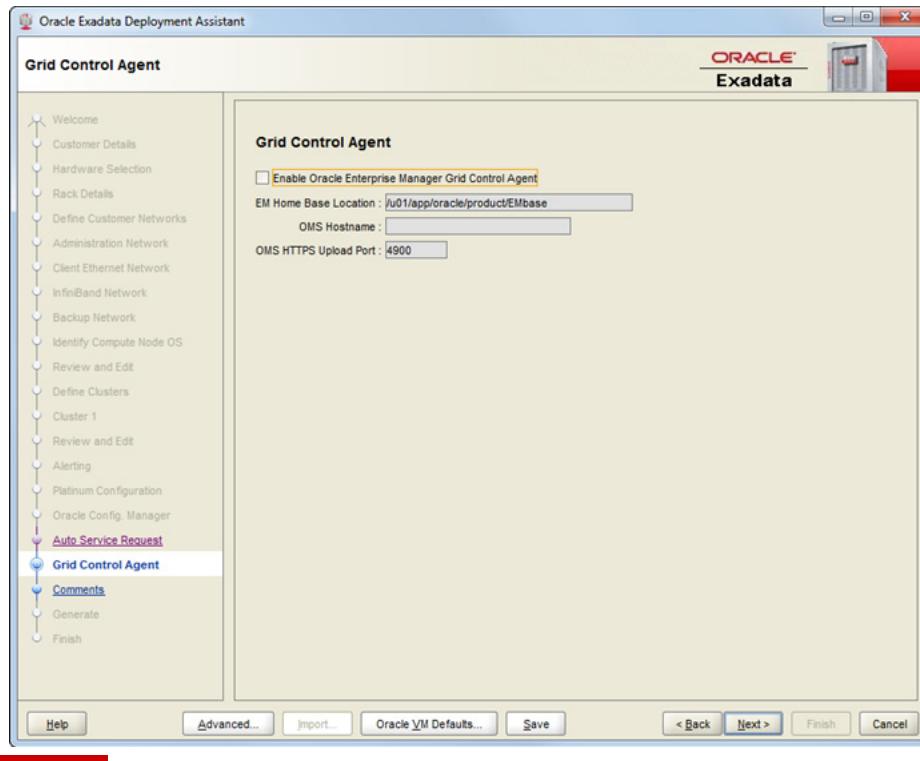


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Auto Service Request (ASR) automatically opens service requests (SRs) with Oracle Support when specific hardware faults occur either in the Exadata Storage Servers or the database servers.

ASR configuration in conjunction with Exadata is optional but recommended. The slide shows a blank Auto Service Request configuration page. The page captures the important configuration attributes for ASR. Further information about ASR configuration and operation is covered in the lesson titled *Exadata Database Machine Automated Support Ecosystem*.

Exadata Configuration Tool: Grid Control Agent



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

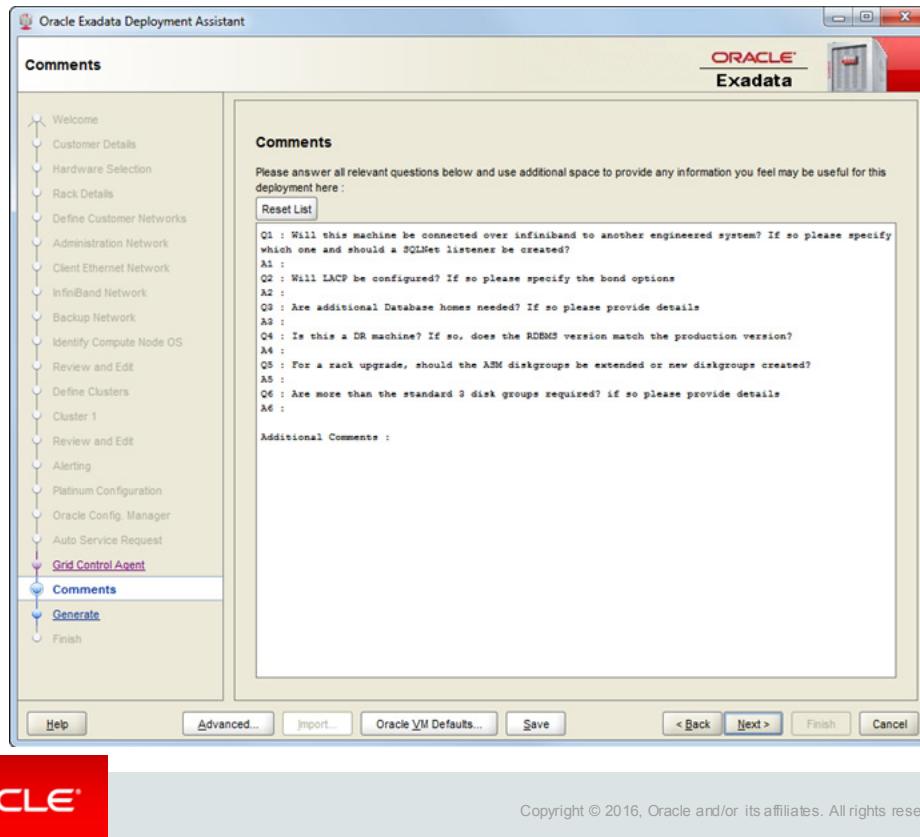
Enterprise Manager is the recommended monitoring environment for Exadata. The Grid Control Agent page shown in the slide facilitates the collection of information required to configure Enterprise Manager agents on the Exadata database servers. Configuration is optional.

If agent configuration is not performed in conjunction with initial Exadata installation and configuration, it can easily be performed later as a separate task.

The configuration and use of Enterprise Manager in conjunction with Exadata is covered in detail in a series of lessons later in the course.

Note that even though the page is named Grid Control Agent, the resulting agent installation is compatible with Oracle Enterprise Manager Cloud Control.

Exadata Configuration Tool: Comments

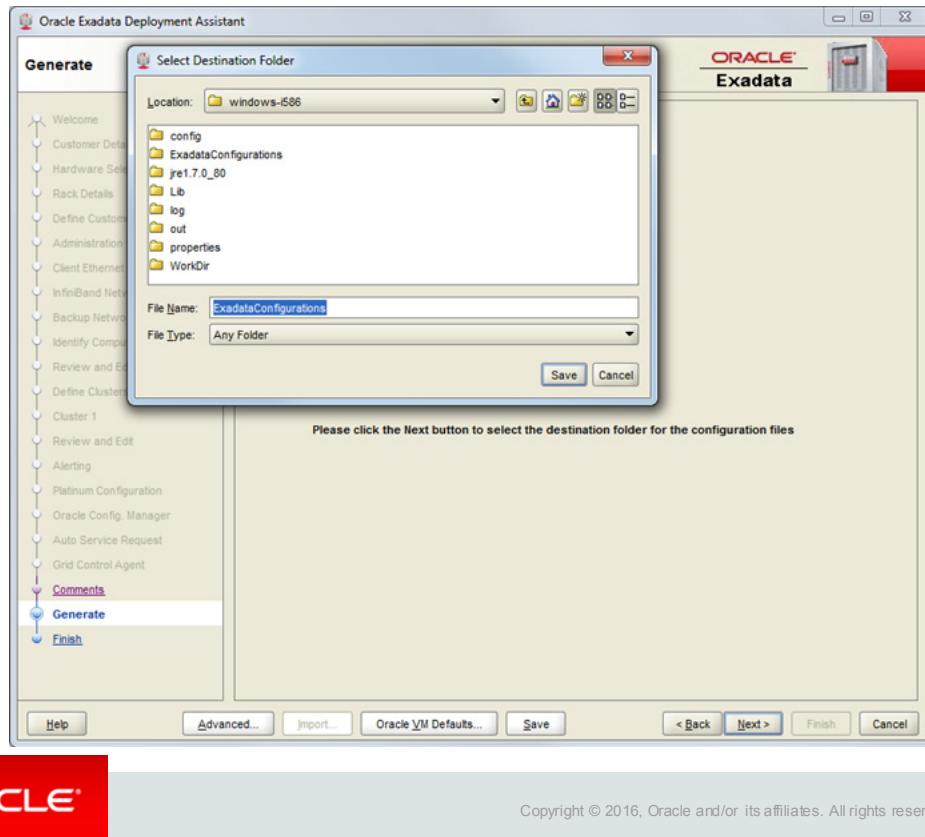


Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Use the Comments page to capture free-format text that documents interesting specifics about your configuration.

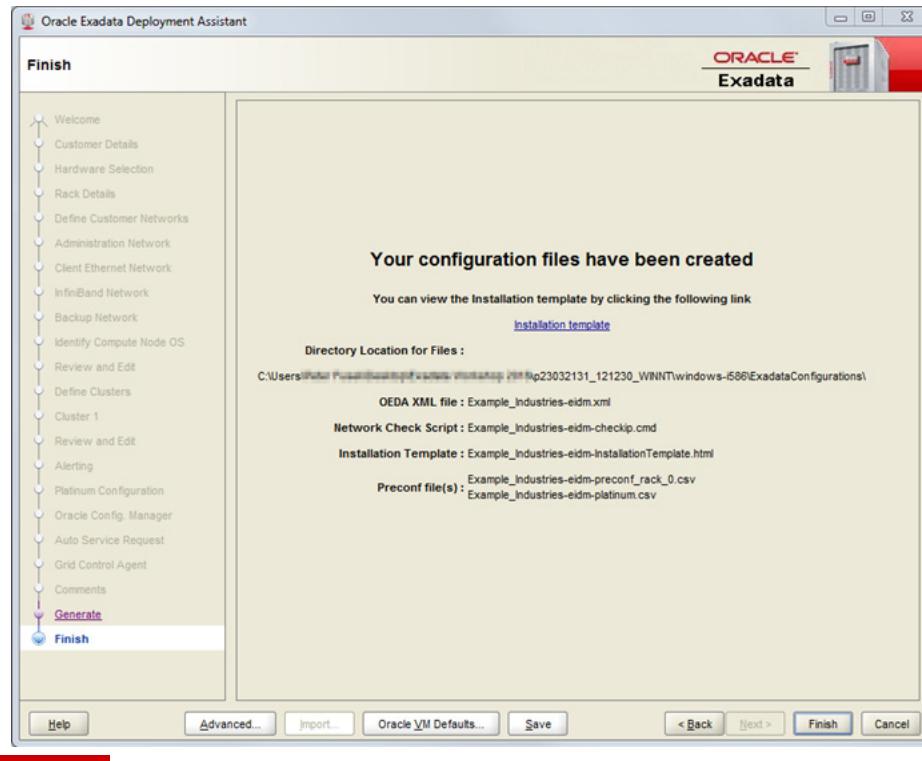
The comments field is also pre-configured with a series of questions relating to common situations that require additional manual configuration of Exadata or supporting infrastructure, such as network switches and firewalls.

Exadata Configuration Tool: Generate



The Generate page invites you to select the location where the Exadata configuration tool writes the files that drive the Exadata installation and configuration processes.

Exadata Configuration Tool: Finish



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The final page in the Exadata configuration tool shows the location of the generated configuration files that can be used to drive the Exadata installation and configuration processes.

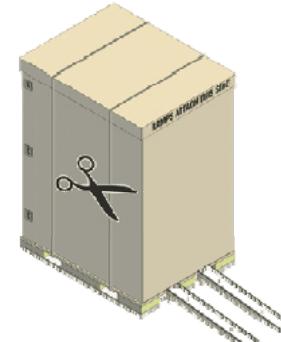
In addition to the configuration files, the Exadata configuration tool generates a network check script. You can use this script to check that your network configuration, particularly your DNS configuration, is ready to support Exadata.

Before you finish the assistant you can view the installation template, which is a HTML document that contains the previously captured configuration details. If you discover a problem, you can navigate back to a previous page and correct the issue. When you are satisfied, click Finish to exit the assistant. Note that a previously generated configuration can be imported into the deployment assistant for later modification if required.

Exadata Hardware Installation: Overview

Refer to the Installation and Configuration Guide for details on the following hardware installation tasks:

- Reviewing Safety Guidelines
- Unpacking Exadata
- Placing Exadata in Its Allocated Space
 - Moving Exadata
 - Stabilizing Exadata
 - Attaching a Ground Cable
- Powering On the System the First Time
 - Inspecting Exadata After it is in Place
 - Connecting Power Cords
 - Powering on Exadata



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

After site planning and preparation, but before the configuration programs can be run, the Exadata hardware must be physically installed at the site. The list on the slide outlines the recommended tasks. Chapter 4 of the Installation and Configuration Guide titled *Installing Oracle Exadata Database Machine or Oracle Exadata Storage Expansion Rack at the Site* describes the recommended tasks in detail.

Configuring Exadata: Overview

- Connect to the Network
- Configure the KVM Switch (if installed)
- Configure Sun Datacenter InfiniBand Switch 36 Switch
- Configure the Cisco Ethernet Switch
- Configure the Power Distribution Units
- Check Exadata Storage Servers
- Check Oracle Database Servers
- Perform Additional Checks and Configuration
- Verify the InfiniBand Network
- Perform Initial Elastic Configuration
- Add Additional Elastic Nodes to an Existing Rack
- **Load the Configuration Information and Install the Software**
- Install Oracle Enterprise Manager



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

After the Exadata hardware is physically installed, initial configuration can be undertaken. Initial configuration uses the information gathered in the Exadata configuration tool to configure the Exadata hardware and software. The list on the slide shows the recommended configuration steps. These are documented in the Installation and Configuration Guide chapter titled *Configuring Oracle Exadata Database Machine*.

Many of the configuration tasks in the list are mechanical in nature, involving the execution of commands and checks to prepare Exadata components. Examples of such tasks include connecting to the network and configuring the network switches.

The main Oracle software installation and configuration process occurs after the various component checks are completed. Further detail follows regarding this process.

Loading the Configuration Information and Installing the Software

Steps performed by the Exadata deployment tool:

1. Validate Configuration File
2. Update Nodes for Eighth Rack
3. Create Virtual Machine
4. Create Users
5. Setup Cell Connectivity
6. Create Cell Disks
7. Create Grid Disks
8. Configure Alerting
9. Install Cluster Software
10. Initialize Cluster Software
11. Install Database Software
12. Relink Database with RDS
13. Create ASM Diskgroups
14. Create Databases
15. Apply Security Fixes
16. Install Exachk
17. Setup ASR Alerting
18. Create Installation Summary
19. Resecure Machine



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The Exadata deployment tool is used to configure the Exadata software stack based on the information the configuration files. The slide lists the current steps (as at Sept 2015) performed by the Exadata deployment tool. These steps are subject to change as the Exadata deployment tool becomes more refined and as new configuration options are developed.

The steps are run sequentially and each step must complete successfully before the next step commences. All the steps, or a specified range of steps, can be run using a single command. Steps can be also be run individually.

If a step fails then in most cases the cause of the failure can be remedied and the process restarted by re-running the failed step. Depending on the exact nature of the problem, some failures may require additional manual effort to return the Database Machine back to a pre-failure state. The README file that accompanies the Exadata deployment tool provides further guidance on these activities, however careful planning and execution of all the installation and configuration steps by experienced personnel is key to avoiding issues.

Depending on the Exadata model and capacity, the entire process (all steps) can take hours to run.

Running the Exadata Deployment Tool

1. Log in as the `root` user on the first database server. The default password is `welcome1`.
2. Copy the configuration files produced by the Exadata configuration tool to the Oracle Exadata Deployment Assistant directory.
 - Typically located under `/u01/oeda` or `/u01/onecommand`
3. Download all necessary Oracle Exadata Storage Server Software and Oracle Database patches.
 - See My Oracle Support note 888828.1 for details.
4. Apply updates for the Oracle Exadata Deployment Assistant.
 - See associated README file for instructions.
5. Change to the Oracle Exadata Deployment Assistant directory.
6. Run the Exadata deployment tool using the following command:

```
# ./install.sh -cf <XML configuration file>
[ -s <n> | -r <n> <N> | -l ]
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

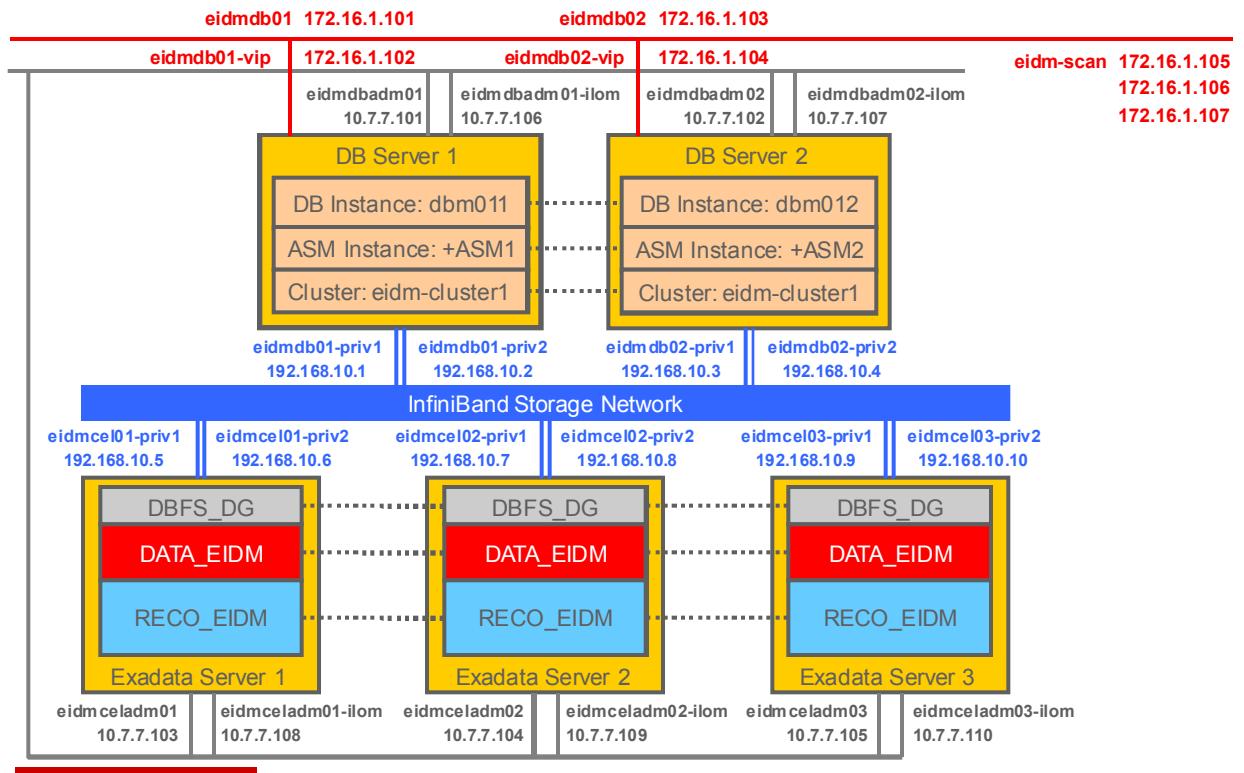
The slide outlines the procedure required to run the Exadata deployment tool. Note that the process uses a script that is run from the first database server; that is the lowest database server in the rack, which is rack position U16. You must transfer to this server the previously generated configuration files created using the Exadata configuration tool.

Prior to running the Exadata deployment tool, all necessary patches and updates should be loaded on to the server. My Oracle Support note 888828.1 is the primary reference to consult for details. You should ensure that you are using the latest Exadata deployment tool by applying available updates for the Oracle Exadata Deployment Assistant.

To run the Exadata deployment tool, change to directory containing the Exadata deployment tool and run `install.sh` using the `-cf` option to specify the XML configuration file generated by the Exadata configuration tool. For example:

- To run all of the installation steps use the following command:
`# ./install.sh -cf Example_Industries-eidm.xml`
- To run a specific step, for example step 13, use the `-s` option:
`# ./install.sh -cf Example_Industries-eidm.xml -s 13`
- To run a range of steps, for example steps 6 to 8, use the `-r` option:
`# ./install.sh -cf Example_Industries-eidm.xml -r 6 8`
- To list all the available steps use the `-l` option:
`# ./install.sh -cf Example_Industries-eidm.xml -l`

Result After Installation and Configuration



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide illustrates the result after installation and configuration based on the Exadata configuration tool examples shown earlier in this lesson.

The host names and IP addresses are based on the default settings and allocation policies. Note that each host name starts with the Exadata Database Machine name (eidm). Note also the default IP address allocation scheme.

Each database server is installed with the same operating system configuration, including the same Oracle user account and group definitions. The database servers are configured as a cluster under the control of Oracle Clusterware software. An ASM cluster is also configured. Finally, Oracle Database software is installed on each database server and a Real Application Clusters (RAC) database is established across all the cluster nodes.

Supported Additional Configuration Activities

- Earthquake protection using a third-party Seismic Isolation Platform



- Replace the Ethernet switch.
- Connect a tape library for backup.
- Customize the storage configuration.
- Create and configure databases.
- Configure database features.
 - Oracle Data Guard
 - Database File System (DBFS)
- Configure Enterprise Manager.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide lists some typical configuration activities which customers can undertake using their own labor and at their own expense. The following points expand on some of these:

- Customers are permitted to implement earthquake protection with a third-party Seismic Isolation Platform as long as the Database Machine is not physically altered or re-racked in an unsupported way.
- Customers are permitted to replace the Cisco Ethernet switch with an equivalent switch of their choice. Alternatively, customers can implement third-party gateway switches to isolate Exadata from other components on the network.
- Customers are permitted to connect additional servers or devices to Exadata via Ethernet or InfiniBand. Typically, customers require such connections to connect their backup and recovery infrastructure or to facilitate data transfer from other systems.

Hardware Re-Racking

Hardware re-racking is supported subject to the following:

- Only Half Racks and smaller can be re-racked
 - Re-racking of Full Racks is not supported
- The Exadata Hardware Re-rack Service must be purchased in addition to the standard installation service.
 - Re-racking occurs prior to installation
- The customer must supply a suitable target rack.
 - The customer may also need to supply alternative PDUs
 - See your Oracle representative for detailed specifications
- The original component layout must not be changed.
- No additional equipment may be installed.
- A re-racked Database Machine may be upgraded.
 - But only up to a Half Rack



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Customers sometimes wish to re-rack a Database Machine to comply with a data center policy, to achieve earthquake protection or to overcome a physical limitation of some sort. However, re-racking potentially opens the possibility for problems associated with component assembly and cabling errors, component damage, thermal management issues, cable management issues and other issues. As a result, there is specific policy regarding hardware re-racking for Exadata.

Hardware re-racking is supported subject to the following requirements and limitations:

- Only Half Racks, Quarter Racks and Eighth Racks may be re-racked. Re-racking of Full Racks is not supported.
- The customer must purchase the Exadata Hardware Re-rack Service in addition to the standard installation service. Re-racking must take place prior to installation.
- The customer must supply a suitable alternative rack. If the proposed rack is smaller than 42 RU (Rack Units) in height, then the customer must also supply suitable alternative Power Distribution Units. Detailed specifications are available from your Oracle representative.
- Due to the custom cable harnesses inside each Database Machine, the original component layout must not be changed during re-racking.

- No additional equipment may be installed in the target rack.
- A re-racked Eighth Rack can be upgraded to a Quarter Rack and a re-racked Quarter Rack can be upgraded to a Half Rack. However, a re-racked Half Rack cannot be upgraded to a Full Rack.

Unsupported Configuration Activities

- Adding components to servers
- Swapping Linux distributions



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The following configuration activities are not supported with Exadata:

- **Adding components to servers:** Customers sometimes wish to add components to Exadata servers. A typical example is the desire to add a Host Bus Adapter (HBA) to the database servers so that they can be attached to existing SAN storage. Adding components to Exadata servers is not supported because of the potential for driver and firmware incompatibilities that could undermine the system.
- **Swapping Linux distributions:** Oracle Linux is provided as the operating system underpinning the Exadata database servers and Exadata Storage Servers. Swapping Linux distributions is not supported.

Note: Consult your Oracle representative to confirm whether other configuration activities are supported or not.

Quiz



Using the Exadata configuration tool, you can set specific IP addresses for each Exadata database server and Exadata cell.

- a. True
- b. False

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a

Initially, the Exadata configuration tool network configuration pages only capture the first in a range of IP addresses for the different Exadata networks. However, individual IP addresses can be adjusted on the “Review and Edit” pages. Beware that if you modify the defaults you must ensure that the resulting addresses are valid and there are no duplicates. Note that while the Exadata configuration tool is quite flexible and allows customers many opportunities for customization, it is highly recommended that customers should stick to the default conventions wherever possible. This not only reduces the possibility for errors or misconfigurations, but it also makes the resulting configuration easier to support.

Quiz



Which of the following options for connecting to existing SAN storage are supported in conjunction with Exadata?

- a. Install a fiber channel HBA into each database server.
- b. Use a server connected to the existing SAN as a storage gateway and connect it to Exadata using NFS over Ethernet.
- c. Use a server connected to the existing SAN as a storage gateway and connect it to Exadata using NFS over InfiniBand.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b, c

Connecting Exadata to other servers using Ethernet or InfiniBand is supported. Adding hardware components to Exadata servers is not supported.

Summary

In this lesson, you should have learned how to describe:

- Installation and configuration process for Exadata
- Default configuration for Exadata
- Supported and unsupported customizations for Exadata



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 5 Overview: Using the Exadata Configuration Tool

In this practice, you are introduced to the Exadata configuration tool. You will use it to generate a set of configuration files for an example Exadata implementation scenario.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

6

Exadata Storage Server Configuration

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Configure Exadata Storage Server software
- Create and configure ASM disk groups using Exadata storage



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Storage Server Administration: Overview

- Each Exadata Storage Server is administered individually.
- Most administration is performed using CellCLI.
 - CellCLI can be executed only on the cell being administered.
 - CellCLI works in conjunction with MS to perform administration tasks.
 - CellCLI session example:

```
[celladmin@exalcel01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production ...

CellCLI> list cell
    exalcel01    online

CellCLI> exit
quitting

[celladmin@exalcel01 ~]$
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Inside Exadata Database Machine, each Exadata Storage Server runs independently from all the other Exadata Storage Servers. In line with that autonomy, each cell is administered individually. Most administration functions are performed using the Exadata cell command-line interface (CellCLI). CellCLI can be used only from within a cell to manage that cell. However, you can run the same CellCLI command remotely on multiple cells with the `dcli` utility, which is described later in this lesson.

CellCLI works in conjunction with the Exadata Storage Server Management Server (MS). CellCLI provides the command interface while MS performs the administrative functions, such as creating and dropping grid disks.

Exadata Storage Server Administrative User Accounts

Three operating system users are configured for each Exadata Storage Server:

- The `root` user can:
 - Edit configuration files such as `cellinit.ora` and `cellip.ora`
 - Change network configuration settings
 - Run support and diagnostic utilities located under the `/opt/oracle.SupportTools` directory
 - Run the CellCLI `CALIBRATE` command
 - Perform all the tasks that the `celladmin` user can perform
- The `celladmin` user can:
 - Perform administrative tasks (CREATE, DROP, ALTER, and so on) using the CellCLI utility
 - Package incidents for Oracle Support using the `adrci` utility
- The `cellmonitor` user can only view (LIST, DESCRIBE) Exadata cell objects using the CellCLI utility.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Three operating system users are configured for each Exadata Storage Server: `root`, `celladmin`, and `cellmonitor`. The slide describes the function of each user account.

After Exadata is initially configured, the operating system user accounts are set with initial passwords. The default initial password for `root` is `welcome1`. The default initial password for the `cellmonitor` and `celladmin` users is `welcome`. It is recommended that the initial passwords for all the user accounts should be changed to more secure passwords after initial configuration is completed.

Exadata Storage Server Users, Roles, and Privileges

In addition to the default administrative user accounts, you can:

- Create users: `CellCLI> create user <username> password = *`
- Create roles : `CellCLI> create role <rolename>`
- Grant privileges on objects to roles:
`CellCLI> grant privilege <action> on <object> ... to role <rolename>`
 - Actions include: alter, create, describe, drop, export, import, list, all actions
 - Objects include: cell, celldisk, flashcache, flashlog, griddisk, role, all objects
- Grant roles to users:
`CellCLI> grant role { all | <rolename> } to user { all | <username> }`



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In addition to the default administrative user accounts, Exadata Storage Server enables you to precisely control cell privileges through a system of users and roles. For example, you can specify that a user can run the LIST GRIDDISK command but not ALTER GRIDDISK.

The slide lists the operations that you can perform, along with an outline of the associated CellCLI commands. For full command syntax, refer to the *Oracle Exadata Storage Server Software User's Guide*.

Exadata Storage Server Users, Roles, and Privileges: Examples

```
CellCLI> create role administrator
CellCLI> grant privilege all actions on all objects to role
      administrator
CellCLI> create user celladministrator password=*
CellCLI> grant role administrator to user celladmininstrator
```

```
CellCLI> create role monitor
CellCLI> grant privilege list on all objects to role monitor
CellCLI> create user cellmon password=*
CellCLI> grant role monitor to user cellmon
```

```
CellCLI> create role gdsk
CellCLI> grant privilege create on griddisk to role gdsk
CellCLI> grant privilege alter on griddisk to role gdsk
CellCLI> grant privilege list on griddisk to role gdsk
CellCLI> create user dskman password=*
CellCLI> grant role gdsk to user dskman
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide contains three examples where a user is created and granted privileges through a role.

In the first example, a user named `celladministrator` is granted all administrative privileges on all object types through a role named `administrator`. Consequently, the `celladministrator` user has administrative capabilities that are similar to the built-in `celladmin` user.

In the second example, a user named `cellmon` is granted the ability to view (LIST) all objects through a role named `cellmon`. Consequently, the `cellmon` user has administrative capabilities that are similar to the built-in `cellmonitor` user.

In the third example, a user named `dskman` is granted the ability to create, alter and view grid disks through a role named `gdsk`.

Running CellCLI Commands from Database Servers

ExaCLI enables cell management from remote hosts:

- Enables tighter control to cell access:
 - No SSH connection to the cell is required
 - Communication uses https
- Requires access through a user with role-based privileges
- Supports CellCLI command syntax, with some restrictions
- Command syntax:

```
$ exacli -c [username@]host [-l username] [--xml]
    [--cookie-jar [filename]] [-e CellCLI_command]
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata software release 12.1.2.2.0 introduces the ExaCLI utility. ExaCLI is essentially the same as CellCLI; however, the main difference is that ExaCLI manages cells from a remote host, typically an Exadata database server, whereas CellCLI runs directly on a cell.

ExaCLI enables users to perform most management functions without the requirement to establish an SSH connection to the cell. Consequently, access to the cell operating system can be tightly controlled to address various operational and compliance requirements.

To use ExaCLI, non-default users must be created on the cell, and the users must be assigned roles that grant appropriate privileges. Creation of users and roles can only be performed directly on the cell using CellCLI.

All communication between ExaCLI and Management Server (MS) running on the cell is over https. Security certificates allow the cells to confirm their identity to ExaCLI. MS is deployed with a default self-signed security certificate, or you can upload a different security certificate issued by a Certificate Authority (CA).

ExaCLI supports the same command syntax as CellCLI. However, not all CellCLI commands can be executed through ExaCLI. Unsupported commands include:

- ALTER CELL with the RESTART, STARTUP or SHUTDOWN options
- CREATE USER, ALTER USER, DROP USER
- CALIBRATE

- DESCRIBE
- HELP
- SET
- SPOOL
- START and @
- Comments using REMARK

Also, ExaCLI cannot get the cell attributes rsStatus, cellsrvStatus, msStatus using the LIST CELL command.

The slide outlines the format of the ExaCLI command. At a minimum, you must specify a cell host name and a cell user.

If you supply the -e option followed by a CellCLI command, the command is executed immediately and then the ExaCLI session ends. If you do not specify the -e option and a command, an interactive session is started.

You can use the --xml option to receive command output in XML format.

ExaCLI can use cookies for authentication to the cell. A cookie enables a user to execute an ExaCLI command without supplying a password for a period of 24 hours from when the cookie is issued. The --cookie-jar option specifies the name of the file where the cookies are stored. If no file name is specified, the cookies are stored in a default cookie jar located at \$HOME/.exacli/cookiejar, where \$HOME is the home directory of the operating system user running the ExaCLI command.

If you specify the --cookie-jar option, the cookie jar is examined and if a valid cookie is found the command proceeds without a password prompt. If a valid cookie is not found, the user is prompted for a password and on successful authentication the cell issues a cookie that is stored in the cookie jar for subsequent use.

If you do not specify the --cookie-jar option, the default cookie jar is still examined and used if a valid cookie is found. However, if a valid cookie does not exist, the new cookie will not be stored in the default cookie jar if the --cookie-jar option is not specified.

ExaCLI: Examples

```
$ exacli -l celladministrator -c exa1cel01  
Password=*****  
ExaCLI>
```

```
$ exacli -c celladministrator@exa1cel01  
Password=*****  
ExaCLI>
```

```
$ exacli -c celladministrator@exa1cel01 --cookie-jar -e list cell  
Password=*****  
  exa1cel01  online  
  ...
```

```
$ exacli -l celladministrator -c exa1cel01 -e list celldisk  
  CD_00_exa1cel01    normal  
  ...
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide contains four ExaCLI command examples. In all cases, it is assumed that the `celladministrator` user has been previously created on the cell `exa1cel01`.

The first two command examples are functionally identical. They both start interactive command sessions on `exa1cel01` as the `celladministrator` user. In both cases, ExaCLI prompts for a password if the default cookie jar does not contain a valid cookie. Because the `--cookie-jar` option is not specified, the new cookie that is generated during the connection is not stored in the cookie jar.

In the third example, the `LIST CELL` command is executed on `exa1cel01` as the `celladministrator` user. As with the first two examples, ExaCLI prompts for a password if the default cookie jar does not contain a valid cookie. However, because the `--cookie-jar` option is specified, the new cookie that is generated during the connection is saved in the default cookie jar.

In the final example, the `LIST CELLDISK` command is executed on `exa1cel01` as the `celladministrator` user. This time the example assumes that a valid cookie exists in the default cookie jar. Consequently, there is no password prompt and the command output is displayed immediately after the command.

Executing Commands Across Multiple Servers Using `dcli`

The `dcli` utility allows you to simultaneously execute a command on multiple Exadata servers:

- Command types:
 - Operating system commands
 - CellCLI commands
 - Operating system scripts
 - CellCLI scripts
- Commands are executed in separate parallel threads.
- Interactive sessions are not supported.
- Python 2.3 and SSH user-equivalence are required.
- Command output is collected and displayed in the terminal session executing the `dcli` utility.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The `dcli` utility facilitates centralized management across Exadata by automating the execution of a command on a set of servers and returning the output to the centralized management location where `dcli` was run. The types of commands supported by `dcli` includes operating system commands, CellCLI commands, operating system scripts, and CellCLI scripts. However, it does not support interactive sessions.

The `dcli` utility runs commands on multiple servers in parallel threads. By default, the `dcli` utility is located at `/opt/oracle/cell/cellsrv/bin/dcli` on each Exadata Storage server and at `/usr/local/bin/dcli` on each database server. You can also copy the `dcli` utility to a server outside of Exadata and launch commands from that server.

The `dcli` utility requires Python version 2.3 or later on the server running `dcli`. You can determine the version of Python by running the `python -V` command. In addition, `dcli` requires prior setup of SSH user-equivalence between all the servers. You can use the `dcli` utility initially with the `-k` option to set up SSH user-equivalence between a group of servers.

Command output (to `stdout` and `stderr`) is collected and displayed after command execution is finished on all the specified servers. The `dcli` options allow command output to be abbreviated to filter output, such as removing messages showing normal status.

dcli: Examples

```
$ dcli -g mycells date  
exalcel01: Sun Oct 30 20:48:09 CDT 2015  
exalcel02: Sun Oct 30 20:48:09 CDT 2015
```

```
$ dcli -c exalcel01,exalcel02 cellcli -e list cell  
exalcel01: exalcel01      online  
exalcel02: exalcel02      online
```

```
$ dcli -g mycells -x cellclicommands.scl
```

```
$ dcli -g mydbservers -l root -x dbwork.sh
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows a series of examples using the `dcli` utility.

The first example shows `dcli` being used to execute the operating system `date` command. The `-g` option specifies a file named `mycells`, which contains the list of target servers to which the `date` command is sent. The servers can be identified by host names or IP addresses. The servers can be database servers or Exadata Storage Servers.

The second example uses the `-c` option to specify the target servers (`exalcel01,exalcel02`) on the command line. It invokes CellCLI to report the cell status.

The third example uses the `-x` option to specify a command file. The command file must exist on the server executing the `dcli` utility. The command file is copied to the target servers and executed. A file with the `.scl` extension is run by the CellCLI utility on the target server. A file with a different extension is run by the operating system shell on the target server. The file is copied to the default home directory of the user on the target server. Files specified using the `-x` option must have execute privileges otherwise `dcli` will report an error.

The final example adds the use of the `-l` option to specify the user to connect as on the remote servers. By default, the `dcli` utility connects as the `celladmin` user.

For more information, refer to the chapter titled *Using the dcli Utility* in the *Oracle Exadata Storage Server Software User's Guide*.

Executing Commands Across Multiple Servers Using `exadcli`

The `exadcli` utility allows you to simultaneously execute a command on multiple Exadata Storage Servers:

- Command types:
 - ExaCLI commands
 - ExaCLI scripts
- Commands are executed in separate parallel threads using ExaCLI:
 - Requires access through a user with role-based privileges
 - Uses the same cookie jar mechanism as ExaCLI
- Interactive sessions are not supported.
- Command output is collected and displayed in the terminal session executing the `exadcli` utility.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata software release 12.1.2.2.0 introduces the `exadcli` utility. Like `dcli`, `exadcli` facilitates centralized management across Exadata by automating the execution of a command on a set of cells and returning the output to the centralized management location where `exacli` was run.

The main difference is that `dcli` uses SSH for communication, while `exadcli` uses https in the same way as ExaCLI. By removing the requirement for SSH, `exadcli` enables tight control over access to the cell operating system, which is useful in addressing various operational and compliance requirements.

Another difference is that `exadcli` can only be used to issue ExaCLI commands to be run on multiple cells. Unlike `dcli`, `exadcli` cannot be used to execute other commands, such as shell commands.

To use the `exadcli` utility, you must set up non-default users and roles on all the cells in the same way as required by ExaCLI. Also, `exadcli` uses the same cookie jar mechanisms as ExaCLI.

Using `exadcli`, you can run one ExaCLI command or you can run a series of commands in order. You can specify multiple commands on the command line by separating them with a semicolon (;) character, or you can specify a file that contains a list of ExaCLI commands. You cannot use `exadcli` interactively.

exadcli: Examples

```
$ exadcli -c exacel01,exacel02 -l celladministrator  
--cookie-jar -e list cell  
Password=*****  
exacel01: exacel01      online  
exacel02: exacel02      online
```

```
$ cat mycells  
exacel01  
exacel02  
  
$ cat cmd.txt  
list cell  
  
$ exadcli -g mycells -l celladministrator -x cmd.txt  
exacel01: exacel01      online  
exacel02: exacel02      online
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide two examples using the `exadcli` utility.

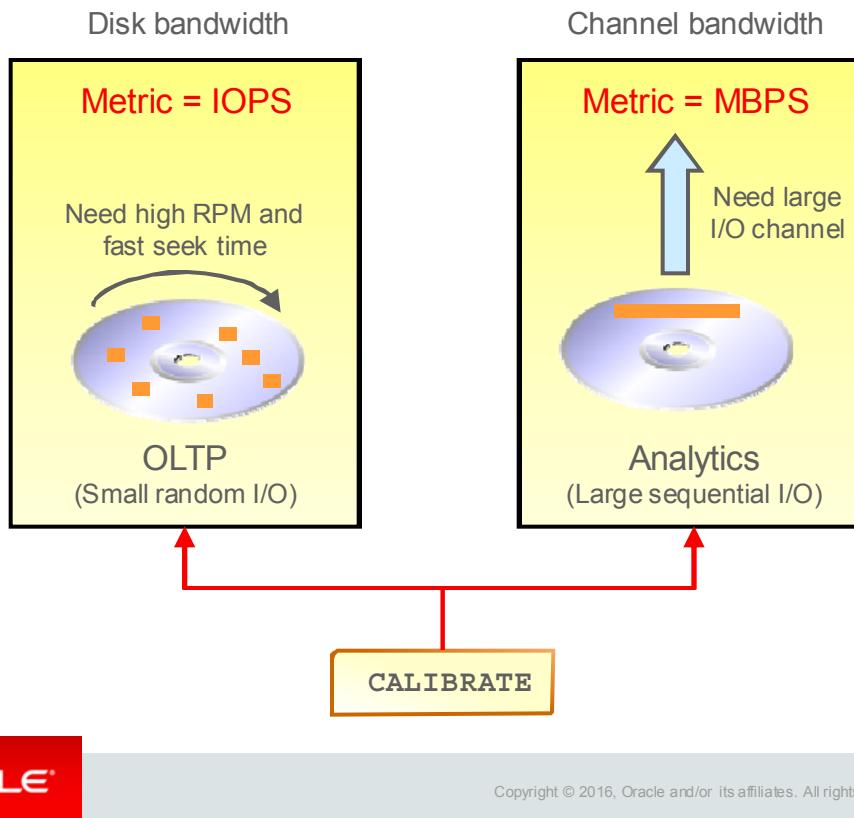
The first example shows `dcli` being used to execute the operating system `date` command. The `-g` option specifies a file named `mycells`, which contains the list of target servers to which the `date` command is sent. The servers can be identified by host names or IP addresses. The servers can be database servers or Exadata Storage Servers.

The first example runs the `LIST CELL` command on cells `exacel01` and `exacel02` as the `celladministrator` user. In this example, a valid cookie was not found in the default cookie jar so the user is prompted for the password. However, because the `--cookie-jar` option is specified, the cookies generated during the connection to each cell are saved in the default cookie jar.

The second example also runs the `LIST CELL` command on cells `exacel01` and `exacel02` as the `celladministrator` user. However, this time the target cells and the `ExaCLI` command are stored in files (`mycells` and `cmd.txt`) rather than being specified on the command line. In this example, a valid cookie is found in the default cookie jar so the user is not prompted for a password.

For more information and complete command syntax, refer to the chapter titled *Using the exadcli Utility* in the *Oracle Exadata Storage Server Software User's Guide*.

Testing Storage Server Performance by Using CALIBRATE



The CALIBRATE command runs raw performance tests on Exadata Storage Server disks and flash modules. This enables the measurement of two important metrics – IOPS and MBPS:

- **IOPS (I/O per second):** This metric represents the number of small random I/O that can be serviced in a second. The IOPS rate typically depends on how fast the disk media can spin and how many disks are present in the storage system.
- **MBPS (megabytes per second):** This metric represents the rate at which data can be transferred between the server and storage array. This mainly depends on the capacity of the I/O channel that is used to transfer the data.

A database I/O workload typically consists of small random I/Os and large sequential I/Os. Small random I/Os are more prevalent in an OLTP application environment while large sequential I/Os are common in analytics.

CALIBRATE: Example

```
[root@exalcel01 ~]# cellcli
CellCLI: Release 12.1.2.1.0 ...

CellCLI> CALIBRATE FORCE
Calibration will take a few minutes...
Aggregate random read throughput across all hard disk luns: 1604 MBPS
Aggregate random read throughput across all flash disk luns: 4242.9 MBPS
Aggregate random read IOs per second (IOPS) across all hard disk luns: 4927
Aggregate random read IOs per second (IOPS) across all flash disk luns: 148695
Controller read throughput: 1608.05 MBPS
Calibrating hard disks (read only) ...
Lun 0_0 on drive [20:0 ] random read throughput: 153.41 MBPS, and 412 IOPS
Lun 0_1 on drive [20:1 ] random read throughput: 155.38 MBPS, and 407 IOPS
...
Lun 0_8 on drive [20:8 ] random read throughput: 154.46 MBPS, and 424 IOPS
Lun 0_9 on drive [20:9 ] random read throughput: 154.63 MBPS, and 426 IOPS
Calibrating flash disks (read only, note that writes will be significantly slower)
Lun 1_0 on drive [[10:0:0:0]] random read throughput: 269.11 MBPS, and 19635 IOPS
Lun 1_1 on drive [[10:0:1:0]] random read throughput: 268.86 MBPS, and 19648 IOPS
...
Lun 5_2 on drive [[11:0:2:0]] random read throughput: 268.33 MBPS, and 19717 IOPS
Lun 5_3 on drive [[11:0:3:0]] random read throughput: 268.14 MBPS, and 19693 IOPS
CALIBRATE results are within an acceptable range.

CALIBRATE stress test is now running...
Calibration has finished.
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The CALIBRATE command enables you to verify the disk and flash memory performance. You must execute this command while being logged into the storage server as the root operating system user.

The CALIBRATE FORCE command allows you to run the tests when CELLSRV is running. If you do not use the FORCE option, CELLSRV must be shut down. Running CALIBRATE at the same time as the CELLSRV will impact performance and it is generally not recommended during normal operations. CALIBRATE FORCE is acceptable in circumstances when the cells are not running a user workload, such as during periods of scheduled maintenance.

The example in the slide shows a typical output where the results matched expectations. A different message is shown if the performance measurements are substandard.

Configuring the Exadata Cell Server Software

```
[celladmin@exacel01 ~]$ cellcli  
CellCLI: Release 12.1.2.1.0 ...  
  
CellCLI> ALTER CELL smtpServer='my_mail.example.com', -  
          smtpFromAddr='exadata.exacel01@example.com', -  
          smtpPwd=<email_address_password> -  
          smtpToAddr='jane.smith@example.com', -  
          notificationPolicy='critical,warning,clear', -  
          notificationMethod='mail'  
Cell exacel01 successfully altered  
  
CellCLI>
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

You can change or configure cell attributes by using the CellCLI `ALTER CELL` command. The slide shows an example `ALTER CELL` command that configures email notification. This facility sends email messages to the administrator (`jane.smith@example.com`) of the storage cell whenever critical, warning, and clear alerts are detected by the cell. In addition to email notification, it is possible to configure notification by using Simple Network Management Protocol (SNMP).

Starting and Stopping Exadata Cell Server Software

```
[celladmin@exa1cel01 ~]$ cellcli  
CellCLI: Release 12.1.2.1.0 ...  
  
CellCLI> ALTER CELL RESTART SERVICES ALL  
  
Stopping the RS, CELLSRV, and MS services...  
The SHUTDOWN of services was successful.  
Starting the RS, CELLSRV, and MS services...  
Getting the state of RS services...  
    running  
Starting CELLSRV services...  
The STARTUP of CELLSRV services was successful.  
Starting MS services...  
The STARTUP of MS services was successful.  
  
CellCLI>
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Occasionally, you may need to start, stop, or restart the Exadata Storage Server software services: Restart Server (RS), Management Server (MS), and Cell Server (CELLSRV). This can be achieved by using the CellCLI ALTER CELL command. An example is shown on the slide. Following is a summary of the relevant syntax:

```
ALTER CELL { SHUTDOWN | RESTART | STARTUP } SERVICES { RS | MS |  
CELLSRV | ALL }
```

Configuring Cell Disks

```
CellCLI> CREATE CELLDISK ALL HARDDISK
CellDisk CD_00_exalcel01 successfully created
...
CellDisk CD_10_exalcel01 successfully created
CellDisk CD_11_exalcel01 successfully created

CellCLI> LIST CELLDISK
    CD_00_exalcel01      normal
    ...
    CD_10_exalcel01      normal
    CD_11_exalcel01      normal
    FD_00_exalcel01      normal
    ...
    FD_14_exalcel01      normal
    FD_15_exalcel01      normal

CellCLI>
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Before the disk storage in an Exadata Storage Server can be used, cell disks must be created using the `CREATE CELLDISK` command.

Note that in most cases the initial configuration process, performed using the Oracle Exadata Deployment Assistant, will configure Exadata Storage Server cell disks and grid disks. Reconfiguring existing cell disks and grid disks is discussed later in this lesson.

The example in the slide shows the use of the `CREATE CELLDISK ALL HARDDISK` command to create 12 disk-based cell disks with default names. In most cases, the default cell disk names are used.

The preceding example also shows the use of the `LIST CELLDISK` command. For a High Capacity Exadata Storage Server, output from the `LIST CELLDISK` command shows the 12 disk-based cell disks along with 16 flash-based cell disks that are normally used in conjunction with Exadata Smart Flash Cache. The command should return a status of `normal` for all the cell disks.

Configuring Grid Disks

```

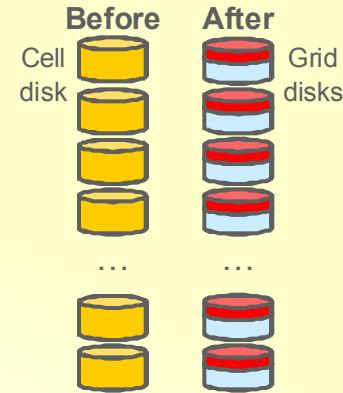
CellCLI> CREATE GRIDDISK ALL PREFIX=data, SIZE=2000G
GridDisk data_CD_00_exa1cel01 successfully created
...
GridDisk data_CD_11_exa1cel01 successfully created

CellCLI> CREATE GRIDDISK ALL PREFIX=reco
GridDisk reco_CD_00_exa1cel01 successfully created
...
GridDisk reco_CD_11_exa1cel01 successfully created

CellCLI> LIST GRIDDISK
      data_CD_00_exa1cel01      active
      ...
      data_CD_11_exa1cel01      active
      reco_CD_00_exa1cel01     active
      ...
      reco_CD_11_exa1cel01     active

CellCLI> exit
[celladmin@exa1cel01 ~] $

```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

After cell disks are created, grid disks must be provisioned using the `CREATE GRIDDISK` command. Like cell disks, grid disks are typically created as part of the initial configuration of Exadata. However, there are times when existing grid disks may be removed and replaced by new ones, such as when the disk groups are resized. This case is considered in further detail later in the lesson.

The slide shows examples of how to create grid disks on empty new cell disks. In the example in the slide, the `ALL PREFIX` option is used to automatically create one grid disk on each cell disk. When the `ALL PREFIX` option is used, the generated grid disk names are composed of the grid disk prefix followed by an underscore (`_`) and then the cell disk name.

It is best practice to use the planned ASM disk group name as the prefix name for the corresponding grid disks. In the example, prefix values `data` and `reco` are the names of planned ASM disk groups that will consume the grid disks. Grid disk names must be unique across all cells within a single Exadata deployment. By following the recommended naming conventions for naming the cell disks and grid disks, you automatically get unique names.

The optional `SIZE` attribute specifies the size of each grid disk. If omitted, the grid disk will automatically consume all the space remaining on the corresponding cell disk.

The `LIST GRIDDISK` command shows all the grid disks that are defined on the storage server.

Sparse Grid Disks

- Sparse grid disks:
 - Allocate space dynamically as data is written
 - Have a virtual size specification that can be much larger than the actual physical size
 - Useful for database files that grow and shrink over time
 - Underpin Exadata database snapshots
- Creating sparse grid disks:

```
CellCLI> CREATE GRIDDISK ... virtualsize=<size>
```

- Creating sparse disk groups:

```
SQL> CREATE DISKGROUP ... DISK '<sparse grid disks>'  
ATTRIBUTE ...  
'compatible.rdbms' = '12.1.0.2', ] or later  
'compatible.asm' = '12.1.0.2',  
'cell.sparse_dg' = 'allsparsed';
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Sparse grid disks allocate space as new data is written to the disk, and therefore have a virtual size that can be much larger than the actual physical size. Sparse grid disks can be used to create a sparse disk group.

A sparse disk group can be used to efficiently accommodate numerous database files grow and shrink over time but that consume only a small proportion of their overall size on average. Sparse disk groups are also used to quickly and efficiently create database snapshots on Oracle Exadata.

To create sparse grid disks, you must specify a virtual size in the `CREATE GRIDDISK` command.

To create a sparse disk group, you must base the disk group on a set of sparse grid disks and include the disk group attributes listed on the slide.

Note that Oracle Database 12c release 12.1.0.2 with Exadata bundle patch 3 is the minimum software requirement for using sparse grid disks.

Configuring Hosts to Access Exadata Cells

- Configuration files on each database server enable access to Exadata storage.
 - `cellinit.ora` identifies the storage network interfaces on the database server.
 - `cellip.ora` identifies the Exadata cells that are accessible to the database server.
 - Example:

```
$ cat /etc/oracle/cell/network-config/cellinit.ora
ipaddress1=192.168.10.1/24
ipaddress2=192.168.10.2/24

$ cat /etc/oracle/cell/network-config/cellip.ora
cell="192.168.10.5; 192.168.10.6"
cell="192.168.10.7; 192.168.10.8"
cell="192.168.10.9; 192.168.10.10"
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

As part of the initial configuration for Exadata, two important configuration files are created on each database server, which enable it to access Exadata Storage Servers:

- The `cellinit.ora` file contains the database server IP address that connects to the storage network. This file is host specific, and contains the IP addresses of the InfiniBand storage network interfaces for that database server. The IP addresses are specified in Classless Inter-Domain Routing (CIDR) format.
- The `cellip.ora` file contains the IP addresses for the InfiniBand storage network interfaces of the Exadata Storage Servers that are accessible to the database server.

If there are ever any issues with connectivity between a database server and Exadata Storage Servers, then verify these files to ensure that the correct settings are present. If changes are ever required to these files, then:

1. Stop the database and the Oracle ASM instances on the database server host.
2. Make the required file changes.
3. Restart the database and the Oracle ASM instances on the database server host.

Configuring ASM and Database Instances to Access Exadata Cells

- Ensure that a compatible version of the Oracle Database software is being used:
 - See My Oracle Support note 888828.1 for an up-to-date list of the supported versions for the Exadata software components.
- Set the `ASM_DISKSTRING` ASM initialization parameter:
 - `ASM_DISKSTRING= 'o/*/*'`
- Set the `COMPATIBLE` database initialization parameter:
 - `COMPATIBLE= '11.2.0.0.0'` (or later)



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

At all times, appropriate versions of the various Exadata software components must be maintained in order to ensure correct operation. This includes the version of Oracle Database software used for the ASM and database instances. Refer to My Oracle Support note 888828.1 for an up-to-date list of the supported versions for the Exadata software components.

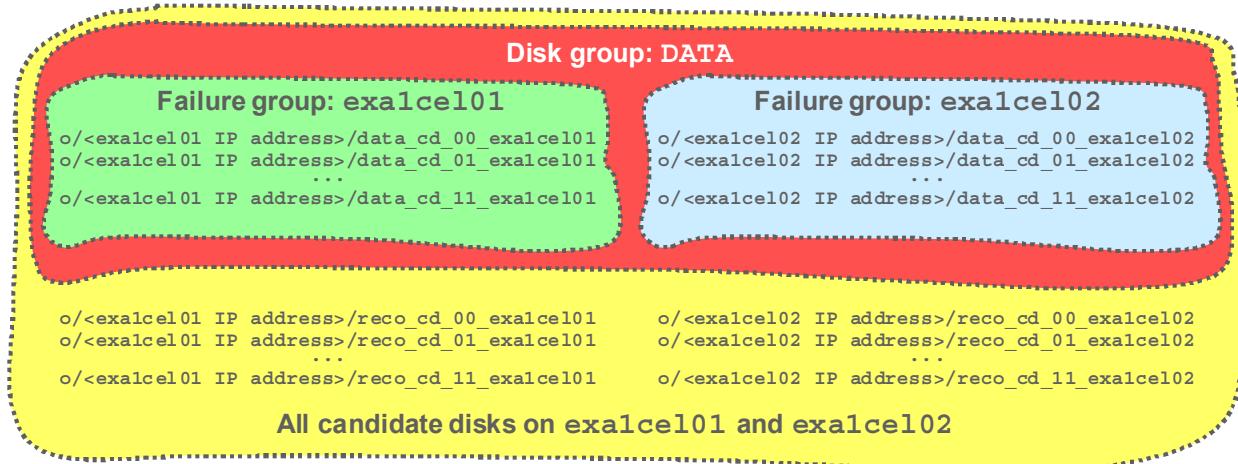
To ensure that ASM discovers Exadata grid disks, set the `ASM_DISKSTRING` initialization parameter. A search string with the following form is used to discover Exadata grid disks:
`o/<cell IP address>/<grid disk name>`

Wildcards may be used to expand the search string. For example, to explicitly discover all the available Exadata grid disks set `ASM_DISKSTRING= 'o/*/*'`. To discover a subset of available grid disks having names that begin with `data`, set
`ASM_DISKSTRING= 'o/*/*data*'`.

Note that if the `ASM_DISKSTRING` initialization parameter is not set, then the default is to discover all the currently accessible Exadata grid disks; that is, the grid disks for the cells listed in `cellip.ora` and that are not prohibited by a security setting. Exadata Storage Server security is discussed later in this lesson.

To configure a database instance to access cell storage, ensure that the `COMPATIBLE` parameter is set to `11.2.0.0.0` or later in the database initialization file.

Configuring ASM Disk Groups by Using Exadata Storage



```
CREATE DISKGROUP data NORMAL REDUNDANCY
DISK 'o/*/data*'
ATTRIBUTE 'compatible.rdbms' = '11.2.0.0.0',
           'compatible.asm' = '11.2.0.0.0',
           'cell.smart_scan_capable' = 'TRUE',
           'au_size' = '4M';
```

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Creating ASM disk groups on Exadata is essentially the same as on any other platform. However, to enable Smart Scan processing the following disk group attribute settings must be used:

```
'compatible.rdbms' = '11.2.0.0.0' (or later)
'compatible.asm' = '11.2.0.0.0' (or later)
'cell.smart_scan_capable' = 'TRUE'
```

In addition, it is recommended that you set the `AU_SIZE` disk group attribute value to 4M to optimize disk scanning.

The example in the slide shows candidate ASM disks from two Exadata cells: `exalcel01` and `exalcel02`. The `CREATE DISKGROUP` statement references all of the candidate ASM disks having names that start with `data`. By default, ASM failure groups corresponding to each cell are automatically defined. As a result, two failure groups are automatically created using corresponding grid disks from each cell. By default, the failure group names correspond to the cell names.

Once created, an Exadata-based disk group can be used to house Oracle data files in the same way as an ASM disk group defined on other storage. To complement the recommended `AU_SIZE` setting of 4 MB, you should set the initial extent size to 8 MB for large segments. This can be done using segment-level or tablespace-level settings. The recommended approaches are discussed in the lesson titled *Optimizing Database Performance with Exadata Database Machine*.

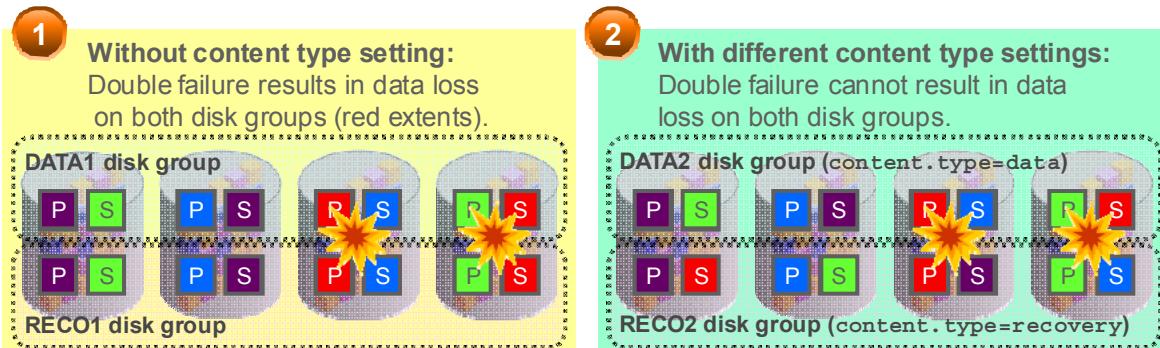
Specifying Content Type for a Disk Group

Configure the disk group attribute: `content.type`

- Possible values: `data`, `recovery` or `system`
- Configuration example:

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'content.type'='data';
SQL> ALTER DISKGROUP DATA REBALANCE POWER <power>;
```

- Decreases the likelihood that multiple failures impact disk groups with different content type settings



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

For `NORMAL` and `HIGH` redundancy ASM disk groups, the algorithm that determines the placement of secondary extents (mirror copies of data) uses an adjacency measure to determine the placement. In prior versions of ASM, the same algorithm and adjacency measure was used for all disk groups.

Commencing with Oracle Database version 11.2.0.3, ASM provides administrators with the option to specify the content type associated with each ASM disk group. This capability is provided by the `CONTENT.TYPE` disk group attribute. Three possible settings are allowed: `data`, `recovery`, or `system`. Each content type setting modifies the adjacency measure used by the secondary extent placement algorithm.

The result is that the contents of disk groups with different content type settings is distributed differently across the available disks. This decreases the likelihood that a double failure will result in data loss across multiple `NORMAL` redundancy disk groups with different content type settings. Likewise, a triple failure is less likely to result in data loss on multiple `HIGH` redundancy disk groups with different content type settings.

To illustrate, consider the diagram at the bottom of the slide. Example 1 shows two `NORMAL` redundancy disk groups, `DATA1` and `RECO1`, which are configured without the content type setting. Both disk groups use the same algorithm for placing secondary extents.

That is, the secondary extent is placed on the disk immediately to the right of the disk containing the primary extent. Where the primary extent is on the far-right disk, the secondary extent is placed on the far-left disk.

In this example, failure of the two disks at the far right results in data loss in both disk groups; that is, the red extents. In this case, it is possible that the double-failure could result in the loss of a data file and the archived log files required to recover it.

Example 2 shows NORMAL redundancy disk groups, DATA2 and RECO2, configured with different content type settings. In this example, the DATA2 disk group uses the same placement algorithm as before. However, the data placement for RECO2 uses a different adjacency measure, and because of this, the contents of RECO2 is spread differently across the disks.

In this example, failure of the two disks at the far right results in data loss only in the DATA2 disk group. However, because of the different distribution of data that is associated with the different content type setting, RECO2 experiences no data loss. In this case, the double failure might result in the loss of a data file, but the archived log files required to recover it are still available.

Note that the diagrams and associated examples described here are illustrative only. The actual placement algorithm is more involved, and each disk is typically partnered with more than one other disk.

The content type attribute is recommended for use in conjunction with Exadata where the following settings should be applied:

- Set `content .type=data` for the DATA disk group, which is typically used to store database data files.
- Set `content .type=recovery` for the RECO disk group, which is typically used to store the Fast Recovery Area (FRA).
- Set `content .type=system` for the DBFS_DG disk group, which is typically used to store the shared clusterware files (cluster registry and voting disks) and can also be used to store a file system using DBFS.

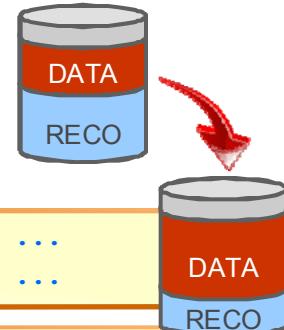
To add the content type attribute setting to an existing diskgroup, simply add the attribute using the `ALTER DISKGROUP` command, and then rebalance the disk group.

Finally, note that the content type attribute setting does not govern the actual contents of the disk group. That is, any type of file can be located on any disk group regardless of the content type setting. For example, a disk group with `content .type=data` can store the flash recovery area for an Oracle database. Likewise, another disk group with `content .type=recovery` can be used to store database data files. It remains the responsibility of the ASM administrator to ensure that each file is located in the appropriate disk group.

Reconfiguring Exadata Storage

Reallocating space between DATA and RECO disk groups:

- Can be performed online (without downtime)
- Can be a time consuming process
- Sufficient free space is required
- Procedure outline:



```
SQL> ALTER DISKGROUP DATA DROP DISKS IN FAILGROUP <CELLNAME> ...
SQL> ALTER DISKGROUP RECO DROP DISKS IN FAILGROUP <CELLNAME> ...
```

```
CellCLI> DROP GRIDDISK ALL PREFIX='DATA' ...
CellCLI> DROP GRIDDISK ALL PREFIX='RECO' ...
CellCLI> CREATE GRIDDISK ALL PREFIX='DATA', SIZE=<NEWSIZE>
CellCLI> CREATE GRIDDISK ALL PREFIX='RECO'
```

```
SQL> ALTER DISKGROUP DATA ADD DISK 'o/<CELL_IB_IP>/DATA*';
SQL> ALTER DISKGROUP RECO ADD DISK 'o/<CELL_IB_IP>/RECO*';
```

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Typically the initial configuration process is used to configure all of the cell disks, grid disks and ASM disk groups on Exadata.

If you want to reconfigure the storage, the best time to do this is before user data is loaded on to the system. If you need to reconfigure storage which is already being used by production databases, it is possible but it requires more time, effort and care.

The most common form of storage reconfiguration is adjusting the size of the default disk groups to reallocate the space between the DATA disk group and the RECO disk group. Typically, the storage is reconfigured one cell (failure group) at a time. That is, the storage on the first cell is reconfigured and after it is completed, and the storage is added back into the disk groups, then the next cell is reconfigured, and so on. The slide shows an overview of the required commands.

The section entitled *Resizing Storage Grid Disks* in the *Oracle Exadata Database Machine Maintenance Guide* outlines alternative methods based on the outline in this slide.

Another reasonably common practice is to make the default disk groups smaller in order to accommodate additional disk groups. Although this is very possible, Oracle recommends that customers should seriously consider if they really need additional disk groups because unnecessarily creating disk groups fragments the available storage and needlessly increases management overheads.

Bear in mind the following general considerations when reconfiguring Exadata storage:

- Reconfiguring an existing disk group requires the ability to drop disks from the disk group, reconfigure them and then add them back into the disk group. If the amount of free space in the disk group is greater than the REQUIRED_MIRROR_FREE_MB value reported in V\$ASM_DISKGROUP, then you can use methods which reconfigure the disk group one cell at a time. If the free space is less than REQUIRED_MIRROR_FREE_MB, then you may need to reorganize your storage to create more free space. It may also be possible, though not recommended, to reconfigure the storage one disk at a time.
- Best practices recommend that all disks in an ASM disk group should be of equal size and have equal performance characteristics. For Exadata this means that all the grid disks allocated to a disk group should be the same size and occupy the same region on each disk. There should not be a mixture of interleaved and non-interleaved grid disks, likewise there should not be a mixture of disks from high-capacity cells and high-performance cells. Finally, the grid disks should all occupy the same location on each disk.
- If you try to drop a grid disk without the FORCE option the command will not be processed and an error will be displayed if the grid disk is being used by an ASM disk group. If you remove a disk from an ASM disk group ensure that the resulting rebalance operation completes before attempting to drop the associated grid disk.
- If you need to use the DROP GRIDDISK command with the FORCE option, use extreme caution since incorrectly dropping an active grid disk could result in data loss.
- If you try to drop a cell disk without the FORCE option the command will not be processed and an error will be displayed if the cell disk contains any grid disks. It is possible to use the DROP CELLDISK command with the FORCE option to drop a cell disk and all the associated grid disks. Use the FORCE option with extreme caution since incorrectly dropping an active grid disk could result in data loss.
- Clusterware files (cluster registry and voting disks) are stored by default in a special ASM disk group named DBFS_DG. Resizing the DBFS_DG disk group is generally not recommended since the grid disks associated with it are sized specially to match the size of the system areas on the first two disk in each cell. If there is a requirement to alter this disk group, or the underlying grid disks or cell disks, special care must be taken to preserve the clusterware files.
- Reconfiguring Exadata storage on an active system without any downtime is possible, however doing so can be a time-consuming process involving many ASM rebalancing operations. The time required depends on the number of storage cells, the existing disk usage and the load on the system. It may take many hours to reconfigure the storage for a Full Rack Database Machine.

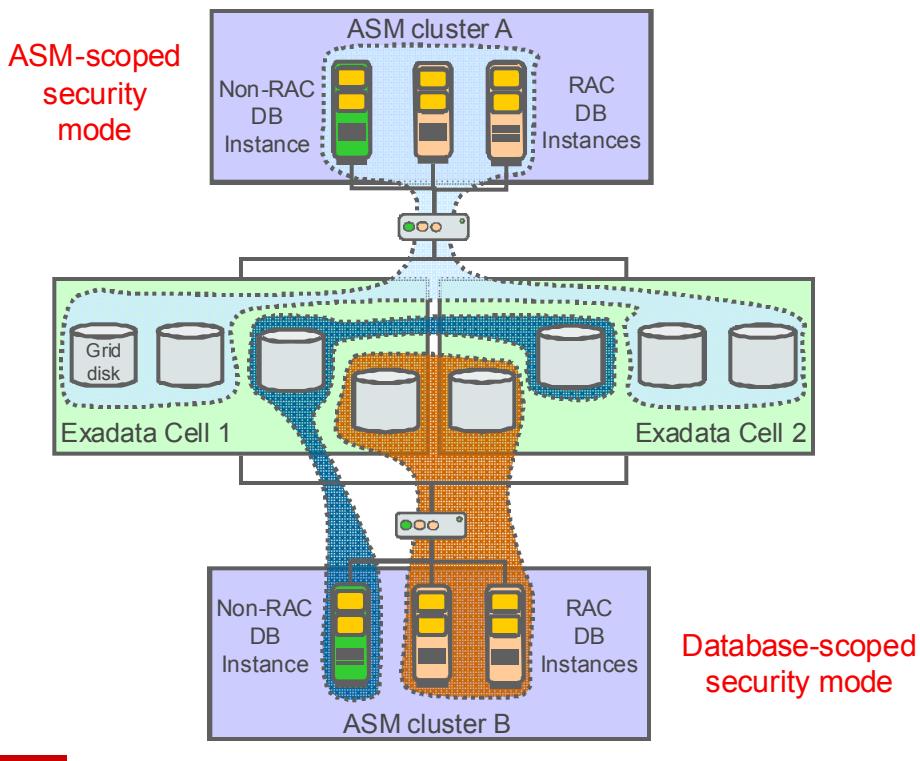
Optional Configuration Tasks

- Configure Exadata Storage Server security.
 - Covered in the next part of this lesson
- Configure I/O Resource Management (IORM).
 - See the lesson titled “I/O Resource Management.”



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Storage Security: Overview



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

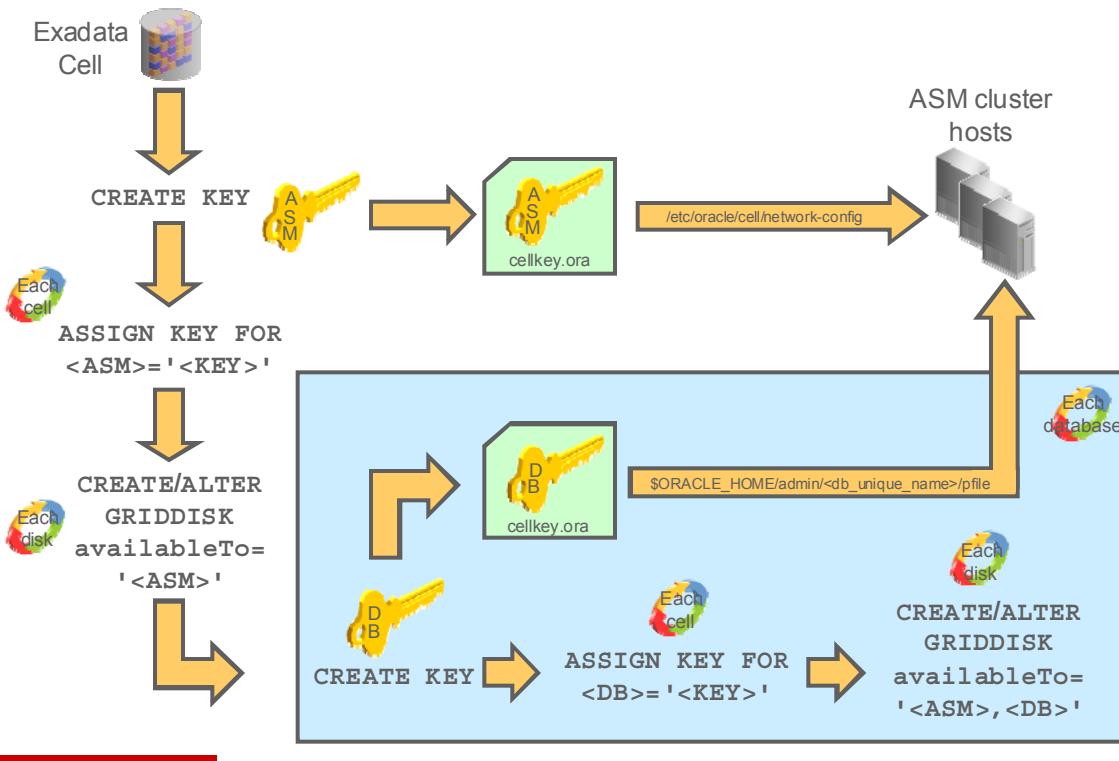
Exadata storage security provides a means to control which ASM clusters and database clients can access specific grid disks on storage cells. It provides a means of ensuring that pools of grid disks are reserved for specific clusters or databases.

- To set up security so that all database clients of an ASM cluster have access to specified grid disks, configure ASM-scoped security.
- To set up security so that specific databases have access to specified grid disks, configure database-scoped security.

Both concepts are illustrated in the slide. ASM cluster A shares two grid disks per cell with all of its database clients. ASM cluster B shares one grid disk per cell to store the single instance database, and another two grid disks (one per cell) to store the RAC database. Exadata storage security ensures that neither cluster can access the grid disks allocated to the other cluster. Also, neither database in cluster B can access the grid disks allocated to the other database.

Note: By default, neither of these security modes are implemented during the initial configuration of Exadata. This situation is called open security, where all database clients can access all grid disks. Open security does not require any configuration, and as long as the network and database hosts are well secured you can use this mode for your production databases. Open security is also useful for non-production environments such as those that house test or development databases.

Exadata Storage Security Implementation



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide outlines the steps to configure ASM-scoped and database-scoped security. It is important to realize that you must set up ASM-scoped security first if you want to set up database-scoped security.

To implement ASM-scoped security, perform the following steps:

1. Shut down the ASM and database instances.
2. Generate a security key using the `CREATE KEY` CellCLI command. Run this command once only on any cell.
3. Use the `ASSIGN KEY` command to assign the security key to the Oracle ASM cluster on all the cells that you want the Oracle ASM cluster to access. You must select a unique name for your ASM cluster to use in the `ASSIGN KEY` command. This name is also used in the `cellkey.ora` file.
4. Enter the Oracle ASM cluster name in the `availableTo` attribute with the `CREATE GRIDDISK` or `ALTER GRIDDISK` command to configure security on the grid disks on all the cells that you want the Oracle ASM cluster to access. At the conclusion of this step, each grid disk has an association with the ASM cluster that is allowed to use the disk.
5. Construct a `cellkey.ora` file using the generated security key. Copy the `cellkey.ora` file into the `/etc/oracle/cell/network-config/` directory on every host in the ASM cluster.
6. Restart the ASM and database instances.

Note that the `cellkey.ora` file contains two entries. One is the generated key and the other identifies the ASM cluster that the key is associated with. For example:

```
key=66e12adb996805358bf82258587f5050  
asm=mycluster
```

After you have configured and tested ASM-scoped security, you can proceed to set up database-scoped security. Perform the following steps for each database you want to configure with database-scoped security:

1. Shut down the ASM and database instances.
2. Generate a security key using the `CREATE KEY CellCLI` command. Run this command once only on any cell.
3. Use the `ASSIGN KEY` command to assign the security key to the database on all the cells that you want the database to access. The database name is determined by the `DB_UNIQUE_NAME` initialization parameter setting.
4. Set the `availableTo` attribute with the `CREATE GRIDDISK` or `ALTER GRIDDISK` command to configure security on all the grid disks (across all the cells) that you want the database to access. The `availableTo` attribute must contain the unique name for both the ASM cluster and the database being used as shown in the following example:

```
alter griddisk DATA_CD_00_EXA1CEL01 availableTo='mycluster,dbm'
```

5. At the conclusion of this step, each grid disk has an association with the ASM cluster and specific database that is allowed to use the disk.
6. Construct a `cellkey.ora` file by using the generated security key. In this case the `cellkey.ora` file contains the generated key assigned to the database and the name of the ASM cluster that is associated with the database.
Copy the `cellkey.ora` file into the
`$ORACLE_HOME/admin/<db_unique_name>/pfile/` directory on every host running the database.
7. Restart the ASM and database instances.

Note: For more information, including examples and further details, refer to the *Oracle Exadata Storage Server Software User's Guide*.

Quiz



Grid disks can be viewed in ASM by using a discovery string that starts with:

- a. c/
- b. o/
- c. g/
- d. e/

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz



The first grid disk you create uses the slowest tracks of the corresponding physical disk:

- a.** True
- b.** False

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz



To create a disk group with Exadata smart storage capabilities enabled, which three attributes must be specified?

- a. compatible.rdbms
- b. compatible.asm
- c. au_size
- d. disk_repair_time
- e. cell.smart_scan_capable

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, e

Summary

In this lesson, you should have learned how to:

- Configure Exadata Storage Server software
- Create and configure ASM disk groups using Exadata storage



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 6 Overview: Configuring Exadata

In these practices, you will perform a variety of Exadata configuration tasks, including:

- Configuring a cell
- Reconfiguring the storage in a Database Machine
- Consuming Exadata storage by using ASM
- Configuring Exadata storage security
- Exercising the privileges associated with the different cell user accounts
- Using the distributed command-line utility (`dcli`)



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

7

I/O Resource Management

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to use Exadata Storage Server I/O Resource Management to manage workloads within a database and across multiple databases.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

I/O Resource Management: Overview

- Traditional benefits of shared storage:
 - Lower administration costs
 - More efficient use of storage
- Common challenge related to shared storage:
 - Workloads interfere with each other. Examples:
 - Large queries interfere with each other.
 - Data loads interfere with analytics.
 - Batch workloads interfere with OLTP performance.
- Using Exadata Storage Server I/O Resource Management, you can govern I/O resource usage among different:

– User types	– Applications
– Workload types	– Databases



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Storage is often shared by different workloads on multiple databases. Shared storage provides some important benefits:

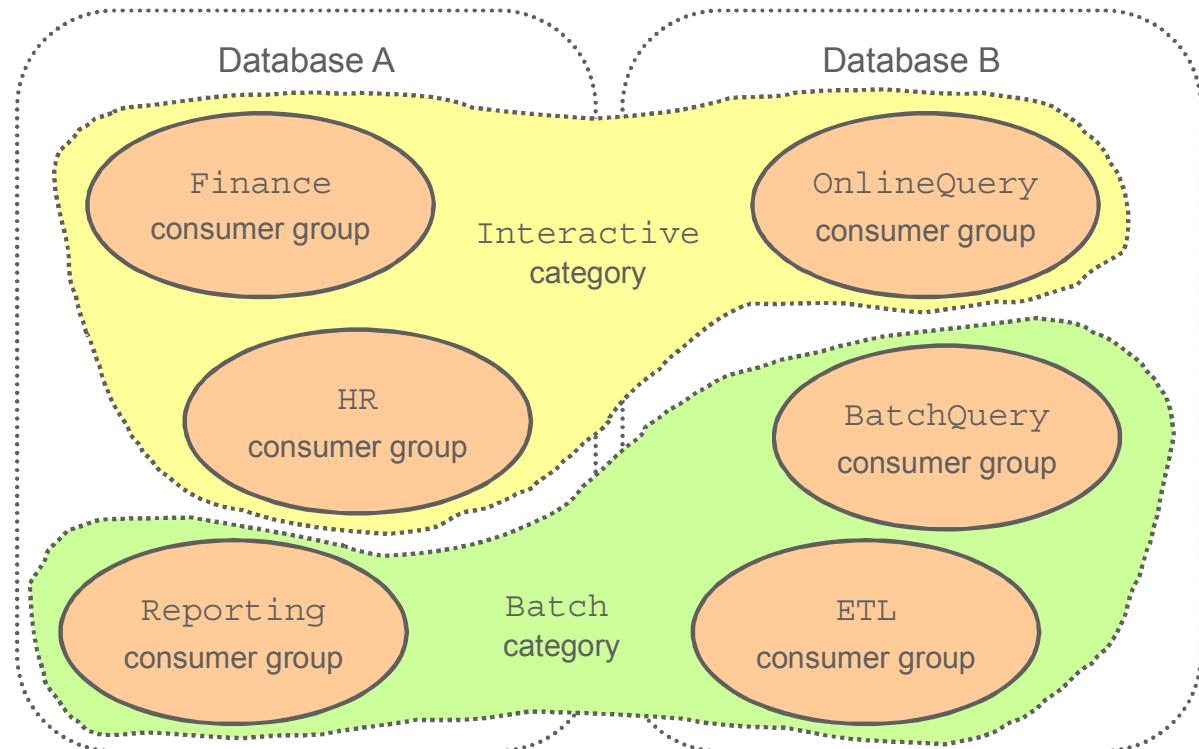
- When a storage system is dedicated to a single database, the administrator must size the storage system based on the database's peak anticipated load and size. The correct balance of storage resources is seldom achieved because real-world workloads are very dynamic. This leads to unused I/O bandwidth and space on some systems, and insufficient amounts on others. Sharing facilitates efficient usage of storage space and I/O bandwidth.
- Sharing lowers administration costs by reducing the number of storage systems.

Shared storage is not a perfect solution. Running multiple types of workloads and databases on shared storage often leads to performance problems. For example, large parallel queries on one database can impact the performance of critical queries on another database. Also, a data load on an analytics database can impact the performance of critical queries running on it. You can mitigate these problems by over-provisioning the storage system, but this diminishes the cost savings of shared storage. You can also avoid running noncritical tasks at peak times, but manually achieving this is laborious. When databases have different administrators who do not coordinate their activities, the task is more difficult.

Exadata Storage Server I/O Resource Management (IORM) allows workloads and databases to share I/O resources automatically according to user-defined policies. To manage workloads within a database, you can define intradatabase resource plans using the Database Resource Manager (DBRM), which has been enhanced to work in conjunction with Exadata Storage Server. To manage workloads across multiple databases, you can define IORM plans.

For example, if a production database and a test database are sharing an Exadata cell, you can configure resource plans that give priority to the production database. In this case, whenever the test database load would affect the production database performance, IORM will schedule the I/O requests such that the production database I/O performance is not impacted. This means that the test database I/O requests are queued until they can be issued without disturbing the production database I/O performance.

I/O Resource Management Concepts



ORACLE®

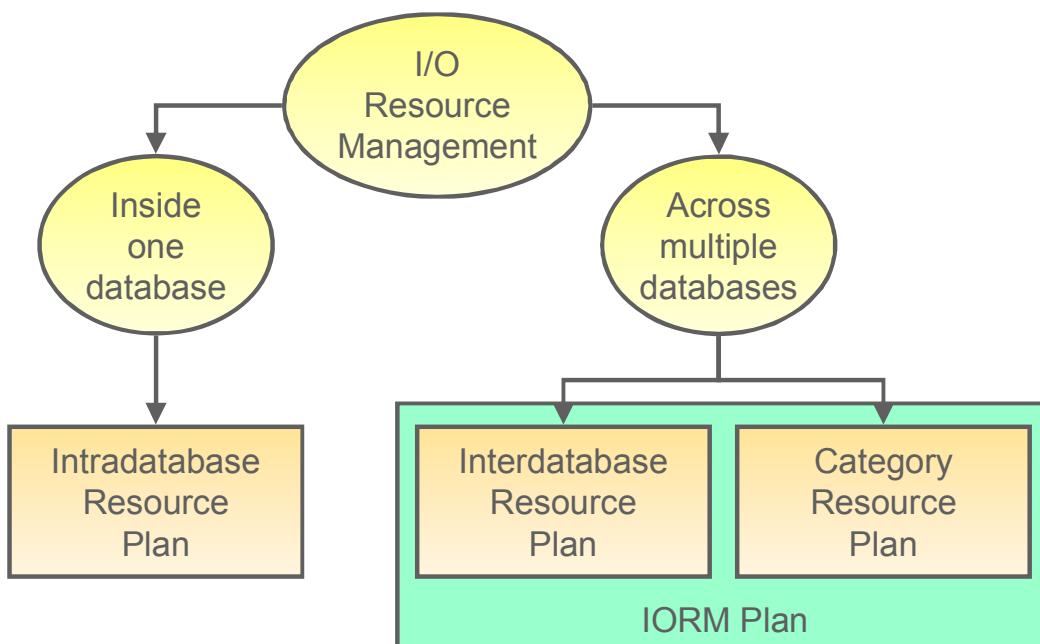
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

A database often has many types of workloads. These workloads may differ in their performance requirements and the amount of resources that they issue. Resource consumer groups provide a way to group sessions that comprise a particular workload. For example, if a database is running four different applications, you can create four consumer groups, one for each application. Alternatively, if your analytics database has three types of workloads, such as online queries, batch queries, and ETL (extraction, transformation, and loading), then you can create a consumer group for each type of workload. After you have created the consumer groups, you must create rules that specify how sessions are mapped to consumer groups.

The database resource plan, or intradatabase resource plan, specifies how resources are allocated among consumer groups in a database. A database may have multiple resource plans, however, only one resource plan can be active at any point in time. This allows database resource management to cater for different requirements associated with different time periods.

Exadata Storage Server IORM extends the consumer group concept using categories. While consumer groups represent collections of users within a database, categories represent collections of consumer groups across all databases. The diagram in the slide shows an example of two categories containing consumer groups across two databases. You can manage I/O resources based on categories by creating a category plan. For example, you can specify priority for consumer groups in the Interactive category over consumer groups in the Batch category for all the databases sharing an Exadata cell.

I/O Resource Management Plans



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

IORM provides different approaches for managing resource allocations. Each approach can be used independently or in conjunction with other approaches.

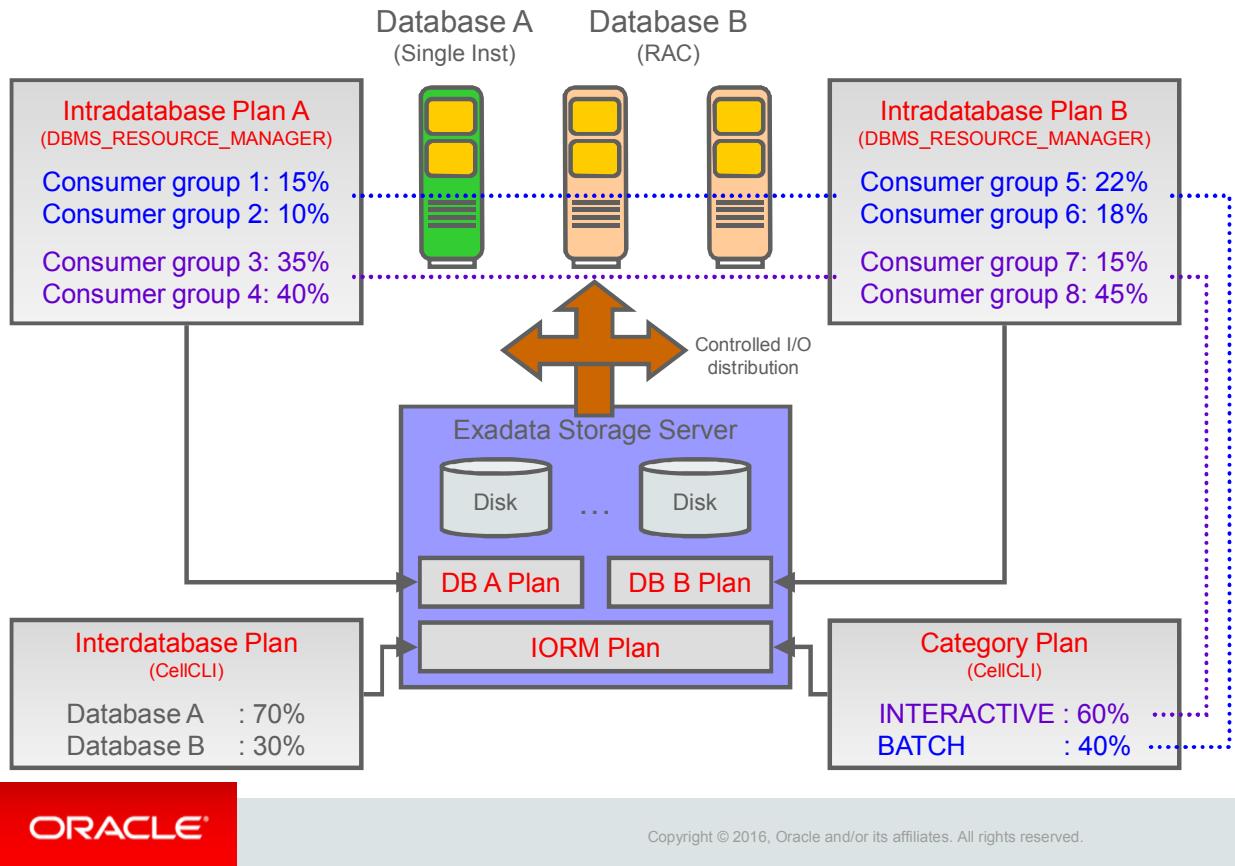
Database resource management enables you to manage workloads within a database. Database resource management is configured within each database, using Database Resource Manager to create an intradatabase resource plan. You should use this feature if you have multiple types of workloads within a database and you need to define a policy for specifying how these workloads share the database resource allocation. If only one database is using Exadata, this is the only IORM feature that you need.

Interdatabase resource management is managed with an interdatabase plan. An interdatabase plan specifies how resources are allocated among multiple databases for each cell. The directives in an interdatabase plan specify allocations to databases, rather than consumer groups.

Category resource management is an advanced feature. It is useful when Exadata is hosting multiple databases and you want to allocate resources primarily by the category of the work being done. For example, suppose all databases have three categories of workloads: OLTP, reports, and maintenance. To allocate the I/O resources based on these workload categories, you would use category resource management.

Both the interdatabase plan and the category plan are defined in a cell object known as the IORM plan.

I/O Resource Management Plans: Example

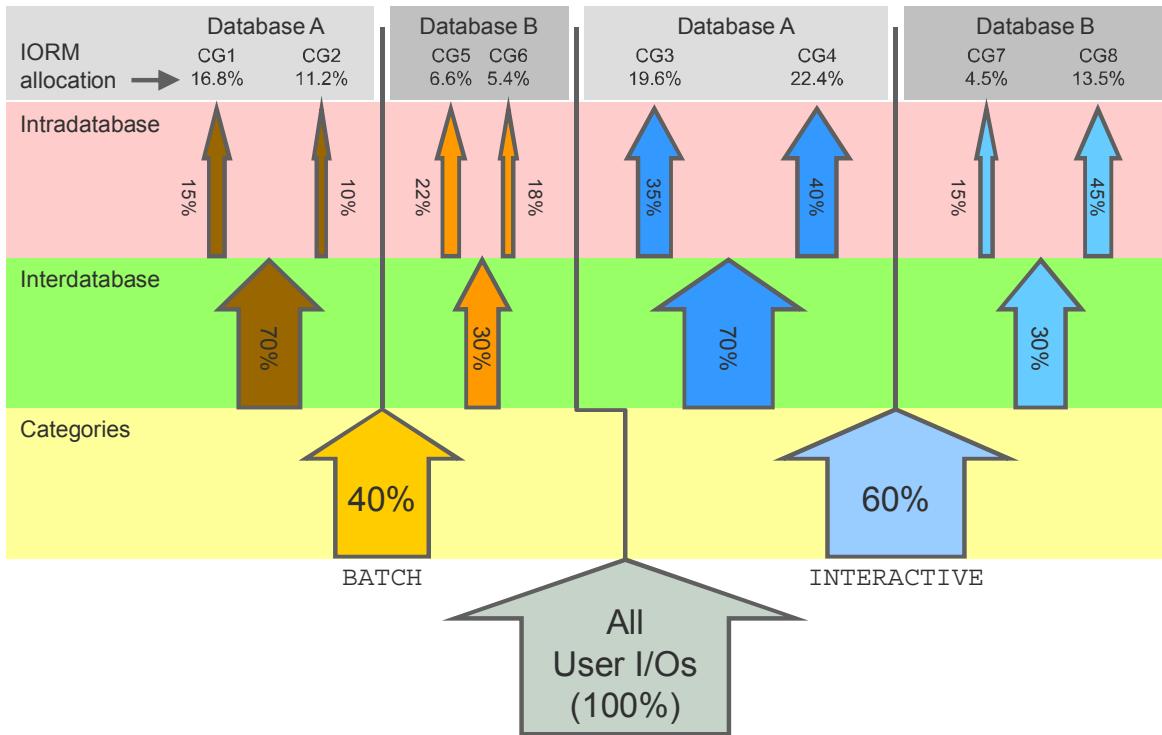


For each database, you can use DBRM to create an intradatabase resource plan. When you set an intradatabase resource plan, a description of the plan is automatically sent to each cell. In the example in the slide, Database A and Database B have separate intradatabase plans. Note also that each consumer group in each intradatabase plan is associated with either the INTERACTIVE or BATCH category.

At each cell, an interdatabase plan can be configured and enabled. In the example in the slide, the interdatabase plan is configured with a larger resource allocation for Database A (70%) than for Database B (30%).

Also within each cell, you can categorize consumer groups from different databases and distribute I/O resources according to the various categories. In the example in the slide, the INTERACTIVE category (60%) is allocated a greater resource share than the BATCH category (40%).

I/O Resource Management Plans: Example



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The category, interdatabase, and intradatabase plans are used together by Exadata to allocate I/O resources.

The category plan is first used to allocate resources among the categories. When a category is selected, the interdatabase plan is used to select a database; only databases that have consumer groups with the selected category can be selected. Finally, the selected database's intradatabase plan is used to select one of its consumer groups. The percentage of resource allocation represents the probability of making a selection at each level.

Expressing this as a formula:

$$P_{Cn} = cgn / \text{sum(catcgs)} * db\% * cat\%$$

where:

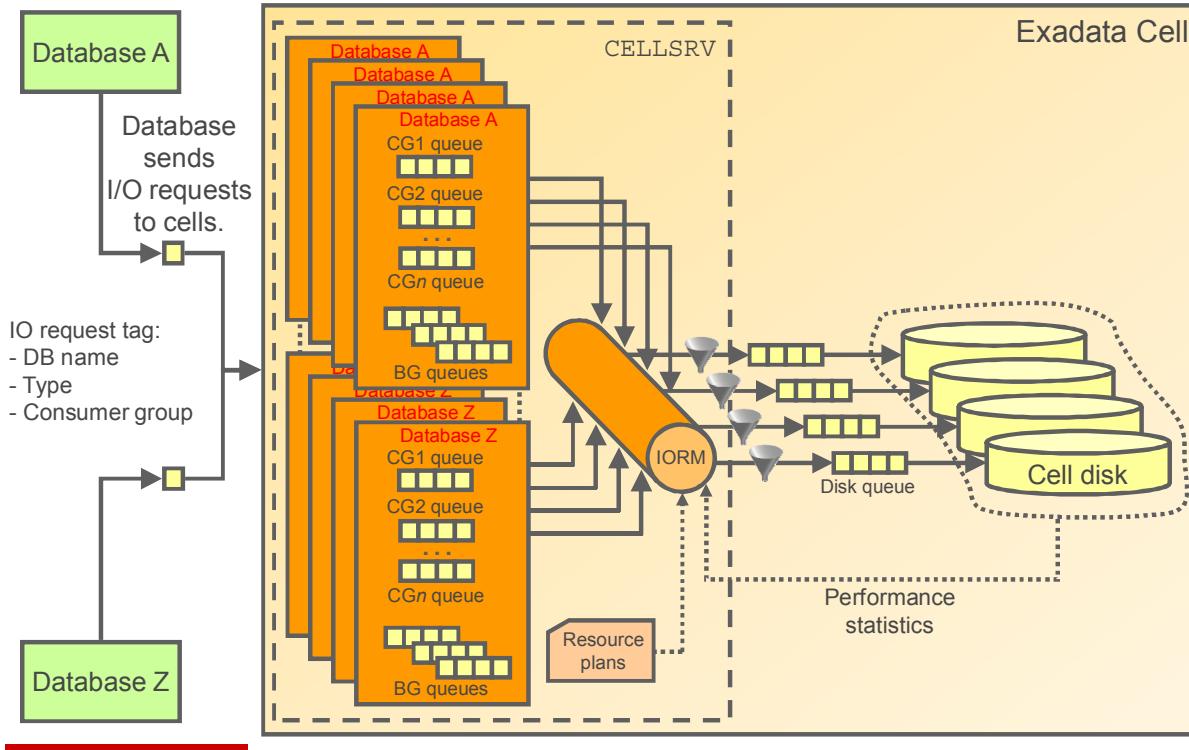
- P_{Cn} is the probability of selecting consumer group n
- cgn is the resource allocation for consumer group n
- sum(catcgs) is the sum of the resource allocations for all consumer groups in the same category as consumer group n and on the same database as consumer group n
- $db\%$ is the database allocation percentage in the interdatabase plan
- $cat\%$ is the category allocation percentage in the category plan

The hierarchy used to distribute I/Os is illustrated in the slide. The example is continued from the previous slide but the consumer group names are abbreviated to CG1, CG2, and so on.

Notice that although each consumer group allocation is expressed as a percentage within each database, IORM is concerned with the ratio of consumer group allocations within each category and database. For example, CG1 nominally receives 16.8% of I/O resources from IORM ($^{15}/(15+10) * 70\% * 40\%$); however, this does not change if the intradatabase plan allocations for CG1 and CG2 are doubled to 30% and 20%, respectively. This is because the allocation to CG1 remains 50% greater than the allocation to CG2. This behavior also explains why CG1 (16.8%) and CG3 (19.6%) have a similar allocation through IORM even though CG3 belongs to the higher priority category (60% versus 40%) and has a much larger intradatabase plan allocation (35% versus 15%).

Note: ASM I/Os (for rebalance and so on) and I/Os issued by Oracle background processes are handled separately and automatically by Exadata. For clarity, background I/Os are not shown in the example.

IORM Architecture



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

IORM manages Exadata I/O resources on a per-cell basis, scheduling incoming I/O requests according to the configured resource plans. IORM schedules I/O by selecting requests from different CELLSRV queues. The resource plans are used to determine the order in which the queued I/O requests are issued to disk. By default, the goal of IORM is to fully use the available disk resources. Any allocation that is not fully utilized is made available to other workloads in proportion to the configured resource plans.

IORM only intervenes when needed. For example, IORM does not intervene if there is only one active consumer group on one database because there is no possibility of contention with another consumer group or database.

Background I/Os are scheduled based on their priority relative to the user I/Os. For example, redo writes and control file I/Os are critical to performance and are always prioritized above all user I/Os. Writes by the database writer process (`DBWn`) are scheduled at the same priority level as user I/Os.

The diagram in the slide illustrates the high-level implementation of IORM. For each disk-based cell disk, each database accessing the cell has one I/O queue per consumer group and three background I/O queues. The background I/O queues correspond to high, medium, and low priority requests with different I/O types mapped to each queue. If you do not set an intradatabase resource plan, all nonbackground I/O requests are grouped into a single consumer group called `OTHER_GROUPS`.

Getting Started with IORM

```
CellCLI> LIST IORMPLAN DETAIL
      name:                      cell01_IORMPLAN
      catPlan:
      dbPlan:
      objective:                 basic
      status:                     active

CellCLI> ALTER IORMPLAN objective=balanced
IORMPLAN successfully altered

CellCLI> LIST IORMPLAN DETAIL
      name:                      cell01_IORMPLAN
      catPlan:
      dbPlan:
      objective:                 balanced
      status:                     active
```



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The settings that govern IORM are contained in an `IORMPLAN` object on each Exadata cell. The `IORMPLAN` object is automatically created when the cell is configured. It cannot be dropped or recreated.

Initially, the IORM plan for every cell is configured as follows:

- The IORM plan has a name of form `<cell_name>_IORMPLAN`.
- The `catPlan` and `dbPlan` attributes are empty. These attributes define the interdatabase I/O resource management plan and the category I/O resource management plan. Details regarding these plans are provided later in this lesson.
- The IORM plan objective is set to `basic`. In basic mode, IORM only attempts to protect small I/Os from extreme latencies that may occur during heavy I/O loads, it does not enforce any user-defined interdatabase or intradatabase resource plans.
- The IORM plan status is set to `active`.

Before I/O resources can be fully managed by IORM, you need to ensure that the IORM plan is active and that the IORM plan objective is set to a value other than `basic`.

The example in the slide shows a default IORM configuration being altered to set the IORM plan objective. The available objective settings are outlined next.

Setting the IORM Objective

Available IORM objective settings:

- basic
 - IORM does not enforce user-defined plans.
 - IORM protects against extreme latencies for small I/O requests.
 - Maximum throughput is maintained.
- low_latency
 - Minimizes latency by limiting the number of concurrent I/O requests
 - Useful for critical OLTP workloads
 - Performance of high-throughput workloads may suffer
- high_throughput
 - Maximizes throughput by not limiting concurrent I/O requests
 - Useful for data warehousing and analytics
 - Performance of latency-critical workloads may suffer
- balanced
 - Balances low disk latency and high throughput
 - Useful for mixed workloads
- auto
 - IORM decides the best objective setting based on active plans and workloads.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The IORM objective setting governs how physical I/O requests are issued to disk by IORM. To understand the different settings and what they achieve you must first understand how a disk services I/O requests.

The largest cost involved in servicing an I/O is positioning the disk head in the right location to perform the I/O. Hence, a disk drive maintains a buffer of concurrent I/O requests that is used to optimize how they are serviced. If the buffer is large then the opportunity for optimizing the I/Os increases. In this case, overall throughput for the disk is maximized however the latency for a particular request can be larger since other I/Os may be serviced along the way. If the buffer is small, then the I/Os will be serviced more sequentially. In this case, latency is reduced because there are fewer concurrent I/Os being serviced. However, overall throughput is also reduced because the I/O request stream will be less optimized.

IORM allows Exadata administrators to control the IORM objective so that I/O performance can be optimized according to the workload characteristics and business requirements. The slide lists the available settings and outlines the characteristics of each setting.

For the `auto` objective, IORM continually analyzes the I/O workload and the active IORM plans to select the appropriate objective setting. For example, if a particular database is allocated the majority of I/O resources and it is running an OLTP workload, then the `low_latency` setting will be automatically engaged.

Enabling Intradatabase Resource Management

- You can enable intradatabase resource management:
 - Manually:
 - Set the database's RESOURCE_MANAGER_PLAN parameter.
 - Automatically:
 - Create a job scheduler window.
 - Associate a resource plan with the window.
- Exadata Storage Server is notified when an intradatabase resource plan is set or modified:
 - Enabled or modified plan sent to each cell using iDB
- You must set the IORMPLAN objective on all Exadata cells.
- The following are the commonly used intradatabase plans:
 - mixed_workload_plan
 - dss_plan
 - default_maintenance_plan



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

An intradatabase resource plan can be manually enabled with the RESOURCE_MANAGER_PLAN initialization parameter or automatically enabled using the job scheduler.

When you set an intradatabase resource plan on the database, a description of the plan is automatically sent to each cell. When a new cell is added or an existing cell is restarted, the current intradatabase plan is automatically sent to the cells. This resource plan is used to manage resources on both the database server and cells.

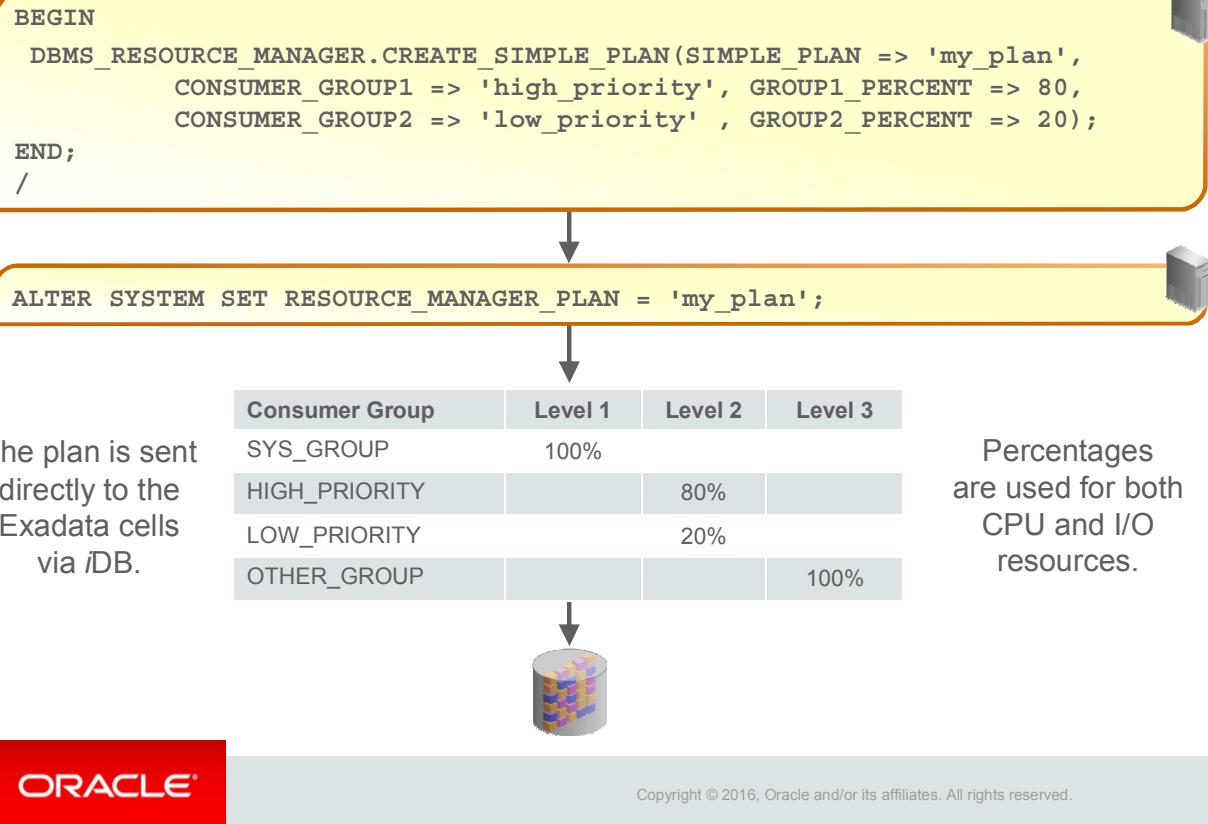
To enforce the intradatabase plan using IORM, you must set the IORM plan objective to a value other than basic.

Oracle Database provides several predefined intradatabase plans. The most commonly used are mixed_workload_plan, dss_plan and default_maintenance_plan.

Intradatabase plans do not contain a directive for background I/O activity. Background I/Os are scheduled based on their priority relative to the user I/Os. For example, redo writes, and control file reads and writes are critical to performance and are always prioritized above all user I/Os.

Note: For Oracle RAC databases on Exadata, all the RAC instances must be set to use the same resource plan.

Intradatabase Plan: Example



The intradatabase I/O resource plan specifies how I/O resources are allocated among consumer groups in a specific database.

An intradatabase I/O resource plan is created with the procedures in the DBMS_RESOURCE_MANAGER PL/SQL package. There are no specific I/O resource parameters or procedures. You create an intradatabase I/O resource plan exactly the same way as you would create a CPU resource plan. When you specify an allocation percentage, this percentage applies to both database server CPU and Exadata Storage Server I/O resources. There are no specific I/O settings because typically you are constrained by CPU or I/O, but not both at the same time.

The example in the slide uses the CREATE_SIMPLE_PLAN procedure to create MY_PLAN. This resource plan is used to manage CPU resources at the database level, and I/O resources at the Exadata cell level.

Note that the MAX_UTILIZATION_LIMIT attribute can also be used to specify the maximum CPU and I/O utilization limit for consumer groups. This attribute can be set by using the DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE procedure.

Enabling IORM for Multiple Databases

- You can enable IORM for multiple databases by configuring an `IORMPLAN`:
 - The category plan assigns I/O resources using categories.
 - The interdatabase plan assigns I/O resources using database names.
 - All combinations are possible.
- Use CellCLI to define and activate the `IORMPLAN` on each cell.
- Configure the same `IORMPLAN` on each cell.
- Only one `IORMPLAN` can be active at a time on a cell.
- `IORMPLAN` settings are persistent across cell reboots.
- All databases get equal allocations in the absence of an `IORMPLAN`.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

I/O resource management for multiple databases is configured with the `IORMPLAN`. The `IORMPLAN` specifies how I/O resources are allocated for each cell.

The `IORMPLAN` contains both an interdatabase plan, also called a DB plan, and a category plan. The directives in the DB plan specify I/O resource allocations to database names, rather than consumer groups. The directives in the category plan specify I/O resource allocations to categories, rather than databases or consumer groups. The `IORMPLAN` is configured and enabled with CellCLI on each cell. Only one `IORMPLAN` can be active on a cell at any given time.

At startup, the `IORMPLAN` is an empty string, which effectively turns off inter-database IORM. In that case, all the databases receive an equal allocation.

Interdatabase Plan: Example

```
CellCLI> ALTER IORMPLAN
```

```
    dbplan= ((name=sales_prod, level=1, allocation=80),
              (name=finance_prod, level=1, allocation=20),
              (name=sales_dev, level=2, allocation=100),
              (name=sales_test, level=3, allocation=50),
              (name=other, level=3, allocation=50)),
    catplan=''
```



Database	Level 1	Level 2	Level 3
sales_prod	80%		
finance_prod	20%		
sales_dev		100%	
sales_test			50%
other			50%

Maximum of
32 directives.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

On each Exadata cell, an interdatabase plan specifies how resources are divided among multiple databases. The directives in an interdatabase plan specify allocations to databases, rather than consumer groups. The interdatabase plan is configured and activated with CellCLI, on each cell.

The above example implements an interdatabase plan following the directives shown in the table.

The interdatabase plan is created by specifying the DBPLAN part of the IORMPLAN. The interdatabase plan is similar to an intradatabase plan in that each directive can specify a level from 1 to 8 and an allocation amount in percentage terms. Using this method, an interdatabase plan can support up to 32 directives. For a given plan, all the allocations at any level must add up to 100 or less.

An interdatabase plan differs from an intradatabase plan in that it cannot contain subplans and it only contains I/O resource directives.

As a best practice, you should create a directive for each database using the same Exadata cell. To ensure that any database without an explicit directive can be managed, you need to create an allocation named OTHER.

You can remove an interdatabase plan using:

```
ALTER IORMPLAN dbplan= ''
```

Category Plan: Example

DBA_RSRC_CONSUMER_GROUPS

CONSUMER_GROUP	CATEGORY
SYS_GROUP	ADMINISTRATIVE
BATCH_GROUP	BATCH
INTERACTIVE_GROUP	INTERACTIVE
ORA\$...	MAINTENANCE
OTHER_GROUPS	OTHER
DEFAULT_CONSUMER_GROUP	OTHER
LOW_GROUP	OTHER
AUTO_TASK_CONSUMER_GROUP	OTHER

DBMS_RESOURCE_MANAGER.CREATE_CATEGORY

Category	Level 1	Level 2	Level 3
Interactive	90%		
Batch		80%	
Maintenance			50%
Other			50%

```
Cell1CLI> ALTER IORMPLAN
dbplan= ''
catplan=(
    (name=interactive, level=1, allocation=90),
    (name=batch, level=2, allocation=80),
    (name=maintenance, level=3, allocation=50),
    (name=other, level=3, allocation=50)
)
```

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Database Resource Manager enables you to specify a category for every consumer group. The predefined categories and their associated consumer groups are listed in the slide. This is the default situation after database creation.

If you decide to use the default categories, you should map all administrative consumer groups in all databases to the ADMINISTRATIVE category. All high-priority user activity, such as consumer groups for important online transaction processing (OLTP) transactions and time-critical reports, should be mapped to the INTERACTIVE category. All low-priority user activity, such as reports, maintenance, and low-priority OLTP transactions, should be mapped to the BATCH, MAINTENANCE, and OTHER categories.

You can create your own categories using the CREATE_CATEGORY procedure in the DBMS_RESOURCE_MANAGER package, and then assign your category to a consumer group using the CREATE_CONSUMER_GROUP or UPDATE_CONSUMER_GROUP procedures.

You can then manage I/O resources based on categories by creating a category plan. The example shown in the slide implements a category plan based on the allocations described in the table. With this plan, consumer groups associated with the INTERACTIVE category get up to 90 percent of I/O resources. 80 percent of the remainder, including any unutilized allocation from the INTERACTIVE category, is allocated to the BATCH category. The MAINTENANCE and OTHER categories share the remainder.

Any consumer group without an explicitly specified category defaults to the OTHER category.

Complete Example

Database A:

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN(SIMPLE_PLAN => 'DB_A_Plan',
                                             CONSUMER_GROUP1 => 'CG1', GROUP1_PERCENT => 15,
                                             CONSUMER_GROUP2 => 'CG2', GROUP1_PERCENT => 10,
                                             CONSUMER_GROUP3 => 'CG3', GROUP1_PERCENT => 35,
                                             CONSUMER_GROUP4 => 'CG4', GROUP2_PERCENT => 40);
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG1',
                                                NEW_CATEGORY => 'BATCH');
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG2',
                                                NEW_CATEGORY => 'BATCH');
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG3',
                                                NEW_CATEGORY => 'INTERACTIVE');
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG4',
                                                NEW_CATEGORY => 'INTERACTIVE');
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DB_A_Plan';
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This slide is the first in a series of three slides that provide a complete example showing the use of different IORM plan types at the same time. The example is based on the scenario introduced on pages 7, 8, and 9 of this lesson.

In this slide, the commands required to configure DBRM on Database A are shown.

Note that the example does not show the creation of any categories using DBMS_RESOURCE_MANAGER.CREATE_CATEGORY because the categories used in the scenario (BATCH and INTERACTIVE) are categories that are predefined inside Oracle Database by default.

Complete Example

Database B:

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN(SIMPLE_PLAN => 'DB_B_Plan',
                                             CONSUMER_GROUP1 => 'CG5', GROUP1_PERCENT => 22,
                                             CONSUMER_GROUP2 => 'CG6', GROUP1_PERCENT => 18,
                                             CONSUMER_GROUP3 => 'CG7', GROUP1_PERCENT => 15,
                                             CONSUMER_GROUP4 => 'CG8', GROUP2_PERCENT => 45);
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG5',
                                                NEW_CATEGORY => 'BATCH');
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG6',
                                                NEW_CATEGORY => 'BATCH');
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG7',
                                                NEW_CATEGORY => 'INTERACTIVE');
  DBMS_RESOURCE_MANAGER.UPDATE_CONSUMER_GROUP(CONSUMER_GROUP => 'CG8',
                                                NEW_CATEGORY => 'INTERACTIVE');
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DB_B_Plan';
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In this slide, the commands required to configure DBRM on Database B are shown. These commands are essentially the same as for Database A except for the different consumer group names and resource allocation percentages.

Complete Example

Exadata Cells:

```
CellCLI> ALTER IORMPLAN objective=auto  
  
CellCLI> ALTER IORMPLAN  
    dbplan=((name=Database_A, level=1, allocation=70),  
            (name=Database_B, level=1, allocation=30)),  
    catplan=((name=INTERACTIVE, level=1, allocation=60),  
             (name=BATCH, level=1, allocation=40))
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This slide shows the commands required to configure IORM on the Exadata cells. Exadata cells use the IORM plan in conjunction with the DBRM plans propagated by the databases to allocate I/O resources.

Note that the IORM plan objective must be set to a value other than basic in order for IORM to enforce the plan.

Using Share-Based Allocation in the Interdatabase Plan

```
CellCLI> ALTER IORMPLAN  
    dbplan=((name=sales_prod, share=8),  
            (name=finance_prod, share=2),  
            (name=sales_dev, share=10),  
            (name=sales_test, share=5),  
            (name=default, share=5),  
    catplan=''
```



Maximum of
1024 directives.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing with Exadata Storage Server software release 11.2.3.1.0, I/O allocations in the interdatabase plan can be expressed as shares rather than using the level and allocation attributes shown on the previous page. Each share is a value between 1 and 32, with 1 being the lowest share, and 32 being the highest share. The share value represents the relative importance of each database rather than specifying an IO allocation percentage.

Share-based allocation is a simplified approach designed to support large numbers of databases. Using shared-based allocations, an interdatabase plan can support up to 1024 directives.

Note that with shared-based allocation, the OTHER directive is not used. Instead, the DEFAULT directive can be used to specify the default share value for any database not listed in the interdatabase plan.

Setting Database I/O Utilization Limits

```
CellCLI> ALTER IORMPLAN
    dbplan=((name=db1, level=1, allocation=50, limit=75),
             (name=db2, level=1, allocation=30, limit=75),
             (name=db3, level=1, allocation=20, limit=50),
             (name=other, level=2, allocation=100)),
    catplan=''
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide contains an example of an interdatabase plan using the `limit` attribute.

By default, the goal of IORM is to fully use the available disk resources. Any allocation that is not fully utilized is made available to other workloads. The `limit` attribute modifies this behavior by specifying the absolute upper limit of I/O resource consumption allowed for a database. This is useful in situations where more consistent I/O performance is desired rather than maximizing I/O resource utilization.

The `limit` attribute can only be specified for interdatabase plan directives. It specifies the I/O utilization limit for a database in percentage terms. If the `limit` attribute is used, then the associated value must be greater than zero, and less than or equal to 100.

Note that the `limit` attribute can also be specified in conjunction with share-based allocations. It can also be specified by itself, that is, without any `level`, `allocation`, or `share` attributes.

I/O Resource Management Profiles

- Simplifies management of interdatabase plans for many databases
- Associates databases with a manageable number of profiles
 - Rather than defining plan directives for each database
- Databases map to profiles using the `DB_PERFORMANCE_PROFILE` instance parameter
- Example:

```
CellCLI> ALTER IORMPLAN dbplan=(  
    (name=gold, share=10, limit=100, type=profile),  
    (name=silver, share=5, limit=60, type=profile),  
    (name=bronze, share=1, limit=20, type=profile))
```

```
SQL> alter system set db_performance_profile=gold scope=spfile;
```



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing with Exadata software release 12.1.2.1.0, IORM interdatabase plans support profiles that reduce management of interdatabase plans for environments with many databases.

Previously, an administrator had to specify plan directives for every database in the interdatabase plan and the plan needed to be updated each time a new database was created. IORM profiles greatly reduce this management overhead. Using profiles, the administrator can now associate IORM directives with a named profile, rather than an individual database. The administrator can then map new and existing databases to one of the profiles by using the database parameter `DB_PERFORMANCE_PROFILE`. Using this mechanism, each database automatically inherits the attributes from the specified profile.

Note that the minimum database software requirement for IORM profiles is Oracle Database 12c release 12.1.0.2 with Exadata bundle patch 4.

Interdatabase Plans and Database Roles

```
CellCLI> ALTER IORMPLAN
    dbplan= ( (name=sales1, level=1, allocation=30, role=primary),
              (name=sales2, level=1, allocation=35, role=primary),
              (name=sales1, level=2, allocation=20, role=standby),
              (name=sales2, level=2, allocation=25, role=standby),
              (name=other, level=3, allocation=50)),
    catplan=''
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide contains an example of an interdatabase plan using the `role` attribute.

The `role` attribute indicates that the directive is applied only when the database is in that database role. This provides the flexibility to automatically adjust the IORM plan according to the role of the database in an Oracle Data Guard environment. If the `role` attribute is not specified, the directive is applied regardless of the database role.

The `role` attribute can only be specified for interdatabase plan directives. However, the `role` attribute cannot be specified for the OTHER directive.

Using Database I/O Metrics

- There are separate metrics for small I/Os (128 KB or less) and large I/Os (greater than 128 KB).
- You can monitor IORM to understand resource consumption and make required adjustments.
 - Which database has the heaviest load?
 - Look for highest DB_IO_RQ_SM + DB_IO_RQ_LG values.
 - Which database was throttled the most?
 - Look for highest DB_IO_WT_SM + DB_IO_WT_LG values.

Name	Description
DB_IO_RQ_SM	Total number of I/O requests issued by the database since any resource plan was set
DB_IO_RQ_LG	
DB_IO_RQ_SM_SEC	I/O requests per second issued by the database in the past minute
DB_IO_RQ_LG_SEC	
DB_IO_WT_SM	Total number of seconds that I/O requests issued by the database waited to be scheduled
DB_IO_WT_LG	



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Storage Server provides three groups of I/O metrics that correspond to the three types of IORM plans: category metrics, database metrics, and consumer group metrics. I/O metrics allow you to understand I/O consumption and make adjustments to optimize performance and resource utilization.

For each I/O metric, a distinction is made between small I/Os, typically associated with OLTP applications, and large I/Os, which are usually indicative of analytical workloads. A small I/O is up to 128 KB in size and a large I/O is greater than 128 KB in size. I/O metric names include _SM or _LG to identify small or large I/Os, respectively.

For database metrics the `objectType` attribute is set to `IORM_DATABASE`. The table in the slide gives you a quick description of some important database I/O metrics. A separate set of metric observations is available for each database specified in the IORM plan. Metric observations for different databases are differentiated by the name of the database, which is set in the `metricObjectName` attribute. You can compare metrics between databases to determine which one has the heaviest load or which one was throttled most as outlined in the slide. A special `metricObjectName` value of `_OTHER_DATABASE_` is used for database I/O metrics associated with ASM and for databases that are not explicitly mentioned in the interdatabase IORM plan.

Although this slide focuses on database metrics, the same principles apply for category metrics and consumer group metrics. For example, the CG_IO_RQ_SM_SEC metric specifies the rate of small I/O requests issued by a consumer group per second over the past minute. A large value indicates a heavy I/O workload from the associated consumer group during the past minute.

See the lesson titled “Monitoring Exadata Storage Servers” for more on storage server metrics.

I/O Resource Management for Flash

Commencing with Exadata software release 12.1.2.1.0, IORM manages flash I/Os as well as disk I/Os:

- Prioritizes flash OLTP I/O requests over flash scans
- Uses existing IORM plan definitions
- No additional user or administrator controls



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing with Exadata software release 12.1.2.1.0, IORM manages flash I/Os in addition to disk I/Os to control I/O contention between databases, pluggable databases, and consumer groups. Before this version, IORM only arbitrated I/Os to disk.

Flash IORM protects the latency of critical OLTP I/O requests in flash cache. When table scans are running on flash concurrently with OLTP I/O requests, the OLTP latency can be significantly impacted. Flash IORM queues and throttles the table scan, and other lower priority I/O requests. The critical OLTP I/O requests are not queued. When the flash disks are not busy serving critical OLTP I/O requests, the queued I/O requests are issued based on the resource allocations in the inter-database plan.

Flash I/O Resource Management is enabled by default and uses existing IORM plan definitions. There are no additional user or administrator controls.

Flash Cache and Flash Log Resource Control

IORM can control whether a database can use Exadata Smart Flash Cache or Smart Flash Log:

```
CellCLI> ALTER IORMPLAN dbPlan= (
    (name=oltp, level=1, allocation=80, flashCache=on, flashLog=on),
    (name=dss, level=1, allocation=20, limit=50, flashCache=off, flashLog=on),
    (name=other, level=2, allocation=100, flashCache=off, flashLog=off))
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In addition to prioritizing I/Os, IORM can be used to control if a database is allowed to use Exadata Smart Flash Cache. This allows flash cache to be reserved for the most important databases, which is especially useful in environments that are used to consolidate multiple databases.

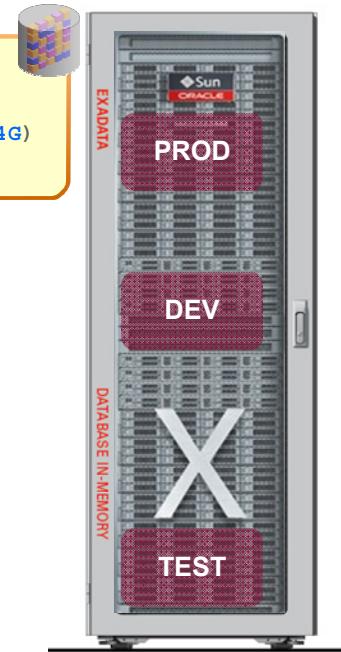
Set `flashCache=on` in the interdatabase plan directive to allow the associated databases to use Exadata Smart Flash Cache, and set `flashCache=off` to prevent databases from using Exadata Smart Flash Cache. If an interdatabase plan directive does not contain the `flashCache` attribute, then `flashCache=on` is assumed.

Similarly, the `flashLog` attribute can be set to specify whether or not databases can use Exadata Smart Flash Log. If the `flashLog` attribute is not specified, then `flashLog=on` is assumed.

Flash Cache Space Resource Management

Flash cache usage limits can be set for each database:

```
CellCLI> ALTER IORMPLAN dbPlan=(  
    (name=prod, share=8, flashCacheMin=4096),  
    (name=dev, share=2, flashCacheMin=1G, flashCacheLimit=4G)  
    (name=test, share=1, limit=40, flashCacheSize=2048))
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

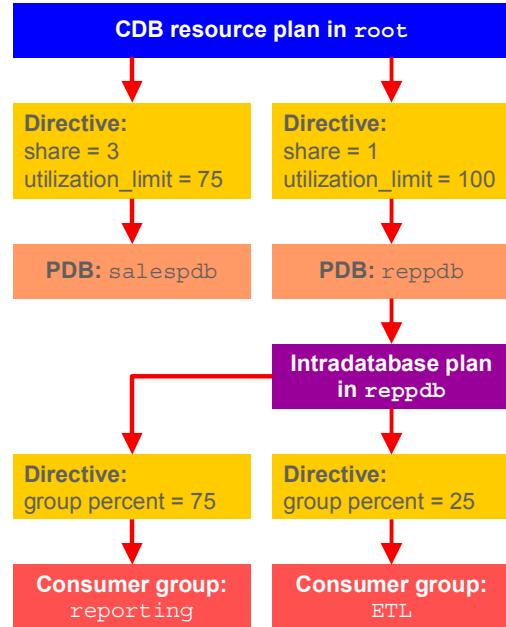
In addition to controlling access to flash cache for a database, flash cache space resource management allows users to specify limits, which define the amount of flash cache space that can be used by a database. The slide shows an example of an interdatabase IORM plan that uses this facility. The available attributes are:

- FlashCacheMin specifies the minimum guaranteed space in flash cache for a database.
- FlashCacheLimit specifies the maximum space in flash cache that a database can use. The attribute is a soft limit, so if the flash cache is not full the database can exceed its allocated quota.
- FlashCacheSize specifies the guaranteed hard limit space in the flash cache for a database. This space in the flash cache is reserved but because this is a hard limit the database cannot exceed its quota even when the flash cache is not full.

The sum of `flashCacheMin` and `flashCacheSize` across all the directives should be less than the size of flash cache.

Using Exadata I/O Resource Management with Oracle Database 12c

- Non-CDB databases use IORM in the same way as version 11.2 databases
- To manage PDBs within a CDB:
 - New CDB resource plan:
 - Defined in the CDB root container
 - Allocates resources to PDBs based on shares
 - Can also enforce utilization limits for PDBs
 - Background resource usage is charged to the root container
 - Example: log file sync
 - Works in conjunction with other resource plans
 - Other resource plans are still available:
 - Intradatabase plan is defined inside each PDB
 - Interdatabase plan allocates resources across databases (including CDBs)
 - Category plan still allocates resources across categories



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Version 12.1 databases that do not use the multitenant architecture (non-CDB databases) can use Exadata IORM in the same way as version 11.2 databases, that is, intradatabase, interdatabase and category plans are defined and operate in the same way as before.

To manage PDBs within a CDB, you can configure a new CDB resource plan in the CDB root container. A CDB resource plan interacts with IORM in essentially the same way as an intradatabase resource plan, except that a non-CDB resource plan manages resources amongst consumer groups and a CDB resource plan manages resources amongst PDBs.

Like intradatabase resource plans, the CDB resource plan manages database server CPU and Exadata Storage Server I/O. Also, to enforce a CDB resource plan, you must set the cell IORM objective to a setting other than `basic`. The recommended initial setting is `auto`.

The slide lists some of the properties of a CDB resource plan.

Note that all the other IORM plans and directives work in conjunction with the new CDB resource plan. The diagram in the slide illustrates how a CDB resource plan works in conjunction with an intradatabase plan. In this example, resources are shared between two PDBs (`salespdb` and `reppdb`) using the CDB resource plan. Furthermore, the resources allocated to `reppdb` are shared between two consumer groups (`reporting` and `ETL`) using the intradatabase plan in `reppdb`.

Quiz



What happens to the leftover allocation if a consumer group does not require its full resource allocation?

- a. It remains unused.
- b. It is divided equally among other consumer groups.
- c. It is allocated to other active consumer groups, according to the resource plan.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: c

Quiz



Which of the following conditions are required for IORM to arbitrate the allocation of I/O resources?

- a. The IORM plan objective must be set to a value other than basic.
- b. The IORM plan must be active.
- c. More than one consumer group or database must be active.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

All of the conditions listed in this question must be present for IORM to arbitrate the allocation of I/O resources.

Quiz



In which order are the different I/O resource plans applied to allocate I/O resources?

- a. Category, intradatabase, interdatabase
- b. Interdatabase, category, intradatabase
- c. Category, interdatabase, intradatabase
- d. Interdatabase, intradatabase, category
- e. Intradatabase, interdatabase, category

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: c

Quiz



You can create categories using the CellCLI utility:

- a. True
- b. False

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b

You can create your own categories using the `CREATE_CATEGORY` procedure in the `DBMS_RESOURCE_MANAGER` package, and then assign your category to a consumer group using the `CREATE_CONSUMER_GROUP` or `UPDATE_CONSUMER_GROUP` procedures.

You can then manage I/O resources based on categories by creating a category plan. The category plan can be created using the CellCLI utility.

Summary

In this lesson, you should have learned how to use Exadata I/O Resource Management to manage workloads within a database and across multiple databases.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Additional Resources

- Lesson demonstrations
 - Intradatabase I/O Resource Management
<https://apex.oracle.com/pls/apex/f?p=44785:24:5154068666419939:::24:P24> CONTENT ID,P24 PREV PAGE:5049,24
 - Interdatabase I/O Resource Management
<https://apex.oracle.com/pls/apex/f?p=44785:24:5154068666419939:::24:P24> CONTENT ID,P24 PREV PAGE:5050,24



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

8

Recommendations for Optimizing Database Performance

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to describe the recommendations for optimizing database performance in conjunction with Exadata Database Machine.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Optimizing Performance

- Start with best practices for ASM and Oracle Database.
- Areas for special consideration:
 - Flash memory usage
 - Compression usage
 - Index usage
 - ASM allocation unit size
 - Extent size
 - Exadata specific system statistics
 - Setting the I/O timeout threshold



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Optimizing the performance of your Exadata environment begins with following good practices for database design and application development. From an administration viewpoint, you should continue to follow the best practices for ASM and Oracle Database in conjunction with advice and statistics provided by tools such as SQL monitor and SQL Tuning Advisor.

In addition, there are several areas for special consideration listed in the slide. These are described in the remainder of this lesson.

Flash Memory Usage

- On Extreme Flash cells, data resides on flash drives:
 - Smart Flash Cache cannot improve performance for frequently accessed data
- On all other cells, flash memory is primarily used for:
 - Exadata Smart Flash Cache:
 - Speeds up access to frequently accessed data
 - Uses most of the available flash memory by default
 - Can be managed automatically for maximum efficiency
 - Users can provide optional hints to influence caching priorities.
 - Administrators can disable Smart Flash Cache for specific databases.
 - Can be configured in write-through mode or write-back mode
 - Beneficial for OLTP and Data Warehouse workloads
 - Exadata Smart Flash Log:
 - Small (512 MB) high-performance temporary store for redo log records
 - Managed automatically by Exadata Storage Server software
 - Administrators can disable Smart Flash Log for specific databases.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

On Extreme Flash Exadata Storage Servers, data permanently resides on high-performance flash drives. Consequently, there is no need for an additional flash-based cache to speed up access to frequently accessed data.

On all other cells, nearly all of the flash memory is configured as Exadata Smart Flash Cache. Exadata Smart Flash Cache provides a caching mechanism for frequently accessed data on each Exadata cell. By default, Exadata Smart Flash Cache operates in write-through mode; however, Exadata Smart Flash Cache can also be configured to operate in write-back mode. Both modes are discussed in detail later in this lesson.

In addition, Exadata Smart Flash Log provides a mechanism for improving the latency of database redo log write operations. Exadata Smart Flash Log uses a small portion of the high-performance flash memory on Exadata Storage Server as temporary storage to provide low latency redo log writes. By default, Exadata Smart Flash Log uses a total of 512 MB on each Exadata Storage Server. The default Exadata Smart Flash Log can be dropped and recreated using a different size. However, the size of the default Exadata Smart Flash Log is sufficient for most uses. Exadata Smart Flash Log is managed automatically by Exadata Storage Server.

Finally, the ability for a database to use Exadata Smart Flash Cache or Exadata Smart Flash Log can be controlled using the I/O Resource Management (IORM) Plan. See the lesson titled “I/O Resource Management” for further details.

Write Back Flash Cache on Extreme Flash Cells

By default, Smart Flash Cache on Extreme Flash cells:

- Runs in write-back mode
 - Can be changed but not recommended
- Consumes 5 percent of the flash space
 - Approximately 720 GB on each cell
- Is used for:
 - Columnar caching
 - Write I/O latency capping*
 - Fast file creation*

* Requires Smart Flash Cache to run in write-back mode



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

On Extreme Flash cells, Smart Flash Cache runs in write-back mode by default, and consumes 5 percent of the flash space (approximately 720 GB on each cell). Flash cache on Extreme Flash cells is not used as a block cache because user grid disks are already created on flash, and therefore additional caching is not beneficial. However, Smart Flash Cache is still used for the following operations on Extreme Flash cells:

- Columnar caching, which caches Hybrid Columnar Compression (HCC) table data on flash cache in a pure columnar layout.
- Write I/O latency capping, which cancels write I/O operations to a temporarily stalled flash device, and redirects the write to flash cache on another healthy flash device.
- Fast data file creation, which persists the metadata about the blocks in the write-back flash cache, eliminating the actual formatting writes to user grid disks.

On Extreme Flash cells, you can configure Smart Flash Cache in write-through mode; however, this is not recommended as there is no specific benefit. Also, write I/O latency capping and fast data file creation require write-back flash cache to be enabled. Columnar caching works with either flash cache mode.

Influencing Caching Priorities

- Users can influence caching priorities using the CELL_FLASH_CACHE storage attribute:
 - DEFAULT uses Smart Flash Cache normally.
 - KEEP uses Smart Flash Cache more aggressively.
 - DEFAULT objects cannot steal cache space from KEEP objects.
 - KEEP objects can consume up to 80% of the cache.
 - Unused KEEP object data is periodically flushed to disk.
 - NONE specifies that Smart Flash Cache is not used.

```
SQL> CREATE TABLE calldetail ( ... )
      STORAGE (CELL_FLASH_CACHE KEEP);
```

```
SQL> ALTER TABLE calldetail STORAGE (CELL_FLASH_CACHE NONE);
```

- Recommendation: Use DEFAULT unless there is a specific reason otherwise.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

On non-Extreme Flash cells, Exadata Smart Flash Cache automatically focuses on caching frequently accessed data and index blocks, along with other performance critical information. DBAs can influence caching priorities using the CELL_FLASH_CACHE storage attribute for specific database objects. The possible settings are:

- DEFAULT specifies that the associated database object is cached using the standard least-recently-used caching policy of Exadata Smart Flash Cache. This is the default value for CELL_FLASH_CACHE when the storage clause is not specified, and this is the recommended setting for all database objects unless you have specific knowledge and requirements that dictate otherwise.
- KEEP specifies that the associated database object has a higher caching priority. KEEP objects have priority over DEFAULT objects so that new data from a DEFAULT object will not push out cached data from any KEEP objects. To prevent KEEP objects from monopolizing the cache, they are allowed to occupy no more than 80% of the total cache size. Also, to prevent unused KEEP objects from indefinitely occupying the cache, they are subject to an additional aging policy, which periodically flushes unused KEEP object data.
- NONE ensures that the corresponding database object is never cached in Exadata Smart Flash Cache. This allows flash cache space to be reserved for other objects.

Choosing the Flash Cache Mode for Non-Extreme Flash Cells

Write-through mode	Write-back mode
Writes go to disk through the disk controller cache	Writes go directly to flash
Cached reads free up disks and help to improve write performance	Data is automatically flushed to disk to free up space for more frequently used data
Mirror copies are generally not cached resulting in a larger effective cache	All mirror copies are written to cache effectively reducing the cache size
Broadly applicable for most applications	Especially useful for write-intensive applications
Use write-through mode by default	Enable write-back mode if write I/Os are a performance bottleneck: <ul style="list-style-type: none">High free buffer waits in the database wait event statisticsHigh disk I/O latency and a large percentage of writes



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

For non-Extreme Flash Exadata Storage Servers, Smart Flash Cache is configured to run in write-through mode by default. In this mode, writes are directed to disk through a cache on the Exadata Storage Server disk controller. Because of this, write I/O latency is typically not a performance issue. Consequently, write-through flash cache offers excellent performance for most applications.

For extremely write-intensive applications, the write volume can flood the disk controller cache, rendering it effectively useless. Write-back flash cache provides a solution for write-intensive applications. When Smart Flash Cache operates in write-back mode, writes go directly to flash and data is only written to disk when it is aged out of the cache. As a result, the most commonly read and written data is maintained in Smart Flash Cache while less accessed data is maintained on disk. Note that applications that are not bottlenecked on writes will see little or no benefit from the extra write throughput enabled by write-back Smart Flash Cache. Also, in write-back mode all mirror copies of data are written to Smart Flash Cache, which effectively reduces the cache size when compared to write-through mode.

Because of the different characteristics of each cache mode, it is recommended to use write-through mode by default and only enable write-back mode when write I/Os are observed as a performance bottleneck. The best way to determine a write bottleneck is to look for free buffer waits in the database wait event statistics. Administrators can also check Exadata Storage Server metrics for high disk I/O latencies and a large percentage of writes.

Setting the Flash Cache Mode

- Providing write-back mode:

```
CellCLI> DROP FLASHCACHE
CellCLI> ALTER CELL SHUTDOWN SERVICES CELLSRV
CellCLI> ALTER CELL flashCacheMode = WriteBack
CellCLI> ALTER CELL STARTUP SERVICES CELLSRV
CellCLI> CREATE FLASHCACHE ALL
```

- Causing write-through mode:

```
CellCLI> ALTER FLASHCACHE ALL FLUSH
CellCLI> DROP FLASHCACHE
CellCLI> ALTER CELL SHUTDOWN SERVICES CELLSRV
CellCLI> ALTER CELL flashCacheMode = WriteThrough
CellCLI> ALTER CELL STARTUP SERVICES CELLSRV
CellCLI> CREATE FLASHCACHE ALL
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Setting the flash cache mode is achieved by using the `ALTER CELL` command to set the `flashCacheMode` cell attribute. To change the `flashCacheMode` cell attribute you must first drop the existing Exadata Smart Flash Cache and shutdown `CELLSRV`.

Note that changing from write-back mode to write-through mode also requires that the flash cache is flushed before it is dropped. Flushing the flash cache temporarily disables it and can be a time-consuming operation.

Setting the flash cache mode can be done in a rolling manner, one cell at a time, or all cells at once. Changing the flash cache mode in all cells at the same time requires down time for all the databases, ASM and Oracle Clusterware.

Changing the flash cache mode in a rolling manner removes the need for downtime however a rolling change requires greater care to ensure that all the steps are performed in the correct order. Also, because the flash cache is temporarily disabled during the change, application performance is likely to be impacted during a rolling change. Finally, flushing the flash cache while databases remain active can take up to 2 hours per cell, depending on the prevailing workload.

The examples in the slide show the minimum set of commands required to set the flash cache mode. Additional verification and status-checking commands are not shown. Refer to My Oracle Support note 1500257.1 for the detailed procedures to implement rolling and non-rolling changes.

Table Compression Usage

Compression Method	CREATE/ALTER TABLE Syntax	Compression Ratio	CPU Consumption	Typical Applications
Basic Compression	COMPRESS or ROW STORE COMPRESS [BASIC]	High for direct path inserts Conventional path inserts and updates are not compressed	Low: Oracle Database performs compression and decompression	Analytics
Advanced Compression	ROW STORE COMPRESS ADVANCED	High for all transaction types	Low: Oracle Database performs compression and decompression, compression for DML is performed in batches	OLTP and Analytics
Warehouse Compression	COLUMN STORE COMPRESS FOR QUERY [LOW HIGH]	Higher for direct path inserts High for conventional path inserts and updates	Medium: Decompression is performed by Exadata Storage Server in some cases	Analytics
Online Archival Compression	COLUMN STORE COMPRESS FOR ARCHIVE [LOW HIGH]	Highest for direct path inserts High for conventional path inserts and updates	High: Decompression is performed by Exadata Storage Server in some cases	Archiving



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Many people first think of data compression as they approach the capacity limits of their storage. Commonly, data compression is seen as a performance overhead, which must be tolerated to deliver extra storage capacity. However, in cases where I/O bandwidth is limited, data compression can be an effective tool to increase performance by using available CPU capacity to effectively increase the I/O throughput of a storage system.

Oracle Database provides the following modes of table compression:

- Basic compression compresses data at the Oracle block level. It allows more data to be stored in each block by replacing duplicate values with a smaller symbolic representation. For example, if the numeric value 99999 was present 50 times in a block of data it could be replaced by 50 occurrences of the # symbol along with an entry in a symbol table (also stored in the block) defining the use of the symbol. The degree of compression depends on the level of duplication in each block. Higher compression ratios can be achieved by using larger block sizes or by sorting data to increase the coincidence of duplicate values. Data remains in row-major format where the columns in each row are stored together. Compression occurs when data is loaded using a direct-path load operation, such as CREATE TABLE AS SELECT or direct-path SQL*Loader. Tables that use basic compression support DML operations; however, any data that is inserted or updated remains uncompressed. Basic compression is useful in analytics where the compressed data is subject to minimal change.

- Advanced compression uses the same compression technique as basic compression, but provides support for compressing transactional data from all DML statements, not just direct-path loads. This makes it useful in analytics and online transaction processing (OLTP) environments. Instead of compressing changes as they are written, all the changes to a data block are compressed as a batch whenever a change causes the block to become fuller than an internally controlled threshold. In other words, only the DML statements that trigger compression experience the overhead of compression for the associated data blocks, however most DML statements experience no compression overhead at all. This means that compression can be implemented on OLTP systems without significantly impacting the overall transactional throughput of the system.

In addition to the basic and advanced compression modes provided by Oracle Database, Exadata Storage Server provides Hybrid Columnar Compression. Hybrid Columnar Compression technology uses a modified form of columnar storage instead of row-major storage. Sets of rows are stored in an internal structure called a compression unit. Within each compression unit, the values for each column are stored together along with metadata that maps the values to the rows. Compression is achieved by replacing repeating values with smaller symbolic references. Because a compression unit is much larger than an Oracle block, and because column organization brings similar values together, Hybrid Columnar Compression can deliver much better compression ratios than both basic and advanced compression. The best rates of compression are achieved using direct path loads.

Hybrid Columnar Compression provides a choice of compression modes to achieve the proper trade-off between disk usage and CPU overhead:

- **Warehouse compression:** This type of compression is optimized for query performance, and is intended for data warehousing and analytics.
- **Online archival compression:** This type of compression is optimized for maximum compression ratios, and is intended for historical data and data that does not change.

Hybrid Columnar Compression supports DML operations on compressed data. However, updated rows and rows added using conventional path inserts are placed into single-block compression units which yield a lower compression ratio than direct-path loads. In addition, updates and deletes on tables using Hybrid Columnar Compression require the entire compression unit to be locked, which may impact concurrency. Finally, updates to rows using Hybrid Columnar Compression cause rowids to change. As a result, Hybrid Columnar Compression is recommended for situations where data changes are infrequent or where data sets are reloaded rather than substantially changed.

In conclusion, Hybrid Columnar Compression makes effective use of Exadata Storage Server hardware to deliver the highest levels of compression for data in an Oracle database. It is best suited to cases where the data is not subject to substantial change. For transactional data sets, you should consider advanced compression instead of Hybrid Columnar Compression. In all cases, you should be aware of the relative merits and overheads associated with each compression type to choose the best approach for your situation.

Index Usage

- Queries that require indexes on a previous system might perform better using Exadata and Smart Scan.
- Consider removing indexes where Smart Scan delivers acceptable performance.
- Removing unnecessary indexes will:
 - Improve DML performance
 - Save storage space
- Test the effect of removing indexes by making them invisible:

```
SQL> ALTER INDEX <index_name> INVISIBLE;
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Some queries that require indexes when using conventional storage will perform acceptably without indexes using Exadata. Review your queries that use indexes to determine if they would run acceptably using Smart Scan.

To test if queries run acceptably without an index, you can make the index invisible to the optimizer. An invisible index still exists and is maintained by DML operations, but it is not used by the optimizer for queries. To make an index invisible, use the following command:

```
ALTER INDEX <index_name> INVISIBLE
```

Removing unnecessary indexes aids the performance of DML operations by removing index-related I/Os and maintenance operations such as index rebalancing. Removing unnecessary indexes also saves storage space.

ASM Allocation Unit Size

- By default, ASM uses an allocation unit (AU) size of 1 MB.
- For Exadata storage, the recommended AU size is 4 MB.
 - AU size must be set when a disk group is created.
 - AU size cannot be altered after a disk group is created.
 - AU size is set using the `AU_SIZE` disk group attribute.

```
SQL> CREATE DISKGROUP data NORMAL REDUNDANCY
      DISK 'o/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///*/*/*/*/*/*/*/*/*/*/*/*/*/*/*///*/*/*/*/*/*/*/*/*/*/* Oracle Confidential */
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

To achieve fast disk scan rates, it is important to lay out segments with at least 4 MB of contiguous space. This allows disk scans to read 4 MB of data before performing another seek at a different location on disk. To ensure that segments are laid out with 4 MB of contiguous space, set the Oracle ASM allocation unit size to 4 MB. The allocation unit can be set with the `AU_SIZE` disk group attribute when creating the disk group.

Extent Size

- Segments should have extents that are a multiple of the ASM AU size:
 - Stops needless proliferation of small extents in the database
 - Optimizes I/O by aligning extent and ASM AU boundaries

```
SQL> CREATE TABLE t1
  (col1 NUMBER(6), col2 VARCHAR2(10))
  STORAGE ( INITIAL 8M MAXSIZE 1G );
```

```
SQL> CREATE BIGFILE TABLESPACE ts1
  DATAFILE '+DATA' SIZE 100G
  DEFAULT STORAGE ( INITIAL 8M NEXT 8M );
```

- For very large segments, it is optimal to stripe each extent across all of the available disks



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Extent size is managed automatically in locally managed tablespaces. This option automatically increases the size of the extent depending on segment size, available free space in the tablespace, and other factors. By default, the extent size starts at 64 KB and increases in increments of 1 MB, 8 MB, or 64 MB. Generally speaking, it is recommended that large segments should be defined with larger than default initial extents to minimize the needless proliferation of small extents in the database.

For large segments on Exadata, the recommendation is to have extents that are a multiple of 4 MB, which is the recommended ASM allocation unit size. This optimizes I/O performance by aligning extent boundaries with ASM allocation unit boundaries.

For very large segments, it is optimal to effectively stripe each extent across all of the available disks in the disk group. To achieve this, you should set your extent sizes using the following formula:

$$\text{Extent Size} = \text{ASM AU Size} * \text{Number of Disks in Disk Group}$$

For example, a high capacity storage server contains 12 disks, so an Exadata system with 3 high capacity cells contains 36 disks. Assuming that your disk groups span all 36 disks, which is typically true, then the optimal extent size for very large segments is 144 GB in this case.

Use the INITIAL and NEXT storage parameters to set extent sizes. They can be set specifically for individual objects or at the tablespace level for locally managed tablespaces. The slide shows examples of both options.

Exadata Specific System Statistics

- Gather Exadata specific system statistics:

```
SQL> exec dbms_stats.gather_system_stats('EXADATA');
```

- Enables the optimizer to more accurately cost operations using actual performance information:
 - CPU speed
 - I/O Performance
- Sets multi block read count (MBRC) correctly for Exadata
- Requires at least Oracle Database version 11.2.0.2 BP 18 or 11.2.0.3 BP 8
- Recommended for all new databases
 - Test thoroughly before changing existing databases.
 - Databases with stable good plans do not require a change.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

New Exadata specific system statistics are introduced commencing with Oracle Database version 11.2.0.2 bundle patch 18 and 11.2.0.3 bundle patch 8.

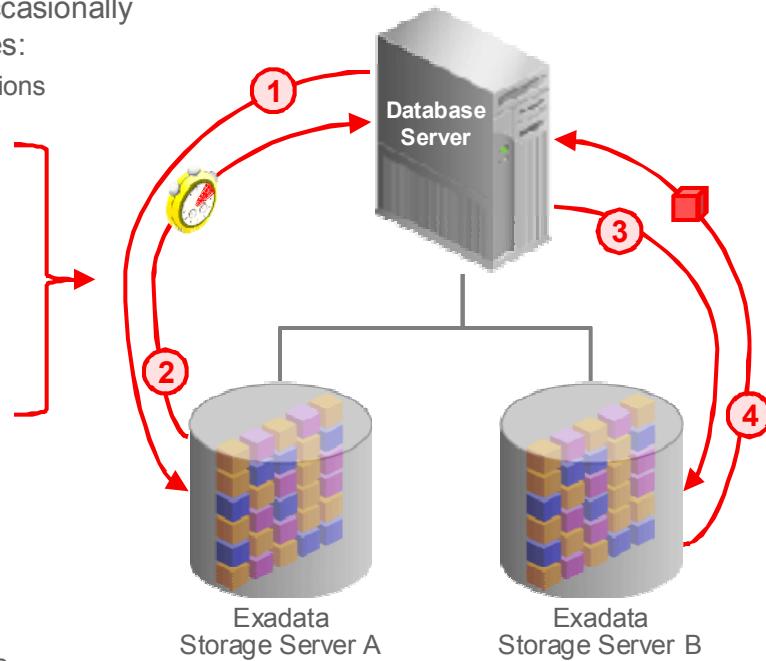
Gathering Exadata specific system statistics ensures that the optimizer is aware of the specific performance characteristics associated with Exadata, which allows it to choose optimal execution plans.

In addition to setting various system statistics relating to system CPU speed and I/O performance, gathering system statistics in Exadata mode sets the multi block read count (MBRC) statistic correctly for Exadata. Before this enhancement, the optimizer assumed an MBRC value of 8 blocks, or 64 KB assuming an 8 KB database block size. For Exadata, the true I/O size used during scans is based on the ASM allocation unit (AU) size, which is typically 4 MB. This difference causes the optimizer to incorrectly judge the cost of scan operations on Exadata, which can lead to an alternative execution plan being chosen when a scan would be the optimal approach.

It is recommended to gather Exadata specific system statistics for all new databases on Exadata. For existing databases, administrators are advised to consider current application performance before making any changes. If the application performance is stable and good then no change is required. If there is evidence to suggest that current application performance is impacted by suboptimal plans, then thoroughly test your system with the updated system statistics before finalizing the change.

Exadata I/O Latency Capping

- Disk and flash devices can occasionally exhibit unusually high latencies:
 - Internal maintenance operations
 - Precursor to failure
- When high read latency is detected:
 - Exadata Storage server messages the database server that initiated the I/O
 - The database redirects the read I/O to a mirror copy of data
- When high write latency to flash is detected:
 - Writes to slow flash devices are automatically redirected to another flash device on the same cell
 - Requires Smart Flash Cache in write-back mode



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Disks drives or flash devices can, on rare occasions, exhibit high latencies for a small amount of time while an internal maintenance operation is running. For example, higher than normal latencies can occur on flash devices when data is reorganized by internal wear levelling operations. In addition, drives that are close to failing can sometimes exhibit high latencies before they fail. Exadata I/O latency capping masks these very rare latency spikes by redirecting read I/O operations to a mirror copy.

Oracle Exadata Storage Server Software automatically redirects read I/O operations to another cell when a storage device exhibits unusually high latency. This is performed by returning a message to the database that initiated the read I/O. The database then redirects the I/O to another mirror copy of the data. Any I/Os issued to the last valid mirror copy of the data is not redirected. This feature is automatic and transparent to users. Note that this features requires Exadata release 11.2.3.3.1 or later, or Exadata release 12.1.1.1.1 or later. Also, Oracle Database version 11.2.0.4.8 or later is required.

In release 12.1.2.1.0, Exadata Storage Server software automatically redirects high latency write I/O operations to another healthy flash device on the same cell, thereby capping the latency. After the write is successfully completed on another healthy flash device, the write I/O is acknowledged as successful to the database. This feature requires write back flash cache to be enabled on the cell.

Setting the Exadata Cell I/O Timeout Threshold

- Use the ALTER CELL command to set the iotimeoutthreshold attribute:

```
CellCLI> ALTER CELL iotimeoutthreshold = '5s'
```

- Read I/O that takes longer than the threshold is cancelled and redirected to a mirror copy of data
- Designed to minimize abnormal read latencies
 - Cannot boost normal system performance
- Setting the threshold too low can negatively impact system performance
 - Review AWR reports to determine a reasonable setting



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In addition to Exadata I/O latency capping, an I/O timeout threshold can be configured for each Exadata Storage Server by using the ALTER CELL command to set the iotimeoutthreshold attribute. Cell read I/O that takes longer than the defined threshold is cancelled and redirected to a mirror copy of the data. The slide shows a command example.

Note that this feature is not designed to improve performance in a system that is operating normally. Rather, it is designed to minimise the adverse impact of an abnormal situation. Note also, that setting the timeout threshold too low can negatively impact system performance because timed-out I/O operations must be reexecuted. Oracle recommends reviewing the Automatic Workload Repository (AWR) reports corresponding to times of peak I/O loads, and setting the threshold value to a value higher than the normal peak I/O latency plus a sufficient safety margin.

Quiz



Under what circumstances should you enable write-back Smart Flash Cache?

- a. Always. Write-back flash cache offers superior performance for all applications.
- b. For any application that performs more writes than reads.
- c. For situations where write IOs are a performance bottleneck.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: c

Write-through flash cache provides excellent performance for most applications and makes the most efficient use of the available flash memory space. Write-back flash cache provides increased performance for situations where write I/Os are a performance bottleneck. However, applications that are not bottlenecked on writes will see little or no benefit from the extra write throughput enabled by write-back Smart Flash Cache. Also, in write-back mode all mirror copies of data are written to Smart Flash Cache, which effectively reduces the cache size when compared to write-through mode.

Quiz



Which of the following CREATE TABLE compression clauses are available for use only in conjunction with Exadata?

- a. ROW STORE COMPRESS BASIC
- b. ROW STORE COMPRESS FOR OLTP
- c. COLUMN STORE COMPRESS FOR QUERY
- d. COLUMN STORE COMPRESS FOR ARCHIVE

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: c, d

Summary

In this lesson, you should have learned how to describe the recommendations for optimizing database performance in conjunction with Exadata Database Machine.



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 8: Overview

Optimizing Database Performance with Exadata

In these practices, you will explore the following performance optimization techniques and technologies:

- Configuring write back flash cache
- Using Hybrid Columnar Compression
- Testing index elimination



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

9

Using Smart Scan

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe Smart Scan and the query processing that can be offloaded to Exadata Storage Server
- Describe the requirements for Smart Scan
- Describe the circumstances that prevent using Smart Scan
- Identify Smart Scan in SQL execution plans
- Use database statistics and wait events to confirm how queries are processed



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Exadata Smart Scan: Overview

Smart Scan includes:

- Full Table and Fast Full Index Scans: Scans are performed inside Exadata Storage Server, rather than transporting all the data to the database server.
- Predicate filtering: Only the requested rows are returned to the database server, rather than all the rows in a table.
- Column filtering: Only the requested columns are returned to the database server, rather than all the table columns.
- Join filtering: Join processing using Bloom filters are offloaded to Exadata Storage Server.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

One of the most powerful features of Exadata is that data search and retrieval processing can be offloaded to the Exadata Storage Servers. Using this feature, Oracle Database can optimize the performance of operations that perform table and index scans by performing the scans inside Exadata Storage Server rather than transporting all the data to the database server. This principle can also be applied to encrypted data and compressed data. In addition, various aspects of SQL processing can be offloaded to the Exadata Storage Server so that it is performed close to the data and so that the amount of data transported back to the database server is minimized.

The slide lists the key database functions that are integrated with Exadata Storage Server. The term Smart Scan is also used to describe instances when these functions are performed by Exadata Storage Server.

This lesson focuses on the requirements for performing Smart Scan processing, and how you can use core SQL monitoring capabilities within Oracle Database to identify Smart Scans.

Smart Scan Requirements

Smart Scan is not governed by the optimizer, but it is influenced by the results of query optimization.

- Query-specific requirements:
 - Smart Scan is possible only for full segment scans.
 - Smart Scan can only be used for direct-path reads:
 - Direct-path reads are automatically used for parallel queries.
 - Direct-path reads may be used for serial queries:
 - They are not used by default for small serial scans.
 - Use `_serial_direct_read=TRUE` to force direct-path reads.
- Additional general requirements:
 - Smart Scan must be enabled within the database.
 - Segments must be stored in appropriately configured disk groups.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Smart Scan optimization is a runtime decision. It is not integrated with the Oracle optimizer however it is influenced by the results of query optimization. In other words, the decision regarding whether or not to use Smart Scan is not made by the optimizer, but the optimizer does indirectly determine when Smart Scan can be used.

The following query-specific requirements must be met before Smart Scan is considered:

- Smart Scan is possible only for full segment scans; that is full table scans, fast full index scans and fast full bitmap index scans.
- Smart Scan can only be used in conjunction with direct-path reads. Direct path reads are used by parallel operations, so any parallel query is automatically a potential candidate for Smart Scan. Serial operations can do direct reads too, depending on factors such as the table size and the state of the buffer cache. By default, direct-path reads are not used for tables that are considered too small. However direct-path reads can be forced for serial access by setting `_serial_direct_read=TRUE` at either the system or session level.

In addition to the query-specific requirements, the following general requirements must also be met to enable Smart Scan:

- Smart Scan must be enabled within the database. The CELL_OFFLOAD_PROCESSING initialization parameter controls Smart Scan. The default value of the parameter is TRUE, meaning that Smart Scan is enabled by default.
- Each segment being scanned must be on a disk group that is completely stored on Exadata cells. The disk group must also have the following disk group attribute settings:

```
'compatible.rdbms' = '11.2.0.0.0' (or later)  
'compatible.asm' = '11.2.0.0.0' (or later)  
'cell.smart_scan_capable' = 'TRUE'
```

Situations Preventing Smart Scan

Smart Scan cannot be used in these circumstances:

- Scan on a clustered table
- Scan on an index-organized table
- Fast full scan on a compressed index
- Fast full scan on a reverse key indexes
- Table has row-level dependency tracking enabled
- ORA_ROWSCN pseudocolumn is being fetched
- Optimizer wants the scan to return rows in ROWID order
- Command is CREATE INDEX using NOSORT
- LOB or LONG column is being selected or queried
- SELECT . . . VERSIONS flashback query is being executed
- More than 255 columns are referenced in the query
- Data is encrypted and cell-based decryption is disabled
- To evaluate a predicate based on a virtual column



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide lists specific circumstances where Smart Scan cannot be used. The following provides additional information for some of these cases:

- More than 255 columns are referenced in the query: This restriction only applies if the query involves tables that are not compressed using Hybrid Columnar Compression. Queries on tables compressed using Hybrid Columnar Compression can be offloaded even if they reference more than 255 columns.
- The data is encrypted and cell-based decryption is disabled: In order for Exadata Storage Server to perform decryption, Oracle Database needs to send the decryption key each cell. If there are security concerns about keys being shipped across the storage network, cell-based decryption can be disabled by setting the `CELL_OFFLOAD_DECRYPTION` initialization parameter to FALSE.

Monitoring Smart Scan in SQL Execution Plans

Relevant Initialization Parameters:

- CELL_OFFLOAD_PROCESSING
 - TRUE | FALSE
 - Enables or disables Smart Scan and other smart storage capabilities
 - Dynamically modifiable at the session or system level using ALTER SESSION or ALTER SYSTEM
 - Specifiable at the statement level using the OPT_PARAM hint
- CELL_OFFLOAD_PLAN_DISPLAY
 - NEVER | AUTO | ALWAYS
 - Allows execution plan to show offloaded predicates
 - Dynamically modifiable at the session or system level using ALTER SESSION or ALTER SYSTEM



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The CELL_OFFLOAD_PROCESSING initialization parameter controls Smart Scan. The default value of the parameter is TRUE which means that Smart Scan is enabled. If it is set to FALSE, Smart Scan is disabled and the database uses Exadata storage to serve data blocks similar to traditional storage. To enable Smart Scan for a particular SQL statement, use the OPT_PARAM hint as shown in the following example:

```
SELECT /*+ OPT_PARAM('cell_offload_processing' 'true') */ ...
```

The CELL_OFFLOAD_PLAN_DISPLAY initialization parameter determines whether the SQL EXPLAIN PLAN statement displays the predicates that can be evaluated by Exadata Storage Server as STORAGE predicates for a given SQL statement. The possible values are:

- AUTO instructs the SQL EXPLAIN PLAN statement to display the predicates that can be evaluated as STORAGE only if a cell is present and if a table is on the cell. AUTO is the default setting.
- ALWAYS produces changes to the SQL EXPLAIN PLAN statement whether or not Exadata storage is present or the table is on the cell. You can use this setting to identify statements that are candidates for offloading before migrating to Exadata.
- NEVER produces no changes to the SQL EXPLAIN PLAN statement due to Exadata. This may be desirable, for example, if you wrote tools that process execution plan output and these tools have not been updated to deal with the updated syntax, or when comparing plans from Exadata with plans from your previous system.

Smart Scan Execution Plan: Example

```
SQL> explain plan for select count(*) from customers where cust_valid = 'A';
Explained.

SQL> select * from table(dbms_xplan.display);

-----| Id | Operation | Name | Rows | Bytes | Cost (%CPU) |
-----| 0 | SELECT STATEMENT | | 1 | 2 | 627K (1) |
| 1 | SORT AGGREGATE | | 1 | 2 | |
|* 2 | TABLE ACCESS STORAGE FULL | CUSTOMERS | 38M | 73M | 627K (1) |

Predicate Information (identified by operation id):
-----
2 - storage("CUST_VALID"='A')
      filter("CUST_VALID"='A')
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows a basic example of a query plan that indicates the use of Smart Scan.

The TABLE ACCESS STORAGE FULL operation indicates that Smart Scan could be used to scan the CUSTOMERS table. The plan also shows evidence of row filtering. In this example, the predicate "CUST_VALID" = 'A' could be evaluated inside Exadata Storage Server.

Note that the execution plan shows no evidence of parallel query, so it is likely that the table in this example is large enough to prompt the use of direct-path reads.

Note also that seeing such a query plan does not guarantee the use of Smart Scan. For example, if the CELL_OFFLOAD_PLAN_DISPLAY initialization parameter is set to ALWAYS, you would always see such a plan, regardless of whether or not Smart Scan is actually used. Other situations are discussed later in the lesson.

Smart Scan Execution Plan: Example

```
SQL> explain plan for select count(*) from customers where cust_id > '10000';
```

Explained.

```
SQL> select * from table(dbms_xplan.display);
```

Id	Operation	Name	Rows	Bytes
0	SELECT STATEMENT		1	6
1	SORT AGGREGATE		1	6
2	PX COORDINATOR		1	6
3	PX SEND QC (RANDOM)	:TQ10000	1	6
4	SORT AGGREGATE		1	6
5	PX BLOCK ITERATOR		77M	443M
*	INDEX STORAGE FAST FULL SCAN	CUSTOMERS_PK	77M	443M

Predicate Information (identified by operation id) :

```
6 - storage ("CUST_ID">>10000)
      filter("CUST_ID">>10000)
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This slide shows another Smart Scan query plan example. This time, offloading of a fast full index scan is indicated. Row filtering is also pushed down to the storage servers.

Unlike the previous example, this example indicates the use of parallel query, which automatically implies the use of direct-path reads.

Example of a Situation Preventing Smart Scan

```
SQL> explain plan for select count(*) from cust_iot where cust_id > '10000';
```

Explained.

```
SQL> select * from table(dbms_xplan.display);
```

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)	Time
0	SELECT STATEMENT		1	13	21232	(1)	00:04:15
1	SORT AGGREGATE		1	13			
*	2	INDEX RANGE SCAN CUST_PK	86M	1071M	21232	(1)	00:04:15

Predicate Information (identified by operation id) :

```
2 - access ("CUST_ID">>10000)
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This slide shows almost exactly the same example as the previous slide. The only difference is that the table being queried in this example is an index-organized table with the CUST_ID column defined as the primary key. Since Smart Scan cannot be applied to index-organized tables, the execution plan contains no storage operations.

Smart Scan Join Processing with Bloom Filters

- A Bloom filter is a data structure that can be used to test if an element is a member of a set.
- Bloom filter properties:
 - The amount of data used in the Bloom filter is much smaller than the set being tested.
 - The time required to check whether an element is a member of the set is constant.
 - False positives are possible but their frequency can be managed.
 - False negatives are not possible.
- Since Oracle 10g Release 2, Bloom filters have been used to optimize parallel joins.
- With Exadata, the Bloom filter can be processed by the storage server, reducing the amount of data unnecessarily transported to the database server.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

A Bloom filter, conceived by Burton Howard Bloom in 1970, is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. The properties of a Bloom filter make it a very efficient way of determining which values are *not* in a set. This is very useful for processing join conditions where a significant proportion of the data does not fulfil the join criteria.

Oracle Database 10g Release 2 first used Bloom filters to optimize parallel join operations. When two tables are joined via a hash join, the first table (typically the smaller table) is scanned and the rows that satisfy the WHERE clause predicates (for that table) are used to create a hash table. During the hash table creation, a Bloom filter bit string is also created based on the join column. The bit string is then sent as an additional predicate to the second table scan. After the WHERE clause predicates relating to the second table are applied, the resulting rows are tested using the Bloom filter. Any rows rejected by the Bloom filter must fail the join criteria and are discarded. Any rows that match by using the Bloom filter are sent to the hash join.

With Exadata, the Bloom filter is passed to the storage servers as an additional predicate. Processing the Bloom filter inside Exadata Storage Server can reduce the amount of data transported to the database server to process a join, which in turn can speed up query performance.

Smart Scan Join Filtering: Example

```
SQL> SELECT AVG(s.amount_sold) FROM customers cu, sales s
  2 WHERE cu.cust_id = s.cust_id
  3 AND cu.cust_credit_limit > 5000;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		1	19	55979 (1)	00:11:12			
1	SORT AGGREGATE		1	19					
2	PX COORDINATOR								
3	PX SEND QC (RANDOM)	:TQ10002	1	19			Q1,02	P->S	QC (RAND)
4	SORT AGGREGATE		1	19			Q1,02	PCWP	
* 5	HASH JOIN		577M	10g	55979 (1)	00:11:12	Q1,02	PCWP	
6	JOIN FILTER CREATE	:BF0000	57M	547M	14499 (1)	00:02:54	Q1,02	PCWP	
7	PX RECEIVE		57M	547M	14499 (1)	00:02:54	Q1,02	PCWP	
8	PX SEND HASH	:TQ10000	57M	547M	14499 (1)	00:02:54	Q1,00	P->P	HASH
9	PX BLOCK ITERATOR		57M	547M	14499 (1)	00:02:54	Q1,00	PCWC	
* 10	TABLE ACCESS STORAGE FULL	CUSTOMERS	57M	547M	14499 (1)	00:02:54	Q1,00	PCWP	
11	PX RECEIVE		774M	6651M	24044 (1)	00:04:49	Q1,02	PCWP	
12	PX SEND HASH	:TQ10001	774M	6651M	24044 (1)	00:04:49	Q1,01	P->P	HASH
13	JOIN FILTER USE	:BF0000	774M	6651M	24044 (1)	00:04:49	Q1,01	PCWP	
14	PX BLOCK ITERATOR		774M	6651M	24044 (1)	00:04:49	Q1,01	PCWC	
* 15	TABLE ACCESS STORAGE FULL	SALES	774M	6651M	24044 (1)	00:04:49	Q1,01	PCWP	

Predicate Information (identified by operation id):

```
5 - access ("CU"."CUST_ID"="S"."CUST_ID")
10 - storage("CU"."CUST_CREDIT_LIMIT">5000)
      filter("CU"."CUST_CREDIT_LIMIT">>5000)
15 - storage(SYS_OP_BLOOM_FILTER(:BF0000,"S"."CUST_ID"))
      filter(SYS_OP_BLOOM_FILTER(:BF0000,"S"."CUST_ID"))
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of Smart Scan join filtering using a Bloom filter. The example query joins a table containing approximately 57 million customer records, with a table containing approximately 774 million sales records, to determine the average transaction amount for customers having a credit limit in excess of 5000. The query is processed as follows:

- Smart Scan is used to filter the customer records and retrieve the CUST_ID values for the customers having a credit limit in excess of 5000 (operation 10).
- A Bloom filter is created representing the set of CUST_ID values which match the query from the CUSTOMERS table (operation 6).
- In the absence of Exadata storage the Bloom filter would be applied to data from the SALES table inside a parallel query server process (operation 13). However, with Exadata, the Bloom filter is sent to the storage servers as a predicate and applied as part of a Smart Scan of the SALES table (operation 15). Because the SALES table is quite large, query performance will benefit considerably if a significant amount of I/O is saved by not transporting unnecessary sales records back to the database. Exadata storage server also performs column filtering for the SALES table so that only the CUST_ID and AMOUNT SOLD values are returned.
- The results from the Smart Scan operations on the CUSTOMERS table (operations 6-10) and the SALES table (operations 11-15) are combined using a HASH JOIN (operation 5).
- Finally, the query is serialized and the result is returned (operations 1-4).

Other Situations Affecting Smart Scan

- Seeing **STORAGE** in the execution plan does not guarantee that the query is satisfied using Smart Scan.
- Even when Smart Scan is indicated by the execution plan, other block I/O might also be used:
 - If Exadata Storage Server is not sure that a block is current, it transfers that block read to the buffer cache.
 - If chained or migrated rows are detected, additional non-Smart Scan block reads may be required.
 - I/O for dynamic sampling does not use Smart Scan.
 - If Exadata Storage Server CPU utilization is significantly greater than CPU utilization on the database server, Smart Scan may send additional data to the database server.
 - If all the required data already resides in the database buffer cache, the buffer cache copy is used and no disk I/O is performed.
 - Smart Scan may be disabled if a statement is affected by a storage server quarantine.
- Statistics and wait events can be used to confirm what is happening.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

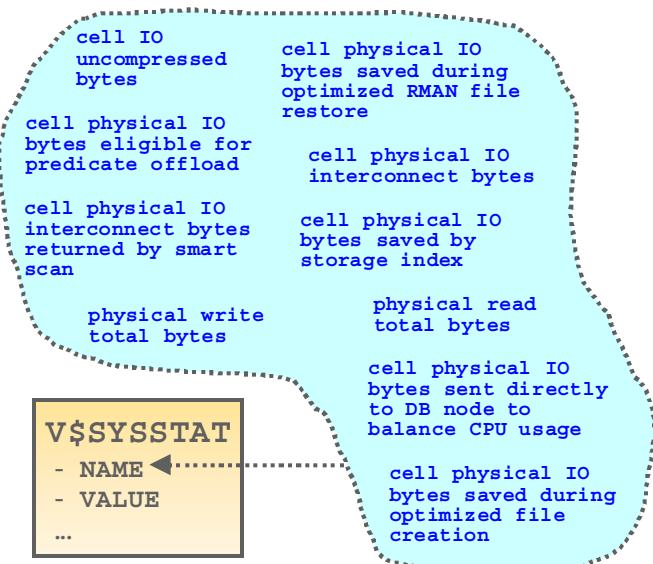
The query execution plan provides an indicator regarding the use of Smart Scan, however seeing a **STORAGE** operation in the plan does not guarantee that the query is satisfied entirely using Smart Scan. Beware that even when Smart Scan is indicated in the execution plan, other block I/O might also be used. Here are some situations where this might occur:

- Because Smart Scan uses direct-path reads, the data being read must be current. If Exadata Storage Server is not sure that a block is current, it transfers the read of that block to the traditional buffer cache read-consistency path. So if you run updates at the same time as queries you will benefit less from Smart Scan.
- The same is also true for indirect rows, that is, chained or migrated rows. To resolve indirect row references additional block reads may be required. So if your tables contain chained or migrated rows, you will benefit less from Smart Scan.
- If the optimizer uses dynamic sampling to formulate the query execution plan, then the sampling I/O does not use Smart Scan even if the query does end up using Smart Scan. In this case you may see Smart Scan I/O mixed in with other block I/O.
- Commencing with Exadata Storage Server version 11.2.2.3.0, if storage server CPU utilization is significantly greater than CPU utilization on the database server, then Exadata Storage Server may choose send additional data blocks back to the database server for processing rather than processing them inside the cell.

- If all the data required to process the query already resides in the database buffer cache, the buffer cache copy is used and no I/O is performed.
- If a statement is affected by a storage server quarantine, then Smart Scan may be disabled for that statement on the storage server that contains the quarantine.

Exadata Storage Server Statistics: Overview

```
SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
WHERE s.statistic# = m.statistic# AND
(s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%'
OR s.name LIKE 'cell IO%');
```



V\$SQL
- SQL_TEXT
- PHYSICAL_READ_BYTES
- PHYSICAL_WRITE_BYTES
- IO_INTERCONNECT_BYTES
- IO_CELL_OFFLOAD_ELIGIBLE_BYTES
- IO_CELL_UNCOMPRESSED_BYTES
- IO_CELL_OFFLOAD_RETURNED_BYTES
- OPTIMIZED_PHY_READ_REQUESTS
...

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Numerous cell-specific statistics are recorded in **V\$SYSSTAT** and **V\$SESSTAT**. These statistics can be used to monitor Exadata Storage Server operations at both the system and session level. The statistics can be used to monitor the effectiveness of Smart Scan. There are also statistics relating to Exadata Smart Flash Cache, Hybrid Columnar Compression, storage index, fast file creation, and optimized incremental backups. In addition, other statistics provide the total volume of I/O exchanged over the interconnect and the total volume of physical disk reads and writes. The slide lists a selection of the available statistics.

The query in the slide shows key cell statistics for the current session. Examples of the output from this query are shown on the following pages.

In addition, **V\$SQL** lists statistics on shared SQL areas. It contains statement-level statistics for the volume of physical I/O (reads and writes), the volume of I/O exchanged over the interconnect, along with information relating to the effectiveness of Smart Scan and other Exadata Storage Server features.

For more information on cell-specific statistics, refer to the *Oracle Exadata Storage Server Software User's Guide*.

Exadata Storage Server Wait Events: Overview

```
SELECT DISTINCT event, total_waits, time_waited/100 wait_secs,  
           average_wait/100 avg_wait_secs  
  FROM V$SESSION_EVENT e, V$MYSTAT s  
 WHERE event LIKE 'cell%' AND e.sid = s.sid;
```

Wait Event	Description
cell interconnect retransmit during physical read	Database wait during retransmission for an I/O of a single-block or multiblock read
cell list of blocks physical read	Cell equivalent of db file parallel read
cell single block physical read	Cell equivalent of db file sequential read
cell multiblock physical read	Cell equivalent of db file scattered read
cell smart table scan	Database wait for table scan to complete
cell smart index scan	Database wait for index or IOT fast full scan
cell smart file creation	Database wait for file creation operation
cell smart incremental backup	Database wait for incremental backup operation
cell smart restore from backup	Database wait during file initialization for restore



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Oracle uses a specific set of wait events for disk I/O to Exadata Storage Server. Information about cell wait events is displayed in v\$ dynamic performance views, such as V\$SESSION_WAIT, V\$SYSTEM_EVENT and V\$SESSION_EVENT.

The slide shows an example of a query used to display a summary of cell wait events for the current session. Examples of the output from this query are shown on the following pages. A list of commonly encountered cell wait events with a brief description is also shown.

Note that for detailed analysis purposes, the cell wait events can also identify the corresponding cell and grid disk being accessed which is more useful for performance and diagnostics purposes than the database file number and block number information that is provided by wait events for conventional storage.

For more information about cell-specific wait events, refer to the *Oracle Exadata Storage Server Software User's Guide*.

Smart Scan Statistics: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

COUNT(*)
-----
8602831

Elapsed: 00:00:11.76

SQL> SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
  2 WHERE s.statistic# = m.statistic# AND
  3 (s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%' 
  4 OR s.name LIKE 'cell IO%');

NAME                                MB
-----
physical read total bytes          18005.6953
physical write total bytes         0
cell physical IO interconnect bytes      120.670433
cell physical IO bytes sent directly to DB node to balance CPU u
cell physical IO bytes saved during optimized file creation
cell physical IO bytes saved during optimized RMAN file restore
cell physical IO bytes eligible for predicate offload      18005.6953
cell physical IO bytes saved by storage index
cell physical IO interconnect bytes returned by smart scan      120.670433
cell IO uncompressed bytes          18005.6953
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows the first example query introduced earlier in this lesson. The execution plan shown earlier in the lesson indicated the use of Smart Scan. This is backed up by the statistics which follow the query. The statistics show that although the query referenced over 18000 MB of data, Smart Scan reduced the amount of I/O transported over the storage interconnect to a little more than 120 MB. In other words, Smart Scan reduced the amount of I/O transported to the database server by more than 99% compared with what would have been required using non-Exadata storage.

Note also that the amount of data returned by Smart Scan matches the amount of data transported across the storage interconnect. This is a sign that all the I/O in this example is associated with Smart Scan.

Smart Scan Wait Events: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

COUNT(*)
-----
8602831

Elapsed: 00:00:11.76

SQL> SELECT DISTINCT event, total_waits, time_waited/100 wait_secs,
  2  average_wait/100 avg_wait_secs
  3  FROM V$SESSION_EVENT e, V$MYSTAT s
  4  WHERE event LIKE 'cell%' AND e.sid = s.sid;

EVENT                      TOTAL_WAITS    WAIT_SECS AVG_WAIT_SECS
-----                      -----
cell smart table scan          9026        11.05      .0012
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows the same example query as the previous slide, but this time the wait events associated with the query session are shown. Because "cell smart table scan" is the only cell-related event shown, you can conclude that all of the I/O associated with the example query was satisfied using Smart Scan. Also notice that the Smart Scan total wait time (11.05 sec) accounts for almost all of the query elapsed time (11.76 sec). This means that nearly all of the processing for this query occurred inside the Exadata cells.

Concurrent Transaction: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

  COUNT(*)
  -----
  8602831

Elapsed: 00:02:13.55

NAME                                     MB
-----
physical read total bytes                19047.2266
physical write total bytes               0
cell physical IO interconnect bytes    4808.85828
cell physical IO bytes sent directly to DB node to balance CPU u 0
cell physical IO bytes saved during optimized file creation   0
cell physical IO bytes saved during optimized RMAN file restore 0
cell physical IO bytes eligible for predicate offload          18005.6953
cell physical IO bytes saved by storage index                  0
cell physical IO interconnect bytes returned by smart scan    3767.32703
cell IO uncompressed bytes                 18005.6953

EVENT                                TOTAL_WAITS  WAIT_SECS AVG_WAIT_SECS
-----
cell list of blocks physical read      1           0         .0006
cell smart table scan              19238       32.7      .0017
cell single block physical read     133286      74.91     .0006
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The slide shows exactly the same query as before however this time a batch process was updating the CUSTOMERS table at the same time. The wait events confirm that Smart Scan is still used however this time a large number of cell single block physical reads are also encountered. The statistics quantify the effect. Notice how the physical I/O over the interconnect rises from approximately 120 MB in the previous example, to over 4800 MB in this case. Also note the increase in the query elapsed time and how it correlates with the wait times.

Extreme Concurrent Transaction: Example

```

SQL> select count(*) from customers where cust_valid = 'A';

  COUNT(*)
-----
  8602831

Elapsed: 00:15:04.29

NAME                                     MB
-----
physical read total bytes                28550.3125
physical write total bytes               0
cell physical IO interconnect bytes    28537.5555
cell physical IO bytes sent directly to DB node to balance CPU u 0
cell physical IO bytes saved during optimized file creation   0
cell physical IO bytes saved during optimized RMAN file restore 0
cell physical IO bytes eligible for predicate offload          18005.6953
cell physical IO bytes saved by storage index                  0
cell physical IO interconnect bytes returned by smart scan     17992.9383
cell IO uncompressed bytes                 18005.6953

EVENT                                TOTAL_WAITS  WAIT_SECS AVG_WAIT_SECS
-----
cell list of blocks physical read      1           0           .0006
cell single block physical read       1349704    683.94    .0005
cell smart table scan                9191        3.29     .0004

```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This time the example query is executed after another session has updated every row in the CUSTOMERS table, but before the update transaction is committed (or rolled-back). In this extreme case Smart Scan is still attempted, however since every record is subject to a pending transaction, every block I/O must be transferred to the traditional buffer cache read-consistency path. In this unusual case, attempting to use Smart Scan actually results in more I/O traffic across the storage interconnect than if Smart Scan was not used.

Following is a summary of the results for the same scenario, but with Smart Scan disabled:

```
SQL> select /*+ OPT_PARAM('cell_offload_processing' 'false') */ count(*) from customers where cust_valid = 'A';
```

Elapsed: 00:14:52.63

NAME	MB
cell physical IO interconnect bytes	28522.4922
cell physical IO bytes eligible for predicate offload	0

EVENT	TOTAL_WAITS	WAIT_SECS	AVG_WAIT
cell single block physical read	1346130	678.83	.0005
cell list of blocks physical read	2	0	.0007

Migrated Rows: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

  COUNT(*)
  -----
  8602831

Elapsed: 00:00:14.02

NAME                                MB
-----
physical read total bytes           22327.5781
physical write total bytes          0
cell physical IO interconnect bytes 130.069008
cell physical IO bytes sent directly to DB node to balance CPU u 0
cell physical IO bytes saved during optimized file creation    0
cell physical IO bytes saved during optimized RMAN file restore 0
cell physical IO bytes eligible for predicate offload           22324.6094
cell physical IO bytes saved by storage index                   0
cell physical IO interconnect bytes returned by smart scan 127.100258
cell IO uncompressed bytes          22324.6094

EVENT                               TOTAL_WAITS   WAIT_SECS AVG_WAIT_SECS
-----
cell single block physical read      236          .14        .0006
cell smart table scan                10880         13.19     .0012
cell multiblock physical read       17           .02        .0009
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

In this example, the CUSTOMERS table has been updated in a way that resulted in row migration across approximately 6.5% of the data blocks in the table. Now when the query is executed, the query timing, the statistics and the wait events are close to the original values observed without any migrated rows. However there are still a noticeable difference between the amount of data returned by Smart Scan and amount of physical interconnect I/O. This difference, along with the cell physical read wait events, are symptoms of the row migration present in the CUSTOMERS table.

I/O Sent Directly to Database Server to Balance CPU Usage: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

  COUNT(*)
  -----
  8602831

Elapsed: 00:01:42.59

NAME                                     MB
-----
physical read total bytes           18005.6953
physical write total bytes          0
cell physical IO interconnect bytes 2475.24233
cell physical IO bytes sent directly to DB node to balance CPU u 2394.57133
cell physical IO bytes saved during optimized file creation 0
cell physical IO bytes saved during optimized RMAN file restore 0
cell physical IO bytes eligible for predicate offload    18005.6953
cell physical IO bytes saved by storage index            0
cell physical IO interconnect bytes returned by smart scan 2475.24233
cell IO uncompressed bytes           18005.6953

EVENT                                TOTAL_WAITS   WAIT_SECS AVG_WAIT_SECS
-----
cell smart table scan                  9128        98.19      .0108
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Commencing with Exadata Storage Server version 11.2.2.3.0, Exadata Storage Server may choose send data blocks back to the database server for processing rather than processing them within the cell. This happens when the storage server CPU utilization reaches an internal threshold and there is spare CPU capacity available in the database layer. The aim is to optimize the overall utilization of CPU resources across the Exadata Database Machine.

This behavior is indicated by a non-zero value associated with the `cell physical IO sent directly to DB node to balance CPU usage` statistic, and is usually an indicator that the storage servers are busy performing CPU intensive activities. For example, this may occur when numerous concurrent Smart Scan operations are already being processed, or when cells are busy performing numerous concurrent data decompression operations.

In this example, almost 2400 MB of data was sent to the database server in an attempt to balance CPU utilization across the environment. While the query ran significantly slower than the same query using the full power of Smart Scan, the result is typically better than further loading already busy storage servers.

Column Filtering: Example

```
SQL> select * from customers;
```

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)
0	SELECT STATEMENT		1239K	244M	10242	(1)
1	TABLE ACCESS STORAGE FULL	CUSTOMERS	1239K	244M	10242	(1)

NAME

MB

physical read total bytes	290.335938
cell physical IO interconnect bytes	290.335938
cell physical IO bytes eligible for predicate offload	0
cell physical IO interconnect bytes returned by smart scan	0

```
SQL> select cust_email from customers;
```

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)
0	SELECT STATEMENT		1239K	20M	10235	(1)
1	TABLE ACCESS STORAGE FULL	CUSTOMERS	1239K	20M	10235	(1)

NAME

MB

physical read total bytes	290.289063
cell physical IO interconnect bytes	29.0223618
cell physical IO bytes eligible for predicate offload	290.289063
cell physical IO interconnect bytes returned by smart scan	29.0223618

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This example examines the effect of column filtering using two simple SQL queries.

The top query selects the entire customers table. The associated query execution plan shows that the table scan is offloaded to Exadata Storage Server. However, because the query asks for all of the columns to be returned, the entire table must be transported across the storage network.

The bottom query selects just one column from the `customers` table. Note that the associated query execution plan provides no explicit notification regarding column filtering. However, it does indicate that the optimizer expects to process a smaller volume of data (20M bytes compared to 244M bytes) which can be used to infer that column filtering will take place. The proof of column filtering can be seen from the statistics associated with the query. This time the entire table is eligible for predicate offload (approximately 290 MB) but only the data associated with the `cust_email` column (approximately 29 MB) is transported across the storage network.

Summary

In this lesson, you should have learned how to:

- Describe Smart Scan and the query processing that can be offloaded to Exadata Storage Server
- Describe the requirements for Smart Scan
- Describe the circumstances that prevent using Smart Scan
- Identify Smart Scan in SQL execution plans
- Use database statistics and wait events to confirm how queries are processed



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Quiz



The `CELL_OFFLOAD_PLAN_DISPLAY` initialization parameter enables Smart Scan:

- a. True
- b. False

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: b

The `CELL_OFFLOAD_PROCESSING` parameter is used to enable Smart Scan.

Quiz



Smart Scan is enabled and the query plan indicates the use of Smart Scan. However, the cell physical IO interconnect bytes returned by smart scan statistic shows values that are much larger than expected. Which of the following could be the cause?

- a. Migrated rows
- b. Full table scan over an index organized table
- c. Concurrent uncommitted transactions
- d. A fast full scan of a compressed index



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Answer: a, c

Smart Scan is used to the extent possible in these two situations however in both cases the efficiency of Smart Scan is reduced. Smart Scan cannot be used for the other situations.

Practice 9 Overview: Using Smart Scan

In these practices, you will exercise Exadata Smart Scan and examine various statistics and wait events to determine what is occurring.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only